Sequential and Patch Analyses for Object Removal Video Forgery Detection and Localization

Mohammed Aloraini, *Member, IEEE*, Mehdi Sharifzadeh, *Member, IEEE*, and Dan Schonfeld, *Senior Member, IEEE*

***Abstract*— In recent years, video surveillance has become essential for security applications used to monitor many organizations and locations, and it is therefore important to ensure the reliability of these surveillance videos. Unfortunately, surveillance videos can be forged with little effort by deleting an object from a video scene while leaving no visible traces. A fundamental challenge in video security is to determine whether or not an object has been removed from a video. This task is particularly challenging due to the lack of ground truth bases that can be used to verify the originality and integrity of video contents. In this paper, we propose a novel approach based on sequential and patch analyses to detect object removal forgery and to localize forged regions in videos. Sequential analysis is performed by modeling video sequences as stochastic processes, where changes in the parameters of these processes are used to detect a video forgery. Patch analysis is performed by modeling video sequences as a mixture model of normal and anomalous patches, with the aim to separate these patches by identifying the distribution of each patch. We localize forged regions by visualizing the movement of removed objects using anomalous patches. We conduct our experiments at both pixel and video levels to determine the effectiveness and efficiency of our approach to detection of video forgery. The experimental results show that our approach achieves excellent detection performance with low-computational complexity and leads to robust results for compressed and low-resolution videos.**

***Index Terms*— Sequential analysis, patch analysis, spatio-temporal analysis, video forensics, object removal video forgery.**

I. Introduction

FOR many years, surveillance videos have become essential for public security that monitors many organizations, and thus, it is important to ensure the reliability of these surveillance videos. If these videos are manipulated, it could lead to many critical problems that are related to public security or legal evidence [1]. These manipulated videos are often eye-deceiving and appear in a way that is realistic and believable. Media are sometimes tricked to use fake videos as if they are real. As a result, video contents should be carefully analyzed to ensure their originality and integrity [2].

In general, video forgery can be divided into two categories: frame-based and object-based forgeries [3]. Frame-based forgery is created by deleting frames from a video scene, inserting frames into the video scene, or duplicating frames in the video scene. This forgery is easy to perform by using any of basic editing tools because a manipulator needs only to divide a video into frames to create a video forgery. Object-based forgery is created by adding new moving objects to a video scene or removing existing moving objects from the video scene. It is difficult to add moving objects without leaving invisible traces since videos might expose different motions and illuminations. Hence, object-based video forgery often refers to removing objects from a video. An example of object removal video forgery is illustrated in Fig.1, where the man in the red box has been removed from the scene. This forgery is more complicated to perform compared to frame-based forgery because a forger needs to manipulate specific regions in video frames while maintaining temporal consistency between these frames.

Creating an automatic approach to detect forged videos is a challenging problem due to the lack of truthful bases that can be used to verify the originality and integrity of video contents. A forged video may not only run through deleting an object from a video scene, but also run through other complex processes including compression, rotation, and resizing. These processes make forgery detection more challenging. Furthermore, if a forger removes an object (e.g., person) from a video scene, it becomes difficult to detect forged regions due to the high correlation between these forged regions and original regions. As a result, it is challenging to ensure the originality and integrity of video contents.

In this paper, we propose a novel approach based on sequential and patch analyses to detect object removal forgery and localize forged regions in videos. We perform sequential analysis by modeling video sequences as stochastic processes, where changes in the parameters of these processes indicate a video forgery. Patch analysis is performed by modeling video sequences as a mixture model of normal and anomalous patches, with the aim to separate these patches by identifying the distribution of each patch. Finally, we localize forged regions in videos by visualizing a movement of removed objects using anomalous patches.

Manuscript received October 4, 2019; revised January 28, 2020 and April 14, 2020; accepted May 3, 2020. Date of publication May 7, 2020; date of current version March 5, 2021. This article was recommended by Associate Editor H. Wang. *(Corresponding author: Mohammed Aloraini.)*

Mohammed Aloraini is with the Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607 USA, and also with the Department of Electrical Engineering, College of Engineering, Qassim University, Unaizah 56452, Saudi Arabia (e-mail: malora2@uic.edu).

Mehdi Sharifzadeh and Dan Schonfeld are with the Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607 USA (e-mail: mshari5@uic.edu; dans@uic.edu).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2020.2993004

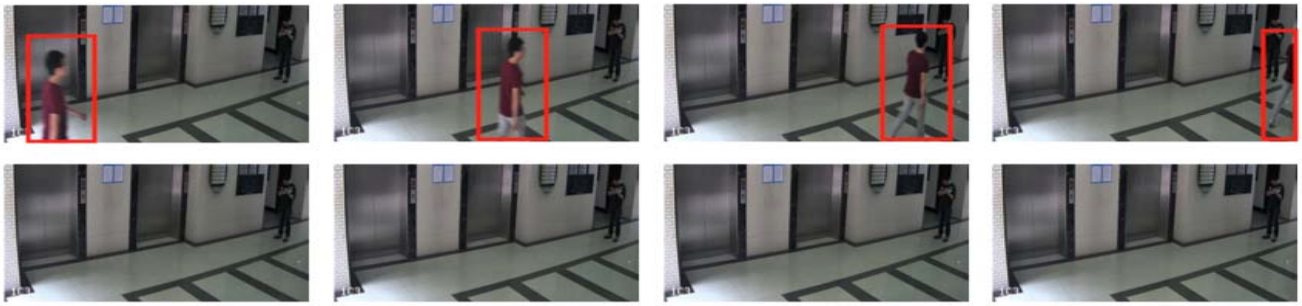


Fig. 1. An example of object removal video forgery: Images on the top row indicate frames from the original video; Images on the bottom row indicate corresponding frames from the tampered video where the man in the red box has been removed from the scene.

This paper is an extension to our paper [4] with the following contributions:

1) We model video sequences as multivariate processes to improve the detection accuracy.
2) We model our patch analysis approach as a mixture model of normal and anomalous patches to further improve the detection accuracy.
3) We use the multivariate sequential and patch analyses to exponentially reduce the computational complexity. As a result, our approach is scalable.

The rest of the paper is organized as follows. Related work is provided in Sec. II. Our data model is explained in Sec. III. Our proposed approach is described in Sec. IV. The spatiotemporal filter is presented in Sec. IV-A. Univariate and multivariate sequential analyses are presented in Sec. IV-B. Our patch analysis approach is explained in Sec. IV-C. Object removal visualization is described in Sec. IV-D. The experimental results are discussed in Sec. V. This paper is concluded in Sec. VI.

II. Related Works

Although several works have been conducted to review video forensic approaches [5]–[9], most of these works focused on detecting frame-based forgery [10]. These works can be divided into three categories: motion-based [11]–[15], correlation-based [16]–[19], and compression-based [20]–[24]. First, motion-based approaches use inconsistencies of motion vectors as an evidence of a frame deletion or insertion. A drawback of these approaches is that the detection accuracy decreases when compression increases. Second, correlation-based approaches use high correlation between suspicious frames as an indication of a frame duplication. These approaches fail to detect the frame duplication when the frame duplication occurs in static background frames or performs in a different order. Third, compression-based approaches declare video forgery by detecting double compression. These approaches are not applicable when a complete group of pictures (GOP) is removed, or recompression is occurred without video tampering.

Only a few works have been conducted to detect object-based forgery compared to frame-based forgery [25]. These works tackle two types of object-based forgery: object insertion video forgery and object removal video forgery. The following works are proposed to detect object insertion video forgery [26]–[32]. Some approaches use correlation between blurring features [26], or edge features [27], to detect blue screen compositing. The forgery is detected by examining changes in correlation patterns between these features. These approaches fail to detect video forgery if the background of a video is green or blue. Other approaches use DCT coefficients [28], or luminance and contrast [30], as local features to measure the similarity between foreground and background. The forgery is detected by identifying inconsistencies in these features between foreground and background. A limitation of these approaches is that the detection accuracy decreases when the bit rate of videos decreases. Conotter *et al.* proposed an approach that uses projectile motion to identify falsified objects [31]. D'Avino *et al.* presented an approach that uses deep learning to learn an intrinsic model of an original video, where a video is classified as forged if it does not fit the learned model [32].

Object removal video forgery is achieved by using inpainting algorithms [33]–[35]. The following works are proposed to detect object removal video forgery [36]–[44]. Zhang *et al.* developed an approach that uses ghost shadow artifact to identify inconsistencies between foreground mosaic and trajectory of moving foreground [36]. Hsu *et al.* introduced an approach that uses temporal correlation of noise residues to identify irregular changes in the correlation of noise residues throughout video frames [37]. A similar approach uses correlation of Hessian matrices to detect object removal forgery [39]. Richao *et al.* presented an approach that uses object contour features with a support vector machine (SVM) algorithm to detect removed moving objects with static background [41]. Another approach uses steganalytic features, which are extracted from motion residual matrices, with ensemble classifiers to classify a frame into three categories: pristine, forged, and double compressed [42]. Lichao et al presented an approach based on compressive sensing to detect removed moving objects with static background [43]. All of the above works can detect video forgery, but they cannot localize forged regions in videos. Lin *et al.* introduced an approach based on spatiotemporal coherence analysis to detect and localize tampered regions [44]. A limitation of this approach is that detection performance drops significantly when tampered videos are saved in compressed formats. Deep Convolutional Neural Networks (CNNs) require large data sets for training to achieve excellent

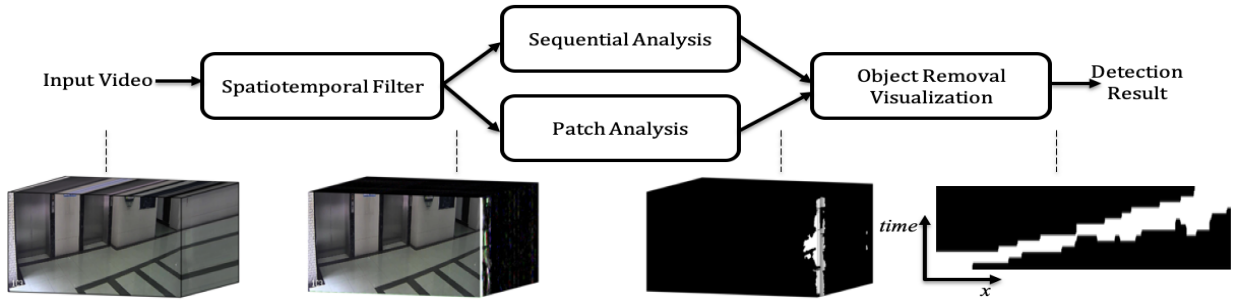


Fig. 2. A flowchart of the proposed approach to object removal video forgery detection and localization.

results [9]. However, there are a few object removal forgery data sets that are publicly available [8], [9]. As a result, CNNs are not ideal to tackle the object removal forgery problem.

III. Data Model

We propose an approach based on sequential and patch analyses. Our approach requires the following assumptions about video sequences. First, video sequences are assumed to be captured by a static camera. Our approach aims to detect changes between video frames due to objects removal. When a camera is moving, it will generate video frames with different backgrounds. Thus, it would be hard to distinguish between changes due to movement of a camera or objects removal. Therefore, our approach mainly focuses on surveillance video clips where the camera is static. Second, video frames must be well-registered into a common reference frame prior to performing video forgery detection. We assume well-registered frames to eliminate changes due to non-registered frames.

In general, a pixel's intensity is corrupted by three sources of additive noise: photon counting noise, readout noise, and quantization noise [45], [46]. Photon counting noise comes from a discrete random number of photons striking the sensor and is modeled as a Poisson process. Readout noise is produced by the amplifier and is modeled as a Gaussian process. Quantization noise results from the selection of discrete pixel values and is modeled as a uniform distribution. It is extremely difficult to find the exact distribution of the additive noise that is added to pixel intensities [47]. Many previous works approximate this additive noise to be normally distributed [48]. Therefore, we assume pixels' values are drawn from a normal distribution, independent and identically distributed, and the variance remains constant throughout video frames while the mean is dependent on the scene.

The following notation and definitions will be used throughout the paper. *Scalars* are written as lowercase letters, *vectors* are written as underlined lowercase letters, and *matrices* are written as uppercase letters. A *block* is defined as a group of spatially adjacent pixels, and it is described by a feature vector. A *patch* is defined as a set of temporally adjacent blocks, and it is described by a set of feature vectors.

IV. Object Removal Video Forgery Detection and Localization

We briefly describe our approach in the following steps, as illustrated in Fig.2. First, we apply spatial decomposition (i.e., Laplacian pyramid) to the video frames, followed by temporal high pass filter to detect edges spatially and highlight variations temporally. Then, we perform sequential analysis by modeling video sequences as stochastic processes, where changes in the parameters of these processes indicate a video forgery. If the patch analysis is performed, we model video sequences as a mixture model of normal and anomalous patches. These patches are subsequently separated by identifying if they have been generated by the normal or anomalous distribution. Finally, we localize forged regions by visualizing a movement of removed objects using anomalous patches.

A. Spatiotemporal Filter

We apply the spatiotemporal filter, which is presented in Fig.4, for two reasons. First, we use the spatiotemporal filter to expose traces (edges) that are left at a removed object boundary due to structure inpainting, texture inpainting, or a combination of the two. Second, we apply the spatiotemporal filter to zero out pixels' values at static regions, as shown in Fig.3. As a result, this filtering process enables sequential and patch analyses to accurately detect changes (i.e., anomalous) in forged videos.

Since the size of the removed objects is unknown, a video is divided into frames, and Laplacian pyramid decomposition [49] (spatial filtering) is applied to each frame to detect edges in all possible scales. The Laplacian pyramid decomposition subtracts each frame from its blurred version to form a video scale, down-samples each frame by half, and repeats this process until the minimum resolution of a frame is reached. This process constructs multiscale videos that represent edges at different scales, as shown in Fig.4. We perform temporal filtering at each scale by using the pixels' values throughout time in a frequency band and apply a high-pass filter to remove static edges.

B. Sequential Analysis

The spatiotemporal filter results in multiscale videos as shown in Fig.4, hence applying sequential analysis in each scale would result in very high computation time. Therefore, we first reconstruct the Laplacian pyramid to transfer multiscale videos to one video scale (i.e., the input video scale) [49]. The Laplacian pyramid reconstruction upsamples and blurs each frame in the lowest scale of Laplacian pyramid decomposition, adds the upsampled and blurred version to

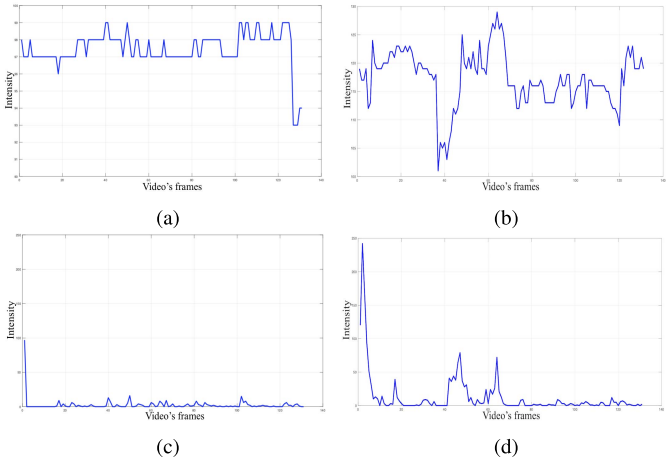


Fig. 3. Intensity traces through video frames for (a) an authentic pixel before using the spatiotemporal filter (b) a forged pixel before using the spatiotemporal filter (c) an authentic pixel after using the spatiotemporal filter (d) a forged pixel after using the spatiotemporal filter.

the next lowest scale to obtain the approximation of each frame at the next scale, and repeats this process until the input video scale is reached. Then, we apply the following univariate or multivariate sequential analysis to the input video scale.

1) Univariate Analysis: We model the object removal forgery as an additive change in the mean value of probability density function associated with a pixel sequence in a video. We begin the analysis by introducing a null hypothesis H_0 that states there is no change in a pixel's mean value, and an alternative hypothesis H_1 that states there are changes in a pixel's mean value. The mean before the change μ_0 is assumed to be known, and the mean after the change μ_1 is assumed to be completely unknown but different than μ_0. We formulate the null and alternative hypotheses as follows:

$$\begin{aligned} \mathbf{H_0} &= \{\mu : \mu = \mu_0, n < t\} \\ \mathbf{H_1} &= \{\mu : \mu \neq \mu_0, n \geq t\} \end{aligned} \tag{1}$$

where n is the frame index, and t is the true change time. Based on our assumption that pixels' values are drawn from a normal distribution, independent and identically distributed as discussed in Sec. III, we form the null and alternative likelihoods as follows:

$$\ell_k^{H_0}(x_i) = p(x_k, \ldots, x_n | H_0) = \frac{1}{\sqrt{2\pi\sigma^2}} \prod_{i=k}^{n} e^{\frac{-(x_i-\mu_0)^2}{2\sigma^2}} \tag{2}$$

$$\ell_k^{H_1}(x_i) = \sup_{\mu_1} p(x_k, \ldots, x_n | H_1) = \sup_{\mu_1} \frac{1}{\sqrt{2\pi\sigma^2}} \prod_{i=k}^{n} e^{\frac{-(x_i-\mu_1)^2}{2\sigma^2}} \tag{3}$$

where x_i represents values of a pixel throughout video frames; μ_i and σ^2 are the mean and variance of the pixel, respectively. Using (2) and (3), we form log-likelihood ratio as follows:

$$\Lambda_k^n = \ln \frac{\sup_{\mu_1} p(x_k, \ldots, x_n | H_1)}{p(x_k, \ldots, x_n | H_0)} \tag{4}$$

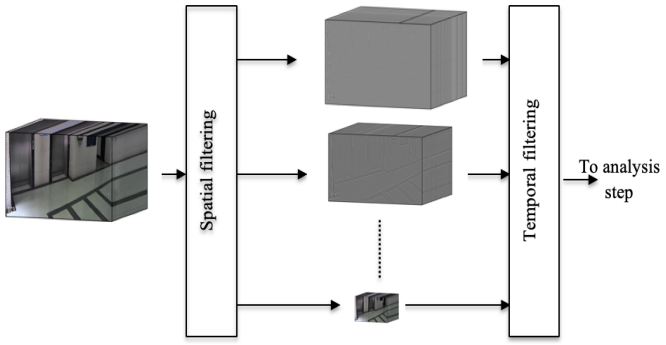


Fig. 4. An overview of the proposed spatiotemporal filter.

$$= \ln \frac{\sup_{\mu_1} \prod_{i=k}^{n} e^{\frac{-(x_i-\mu_1)^2}{2\sigma^2}}}{\prod_{i=k}^{n} e^{\frac{-(x_i-\mu_0)^2}{2\sigma^2}}} \tag{5}$$

The unknown mean is replaced by its maximum likelihood estimate (MLE) as follows:

$$\hat{x}_k^n = \frac{1}{n-k+1} \sum_{i=k}^{n} x_i. \tag{6}$$

Then, the log-likelihood ratio becomes

$$\Lambda_k^n = \frac{1}{2\sigma^2} \left[\sum_{i=k}^{n} (x_i - \mu_0)^2 - \sum_{i=k}^{n} (x_i - \hat{x}_k^n)^2 \right] \tag{7}$$

$$= \frac{1}{2\sigma^2} \sum_{i=k}^{n} \left[(x_i - \mu_0)^2 - (x_i - \hat{x}_k^n)^2 \right] \tag{8}$$

$$= \sum_{i=k}^{n} \left[\frac{(\hat{x}_k^n - \mu_0) x_i}{\sigma^2} + \frac{\mu_0^2 - \hat{x}_k^{n^2}}{2\sigma^2} \right]. \tag{9}$$

Then, the generalized log-likelihood g_k^n and alarm detection τ become

$$g_k^n = \max_{1 \leq k \leq n} \Lambda_k^n \tag{10}$$

$$\tau = \min\{n \geq 1 : g_k^n \geq h_u\}. \tag{11}$$

In (11), τ is a frame index where a change occurs, n is the discrete time index (frame index), and h_u is a threshold.

Let us summarize the univariate analysis. First, μ_0 and σ^2 are assumed to be known. In fact, they can be estimated using a pixel's values throughout all video frames. $\hat{x}_k^n$ is calculated sequentially by using all previous values of a pixel as described in (6). Finally, a change is declared if g_k^n exceeds a certain threshold h_u and this change is located at frame index τ.

2) Multivariate Analysis: We model the object removal forgery as an additive change in the mean parameter of probability density function associated with feature vectors that are extracted from dividing video frames into distinct blocks. We assume feature vectors are drawn from a Gaussian distribution, independent and identically distributed, with the following probability density function

$$p(\underline{y}_i) = \frac{1}{\sqrt{(2\pi)^r |\Sigma|}} e^{-\frac{1}{2}\left(\underline{y}_i - \underline{\mu}\right)^T \Sigma^{-1} \left(\underline{y}_i - \underline{\mu}\right)} \tag{12}$$

Algorithm 1 Object Removal Forgery Detection and Localization Based on Patch Analysis

Require: V ▷ Input video
b ▷ Block size
p ▷ Patch size
Ensure: M ▷ Object removal visualization result
$N_f \leftarrow NumberOfFrames$
$R \leftarrow EmptyArray$ ▷ Size of R is the same as size of V
$Apply\ spatiotemporal\ filter\ on\ V$
$Divide\ R\ and\ filtered\ V\ into\ distinct\ blocks\ b \times b$
for *each block* (B_{Vi}) *in* V **do**
  $\{\underline{y}_1, \ldots, \underline{y}_{N_f}\} \leftarrow GetAllFeatureVectorsThrough\ V$
  $Initialization: k = 0,\ N_0 = \{\underline{y}_1, \ldots, \underline{y}_{N_f}\},\ A_0 = \{\}$
  while $k \leq N_f - p$ **do**
    $C = \{y_{k+1}, \ldots, y_{k+p}\}$
    $Compute\ likelihood\ ratio\ \Lambda_k\ based\ on$ (31)
    $Compute\ log-likelihood\ \ln(\Lambda_k)\ based\ on$ (32)
    if $\ln(\Lambda_k) < h_p$ **then**
      $N_k = N_{k-1} - \{y_{k+1}, \ldots, y_{k+p}\}$
      $A_k = A_{k-1} \cup \{y_{k+1}, \ldots, y_{k+p}\}$
      $k = k + p$ ▷ C is an anomalous patch, hence k is increased by p
    else
      $N_k = N_{k-1}$
      $A_k = A_{k-1}$
      $k = k + 1$ ▷ C is a normal patch, hence k is increased by 1
    end if
  end while
  $B_{Ri} \leftarrow BinaryArray$ ▷ Explained in Sec.IV-D
end for ▷ $B_{Ri} \equiv$ Corresponding block in R
$M \leftarrow RemovalVisualization$ ▷ Explained in Sec.IV-D

where $\underline{\mu}$ and Σ are the mean vector and covariance matrix of feature vectors, respectively; r is the dimension of the feature vector.

We begin with a general case [50] where the mean vector before the change $\underline{\mu_0}$ is limited by an upper bound, and mean vector after the change $\underline{\mu_1}$ is limited by a lower bound. Then, the null and alternative hypotheses become

$$\begin{aligned} \mathbf{H_0} &= \{\underline{\mu} : ||\underline{\mu} - \underline{\mu_0}||_\Sigma^2 \leq a^2, n < t\} \\ \mathbf{H_1} &= \{\underline{\mu} : ||\underline{\mu} - \underline{\mu_0}||_\Sigma^2 \geq b^2, n \geq t\} \end{aligned} \tag{13}$$

where $||\underline{\mu} - \underline{\mu_0}||_\Sigma^2 = \left(\underline{\mu} - \underline{\mu_0}\right)^T \Sigma^{-1} \left(\underline{\mu} - \underline{\mu_0}\right)$; t is the true change time; n is the frame index; $a < b$. Then, the log-likelihood ratio becomes

$$\Lambda_k^n = \ln \frac{\sup_{||\underline{\mu} - \underline{\mu_0}||_\Sigma \geq b} \prod_{i=k}^n p(\underline{y}_i)}{\sup_{||\underline{\mu} - \underline{\mu_0}||_\Sigma \leq a} \prod_{i=k}^n p(\underline{y}_i)} \tag{14}$$

$$= \ln \frac{\sup_{||\underline{\mu} - \underline{\mu_0}||_\Sigma \geq b} e^{-\frac{1}{2}\sum_{i=k}^n \left(\underline{y}_i - \underline{\mu}\right)^T \Sigma^{-1} \left(\underline{y}_i - \underline{\mu}\right)}}{\sup_{||\underline{\mu} - \underline{\mu_0}||_\Sigma \leq a} e^{-\frac{1}{2}\sum_{i=k}^n \left(\underline{y}_i - \underline{\mu}\right)^T \Sigma^{-1} \left(\underline{y}_i - \underline{\mu}\right)}} \tag{15}$$

$$= \sup_{||\underline{\mu} - \underline{\mu_0}||_\Sigma \geq b} \left\{ -\frac{1}{2} \sum_{i=k}^n \left(\underline{y}_i - \underline{\mu}\right)^T \Sigma^{-1} \left(\underline{y}_i - \underline{\mu}\right) \right\} - \sup_{||\underline{\mu} - \underline{\mu_0}||_\Sigma \leq a} \left\{ -\frac{1}{2} \sum_{i=k}^n \left(\underline{y}_i - \underline{\mu}\right)^T \Sigma^{-1} \left(\underline{y}_i - \underline{\mu}\right) \right\}. \tag{16}$$

The unknown parameter is replaced by its maximum likelihood estimate (MLE) as follows:

$$\hat{\underline{y}}_k^n = \frac{1}{n-k+1} \sum_{i=k}^n \underline{y}_i. \tag{17}$$

Then, the log-likelihood ratio becomes

$$\frac{2}{n-k+1} \Lambda_k^n = \begin{cases} -(Z_k^n - b)^2, & Z_k^n < a \\ -(Z_k^n - b)^2 + (Z_k^n - a)^2, & a \leq Z_k^n \leq b \\ (Z_k^n - a)^2, & Z_k^n > b \end{cases} \tag{18}$$

where Z_k^n is given by

$$Z_k^n = [\left(\hat{\underline{y}}_k^n - \underline{\mu_0}\right)^T \Sigma^{-1} \left(\hat{\underline{y}}_k^n - \underline{\mu_0}\right)]^{1/2}. \tag{19}$$

We set $a = b = 0$ in (18) because we are interested in the case where the mean vector before the change $\underline{\mu_0}$ is assumed to be known and the mean vector after the change $\underline{\mu_1}$ is assumed to be completely unknown but different than $\underline{\mu_0}$. Then, the generalized log-likelihood g_k^n and alarm detection τ become

$$g_k^n = \max_{1 \leq k \leq n} \left\{ \frac{n-k+1}{2} (Z_k^n)^2 \right\} \tag{20}$$

$$\tau = \min\{n \geq 1 : g_k^n \geq h_m\}. \tag{21}$$

Let us summarize the multivariate analysis. First, $\underline{\mu_0}$ and Σ are assumed to be known. In fact, they can be estimated using feature vectors of a particular block throughout all video frames. $\hat{\underline{y}}_k^n$ and Z_k^n are calculated sequentially by using all previous feature vectors of a particular block as described in (17) and (19), respectively. Finally, a change is declared if g_k^n exceeds a certain threshold h_m and this change is located at frame index τ.

The current formulation of univariate and multivariate analyses enables us to detect only a single change in the whole time (frame) series. However, we need to detect multiple changes, hence we use binary segmentation [51]. Binary segmentation starts by detecting a single change in the complete time series. If there is a change, it splits the time series around this change into two sub-series and repeats this process until no changes are detected. By using binary segmentation, the time series that represents video frames will be divided into segments.

A segment is considered as a forged segment (removed object segment) if two conditions are met: (1) the mean of this segment exceeds a certain threshold to identify whether this segment belongs to a background or a removed object, and (2) the length of this segment is less than a certain threshold based on our definition that removed objects are moving as discussed in Sec.I.

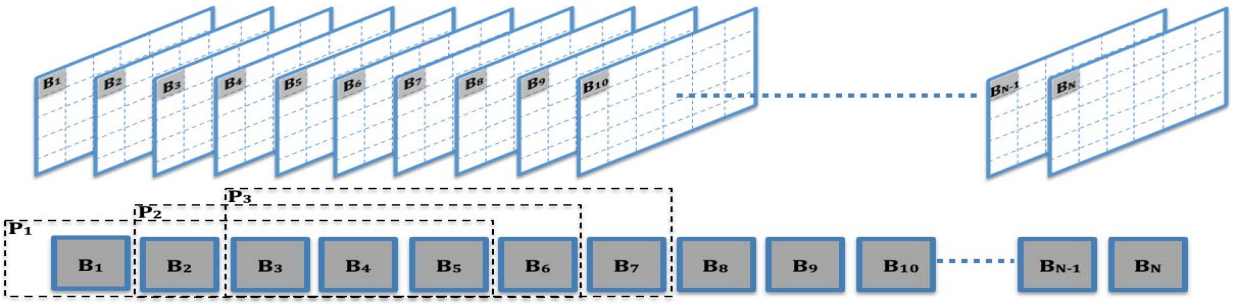


Fig. 5. Illustration of the proposed patch analysis approach. Top sequence shows video frames that are divided into non overlapping blocks with a selected gray block to apply patch analysis; bottom sequence indicates patch size $p = 5$ with overlapping step $s = 1$. Patch analysis starts by calculating the log-likelihood under the assumption that all blocks belong to normal set. Then, it calculates the log-likelihood under the assumption that P_1 belongs to anomalous set. If the difference between these two log-likelihoods is less than a threshold h_p, P_1 is moved from normal set to anomalous set. Otherwise, P_1 remains in the normal set. The analysis is repeated for $P_2, P_3, \ldots, P_{N-p}$.

C. Patch Analysis

We model the object removal forgery as a mixture model of normal and anomalous patches. A *patch* is defined as a set of temporally adjacent blocks, and it is described by a set of feature vectors. We assume that all feature vectors in a patch are either normal or anomalous. Some patches that are located at the border between forged and original regions may contain feature vectors that belong to both normal and anomalous sets. However, these patches are few because forged regions are generally small compared to the original regions in a video. Hence, we neglect patches at the border and consider only patches that contain either normal or anomalous feature vectors. We also assume that normal features are drawn from a Gaussian distribution, and anomalous features are drawn from a uniform distribution because anomalies are often assumed to be uniform [52], [53]. The probability density functions for normal $p_N(\underline{y})$ and anomalous $p_A(\underline{y})$ feature vectors are defined as

$$p_N(\underline{y}_i) = \frac{1}{\sqrt{(2\pi)^r |\Sigma|}} e^{-\frac{1}{2}\left(\underline{y}_i - \underline{\mu}\right)^T \Sigma^{-1} \left(\underline{y}_i - \underline{\mu}\right)} \tag{22}$$

$$p_A(\underline{y}_i) = \begin{cases} \dfrac{1}{(b-a)^r}, & \underline{y}_i \in (a,b)^r \\ 0, & Otherwise \end{cases} \tag{23}$$

where $\underline{\mu}$ and Σ are the mean vector and covariance matrix of feature vectors, respectively; r is the dimension of the feature vector; a and b are the minimum and maximum values of arbitrary feature vectors, respectively. We let N_k and A_k be the sets of normal and anomalous feature vectors, at frame index k, respectively. Initially, all feature vectors of a particular block are put in a normal set while an anomalous set is empty.

We begin with a general case where the null hypothesis H_0 states that there is at least one feature vector $\underline{y}_i$ in a patch belongs to a normal set and an alternative hypothesis H_1 states that all feature vectors $\underline{y}_i$ in the patch belong to an anomalous set. We formulate the null and alternative hypotheses as follows:

$$\begin{aligned} \mathbf{H_0} &= \{\exists \underline{y}_i \in C, \underline{y}_i \sim p_N(\underline{y}_i)\} \\ \mathbf{H_1} &= \{\forall \underline{y}_i \in C, \underline{y}_i \sim p_A(\underline{y}_i)\} \end{aligned} \tag{24}$$

where $C = \{y_{k+1}, \ldots, y_{k+p}\}$ is a patch that consists of p feature vectors. By assuming that the patches are generated in an independent manner, the likelihood of null ($\ell_k^{H_0}(\underline{y})$) and alternative ($\ell_k^{H_1}(\underline{y})$) hypotheses of the entire feature vectors for a particular block at an arbitrary frame k are as follows:

$$\begin{aligned} \ell_k^{H_0}(\underline{y}) = &\sum_{c_i \in (\mathbf{P}(C) - \{C\})} (1-\lambda)^{|N_{k-1} - c_i|} \prod_{\underline{y}_i \in (N_{k-1} - c_i)} p_N(\underline{y}_i) \\ &\times \Big((\lambda)^{|A_{k-1} \cup c_i|} \prod_{\underline{y}_i \in (A_{k-1} \cup c_i)} p_A(\underline{y}_i) \Big) \end{aligned} \tag{25}$$

$$= (1-\lambda)^{|N_{k-1}|} \prod_{\underline{y}_i \in N_{k-1}} p_N(\underline{y}_i) (\lambda)^{|A_{k-1}|} \tag{26}$$

$$\prod_{\underline{y}_i \in A_{k-1}} p_A(\underline{y}_i) \sum_{c_i \in (\mathbf{P}(C) - \{C\})} \left(\frac{\lambda}{1-\lambda}\right)^{|c_i|} \prod_{\underline{y}_j \in c_i} \frac{p_A(\underline{y}_j)}{p_N(\underline{y}_j)} \tag{27}$$

$$\ell_k^{H_1}(\underline{y}) = (1-\lambda)^{|N_{k-1} - C|} \prod_{\underline{y}_i \in (N_{k-1} - C)} p_N(\underline{y}_i) \tag{28}$$

$$\begin{aligned} &\Big((\lambda)^{|A_{k-1} \cup C|} \prod_{\underline{y}_i \in (A_{k-1} \cup C)} p_A(\underline{y}_i) \Big) \\ &= (1-\lambda)^{|N_{k-1}|} \prod_{\underline{y}_i \in N_{k-1}} p_N(\underline{y}_i) (\lambda)^{|A_{k-1}|} \\ &\quad \times \prod_{\underline{y}_i \in A_{k-1}} p_A(\underline{y}_i) \Big(\left(\frac{\lambda}{1-\lambda}\right)^{|C|} \prod_{\underline{y}_i \in C} \frac{p_A(\underline{y}_i)}{p_N(\underline{y}_i)} \Big) \end{aligned} \tag{29}$$

where $\mathbf{P}(C)$ is the power set of C, which is the set of all subsets of C; $|\cdot|$ is the cardinality of a set; and λ is the expected fraction of anomalies. Then, the likelihood ratio becomes

$$\begin{aligned} \Lambda_k = \frac{\ell_k^{H_0}(\underline{y})}{\ell_k^{H_1}(\underline{y})} = &\left(\frac{1-\lambda}{\lambda}\right)^{|C|} \prod_{\underline{y}_i \in C} \frac{p_N(\underline{y}_i)}{p_A(\underline{y}_i)} \\ &+ \sum_{c_i \in (\mathbf{P}(C) - \{C, \{\}\})} \left(\frac{1-\lambda}{\lambda}\right)^{|c_i|} \prod_{\underline{y}_j \in c_i} \frac{p_N(\underline{y}_j)}{p_A(\underline{y}_j)}. \end{aligned} \tag{30}$$

Based on our assumption that all feature vectors in a patch are either normal or anomalous, the probability of the second term in (30) to occur is zero. Hence, the likelihood and

log-likelihood become

$$\Lambda_k = \left(\frac{1-\lambda}{\lambda}\right)^{|C|} \prod_{\underline{y}_i \in C} \frac{p_N(\underline{y}_i)}{p_A(\underline{y}_i)} \tag{31}$$

$$\ln(\Lambda_k) = |C| \ln\left(\frac{1-\lambda}{\lambda}\right) + \sum_{\underline{y}_i \in C} p_N(\underline{y}_i) - \sum_{\underline{y}_i \in C} p_A(\underline{y}_i) \underset{H_1}{\overset{H_0}{\gtrless}} h_p \tag{32}$$

where h_p is the decision threshold.

Let us summarize patch analysis as described in Algorithm 1. Initially, all feature vectors of a particular block are put in a normal set while an anomalous set is empty. A patch is chosen in an overlapping manner with overlapping step $s = 1$. Then, we calculate the likelihood and log-likelihood using equations described in (31) and (32), respectively. If the log-likelihood is less than a threshold h_p, this patch is declared as an anomaly and it is moved from the normal set to the anomalous set. Otherwise, this patch remains in the normal set. The main steps of patch analysis approach are illustrated in Fig.5.

D. Object Removal Visualization

We construct a binary video where a pixel equals one in frames that belong to anomalous sets (changed segments) and equals zero in frames that belong to normal sets (unchanged segments). A video forgery is detected if a number of consecutive frames h_F have an area that is larger than a threshold h_A and contains only ones. In the experiment, we set h_F to 25 frames and h_A to 500 pixels based on the results of ROC curve shown in Fig.6b.

We localize the movement of removed objects by constructing another binary video where a pixel equals one in a frame where a change occurs until the last video frame and equals zero in the other frames; essentially, once a pixel's value becomes one, it maintains that value until the last video frame. This process will create paths of removed objects; these paths can be visualized by plotting the last spatiotemporal XT slice (width vs. time), which is a bird's-eye view of a video as shown in Fig.2.

V. Experimental Analysis

In this section, we describe the data set and detection performance measurements. We also analyze the results of our approach and compare our approach with state-of-the-art approaches. We carry out our experiments using a MacBook Pro with 2.9 GHz Intel dual core i7 CPU and 8 GB RAM.

A. Data Set

To the best of our knowledge, the only available video forgery data sets are SULFA [54] and SYSU-OBJFORG [42]. SULFA is a frame-based forgery, which is beyond the scope of this work. Therefore, we use SYSU-OBJFORG, where all videos are extracted from a static surveillance camera with a resolution of 1280×720 and 25 frames per second. This data set consists of 100 original videos and 100 object-based forged videos; each video is approximately 11 seconds in duration. According to the authors of [42], SYSU-OBJFORG is the largest object-based forged video data set in the literature. However, most of the forged videos are not realistic because the counterfeit regions can be identified using the naked eye. In other words, object-based forgery is performed in the middle of frames, e.g., a walking person is removed before leaving a video scene, so this person is seen for a couple of seconds and suddenly disappeared from the video scene. Hence, we use SYSU-OBJFORG data set to generate realistic object removal forged videos by using two recent inpainting algorithms [33], [35]. Fig.7 shows three examples of object removal forgery from the data set.

To evaluate the effectiveness of our approach, we generate three video sets from the data set. The first set is an uncompressed video set, which has object removal forged videos without compressing these videos. The second set is a compressed video set, which has object removal forged videos with compressing these videos using H.264/MPEG-4 with 1 Mbps. The third set is a low-resolution video set, which has object removal forged videos with reducing the original resolution by half, i.e., 640×360.

B. Evaluation Metric

We evaluate object removal forgery detection on video and pixel levels. The most important aspect in practice is to determine whether a video is forged or not, i.e., video level performance. However, the effectiveness of an algorithm is determined by how accurately the tampered regions can be identified in a video, i.e., pixel level performance. We measure the performance at video level by defining T_P as the correctly detected forged videos, F_P as original videos that have been incorrectly detected as forged, and F_N as falsely missed forged videos. Then, $Precision$, $Recall$, $F1$, and $Intersection\ over\ Union$ (IoU) are as follows:

$$Precision = \frac{T_P}{T_P + F_P} \tag{33}$$

$$Recall = \frac{T_P}{T_P + F_N} \tag{34}$$

$$F1 = \frac{2T_P}{2T_P + F_P + F_N} \tag{35}$$

$$IoU = \frac{T_P}{T_P + F_P + F_N} \tag{36}$$

$Precision$ shows the probability that a detected forgery is truly a forgery, $Recall$ indicates the probability that a forged video is detected, $F1$ score shows the average performance, and IoU score shows the worst case performance.

We measure the performance at pixel level by defining T_P as the correctly detected forged pixels, F_P as original pixels that have been incorrectly detected as forged, and F_N as falsely missed forged pixels. Then, we compute $Precision$, $Recall$, $F1$, and IoU as in (33), (34), (35), and (36) respectively.

C. Feature Selection

There are several feature extraction approaches that have been used to detect image forgery such as SIFT [55], and

TABLE I

DETECTION RESULTS AT PIXEL AND VIDEO LEVELS OF OBJECT REMOVAL VIDEO FORGERY USING SEQUENTIAL ANALYSIS WITH DIFFERENT BLOCK SIZES AND DIFFERENT VIDEO SETS

(A) DETECTION RESULTS OF UNCOMPRESSED VIDEO SET.

Block size	Pixel level				Video level			
	Precision (%)	Recall (%)	F1 (%)	IoU (%)	Precision (%)	Recall (%)	F1 (%)	IoU (%)
1	**76.54**	78.79	77.65	63.47	93.33	93.33	93.33	87.49
5	73.68	82.48	77.84	63.72	91.40	94.44	92.90	86.73
10	73.62	**85.80**	**79.25**	**65.63**	93.48	**95.56**	**94.51**	**89.58**
15	73.01	85.26	78.66	64.83	92.22	92.22	92.22	85.57
20	74.51	82.45	78.28	64.31	**94.25**	91.11	92.66	86.32

(B) DETECTION RESULTS OF COMPRESSED VIDEO SET.

Block size	Pixel level				Video level			
	Precision (%)	Recall (%)	F1 (%)	IoU (%)	Precision (%)	Recall (%)	F1 (%)	IoU (%)
1	**75.78**	78.50	77.11	62.75	92.22	92.22	92.22	85.57
5	74.26	80.69	77.34	63.05	89.36	93.33	91.30	84.00
10	73.29	**84.15**	78.34	64.39	91.30	93.33	92.31	85.71
15	74.00	83.23	**78.35**	**64.41**	92.39	**94.44**	**93.41**	**87.63**
20	75.31	81.03	78.07	64.03	**93.18**	91.11	92.13	85.42

(C) DETECTION RESULTS OF LOW-RESOLUTION VIDEO SET.

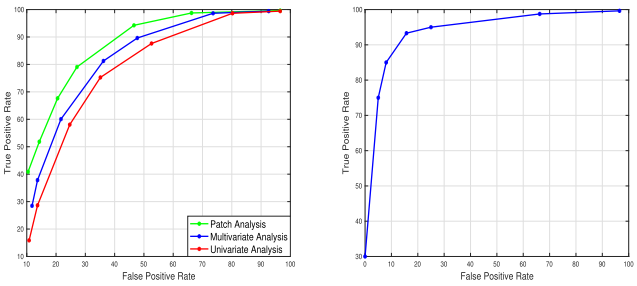
Block size	Pixel level				Video level			
	Precision (%)	Recall (%)	F1 (%)	IoU (%)	Precision (%)	Recall (%)	F1 (%)	IoU (%)
1	**76.06**	75.89	75.97	61.25	**92.86**	86.67	89.66	81.25
5	71.30	**84.77**	**77.45**	**63.20**	89.36	**93.33**	**91.30**	**84.00**
10	73.02	82.07	77.28	62.97	90.11	91.11	90.61	82.83
15	75.61	75.63	75.62	60.80	91.86	87.78	89.77	81.44
20	75.16	72.82	73.98	58.70	92.77	85.56	89.02	80.21

SURF [56]. However, these approaches lead to a high dimensional feature vector that reduces detectability of changes, especially if this vector contains irrelevant features [57]. One way to overcome this problem is to use one of dimension reduction approaches such as PCA [58], but these approaches often result in loss of relevant features. Since feature vectors are required to be relevant with small size, we experimentally observe that *mean* and *variance* are relevant features to our model. In particular, we observe that removed object traces that are exposed by the proposed spatiotemporal filter disrupt *mean* and *variance* of video frame blocks. Therefore, we believe that *mean* and *variance* are appropriate features for our model. As a result, we compute the *mean* and *variance* for each block in video frames and use them as feature vectors throughout multivariate and patch analyses.

D. Threshold Settings

Initially, we choose 10% of the dataset to tune threshold values of univariate (h_u), multivariate (h_m), and patch (h_p) analyses. The receiver operating characteristic (ROC) curves that are illustrated in Fig.6a suggest the best tradeoff between the true positive and false positive rates at pixel level for the three analyses can be achieved when $h_u = 15$, $h_m = 35$, and $h_p = 10$. Thus, we choose these threshold values based on the results of ROC curves.

We also tune threshold values for the number of consecutive frames (h_F) and areas (h_A) that are used to declare forgery at video level. We start with $h_F = 5$ with an increment of 10 frames and $h_A = 300$ with an increment of 100 pixels. The ROC curve that is shown in Fig.6b suggests the best tradeoff between the true positive and false positive rates can be achieved when $h_F = 25$ and $h_A = 500$. Thus, we selected these threshold values for all experiments.



(a) (b)

Fig. 6. ROC curves: True positive vs. false positive rates at (a) pixel level for different change thresholds using univariate, multivariate, and patch analyses, and (b) at video level using different thresholds for the number of consecutive frames (h_F) and areas (h_A).

E. Detection Results

We evaluate detection results at both pixel and video levels using sequential analysis, followed by patch analysis.

1) Results of Sequential Analysis: We evaluate both the effectiveness of (1) the univariate analysis when the block size equals one, and (2) the multivariate analysis with changes in block size.

Detection results of the uncompressed video set at pixel and video levels for different block sizes are shown in Table I. We observe that Recall and F1 values at the pixel level increase when the block size increases until they reach their largest values at the block size equals 10. Then, these values slightly decrease, which suggests that the optimal block size is 10. Similarly, Recall and F1 values at the video level follow the same pattern with better detection results because detecting only one forged second (i.e., 25 frames) is enough to declare that a video is forged as discussed in Sec.V-D.

Detection results of the compressed video set at pixel and video levels for different block sizes are shown in Table I. We observe that the largest Precision value at the pixel level occurs at the block size equals one, which indicates that the false positive rate increases when the block size increases. The largest F1 and IoU scores at both pixel and video levels happen at the block size equals 15, which suggests that the optimal block size is 15. Furthermore, we notice that detection results are still high even though videos in this video set are compressed, which indicates that the sequential analysis is robust against compressed videos.

Detection results of the low-resolution video set at pixel and video levels for different block sizes are shown in Table I(c). We observe that the smallest Precision value at both pixel and video levels occurs at the block size equals five. However, the largest Recall and F1 values at both pixel and video levels happen at the block size equals five, which suggests that the optimal block size is five. Moreover, we notice that detection results are still high even though videos in this video set have low resolutions, which indicates that the sequential analysis is also robust against lower resolution videos.

We observe that detection results at video level are better than detection results at pixel level. This result is expected because it is enough to detect a small number (i.e., 25) of

TABLE II

DETECTION RESULTS AT PIXEL AND VIDEO LEVELS OF OBJECT REMOVAL VIDEO FORGERY USING PATCH ANALYSIS WITH DIFFERENT PATCH SIZES AND DIFFERENT VIDEO SETS

(A) DETECTION RESULTS OF UNCOMPRESSED VIDEO SET.

Patch size	Pixel level				Video level			
	Precision (%)	Recall (%)	F1 (%)	IoU (%)	Precision (%)	Recall (%)	F1 (%)	IoU (%)
4	70.32	88.82	78.49	64.60	90.63	96.67	93.55	87.88
8	74.17	**89.19**	80.99	68.05	92.55	96.67	94.57	89.69
12	**74.49**	88.90	**81.06**	**68.15**	**95.65**	**97.78**	**96.70**	**93.62**
16	73.52	86.94	79.67	66.21	93.41	94.44	93.92	88.54
20	72.57	85.38	78.45	64.54	94.32	92.22	93.26	87.37

(B) DETECTION RESULTS OF COMPRESSED VIDEO SET.

Patch size	Pixel level				Video level			
	Precision (%)	Recall (%)	F1 (%)	IoU (%)	Precision (%)	Recall (%)	F1 (%)	IoU (%)
4	71.31	**87.91**	78.74	64.93	89.58	95.56	92.47	86.00
8	74.80	87.35	80.59	67.49	93.55	**96.67**	**95.08**	**90.63**
12	**75.17**	87.08	**80.69**	**67.63**	**95.51**	94.44	94.97	90.43
16	74.53	85.18	79.50	65.98	94.38	93.33	93.85	88.42
20	73.36	82.58	77.70	63.53	90.11	91.11	90.61	82.83

(C) DETECTION RESULTS OF LOW-RESOLUTION VIDEO SET.

Patch size	Pixel level				Video level			
	Precision (%)	Recall (%)	F1 (%)	IoU (%)	Precision (%)	Recall (%)	F1 (%)	IoU (%)
4	78.94	78.88	78.91	65.17	92.05	90.00	91.01	83.51
8	**79.38**	81.57	**80.46**	**67.31**	**94.38**	93.33	93.85	88.42
12	78.58	**82.38**	80.44	67.28	93.48	**95.56**	**94.51**	**89.58**
16	78.19	79.77	78.97	65.25	91.40	94.44	92.90	86.73
20	77.67	78.94	78.30	64.34	91.21	92.22	91.71	84.69

consecutive forged frames to declare forgery at video level (i.e., video forgery detection) as discussed in Sec.V-D, whereas detecting forgery at pixel level requires to detect all forged pixels, which is often significantly large, to localize forged regions in a video (i.e., video forgery localization).

In summary, we observe that the optimal block sizes of uncompressed, compressed, and low-resolution video sets are ten, fifteen, and five, respectively, because these block sizes lead to the highest detection performance (F1 score). We also consider ten as the optimal block size of compressed video set because F1 scores are almost the same at block sizes equal ten and fifteen. Therefore, the optimal block size of uncompressed video set is the same as the optimal block size of compressed video set because these video sets have the same resolution (i.e., 1280×720). The optimal block size of low-resolution video set is half of the optimal block size of uncompressed video set, which is expected because the resolution of low-resolution video set is reduced by half compared to the resolution of uncompressed video set. The highest detection performance is achieved when using the uncompressed video set, and then it slightly decreases throughout the compressed and low-resolution video sets. The overall detection performance (F1 score) of the three video sets is improved when using the multivariate analysis compared to the univariate analysis, which is one of our key contributions. We believe the reason for this improvement is that forged regions in a video are always larger than one pixel, hence applying multivariate analysis, which is based on blocks, increases detection results throughout the three video sets.

2) Results of Patch Analysis: We need to fix the block size while the patch size is varied in order to evaluate the effectiveness of the patch analysis. We notice that the optimal block size is not the same for the three video sets. However, the difference between the largest F1 score and the other F1 scores at the block size equals 10 is very small across the three video sets. Hence, we set the block size to 10 throughout the patch analysis.

Detection results of the uncompressed video set at pixel and video levels for different patch sizes are shown in Table II. We observe that the Recall value at the pixel level peaks when the patch size is eight, and then subsequently decreases, indicating that the false negative rate increases as the patch size increases. The largest F1 and IoU scores at both pixel and video levels happen at the patch size equals 12, which suggests that the optimal patch size is 12. We also notice that when the detection results at the pixel level improve, the detection results at the video level improve as well, which is expected because the pixel is a fundamental unit of videos.

Detection results of the compressed video set at pixel and video levels for different patch sizes are shown in Table II. We observe that the largest Recall value at the pixel level occurs at the patch size equals four, which indicates that the true positive rate does not improve when the patch size increases. The largest Precision value at both pixel and video levels occurs at the same patch size, which is 12. However, the largest F1 score at pixel and video levels happens at the patch sizes equal 12 and eight, respectively. Hence, the optimal patch size at pixel level is not the same as the optimal patch size at video level.

Detection results of the low-resolution video set at pixel and video levels for different patch sizes are shown in Table II(c). We observe that Precision value at both pixel and video level increases when the patch size increases until it reaches its largest value at the patch size equals eight. Then, its value slightly decreases, which indicates that the false positive rate increases when the patch size increases beyond eight. The largest F1 score at pixel and video levels happens at the patch sizes equal eight and twelve, respectively. However, at the pixel level, the difference between F1 score at patch size equals eight and twelve is very small. Hence, the optimal patch size at both pixel and video levels is 12.

As can be seen from Table II, the pixel level performance of our patch analysis approach (which is currently one of few approaches available for detection of object removal video forgery) achieves a detection rate of only 81.06%, therefore the data set employed provides a challenging framework for evaluation of the detection of object removal video forgery.

In summary, we consider that the optimal patch size of uncompressed, compressed, and low-resolution video sets is twelve because this patch size leads to the highest detection performance (F1 score). We observe that there are no significant differences between the largest F1 scores in the three video sets. Hence, our patch analysis approach is robust against compressed and lower resolution videos. We believe that there are two reasons for this robustness. First, the proposed spatiotemporal filter is able to expose traces that are left at a removed object boundary even though videos are compressed or have low resolutions. Second, compression or low resolution is applied to all video frames, hence all patches are compressed or have low resolutions. As a result, our patch analysis can distinguish between normal and anomalous patches in a video if the attack (e.g., compression) is applied to all patches. The highest detection performance is achieved

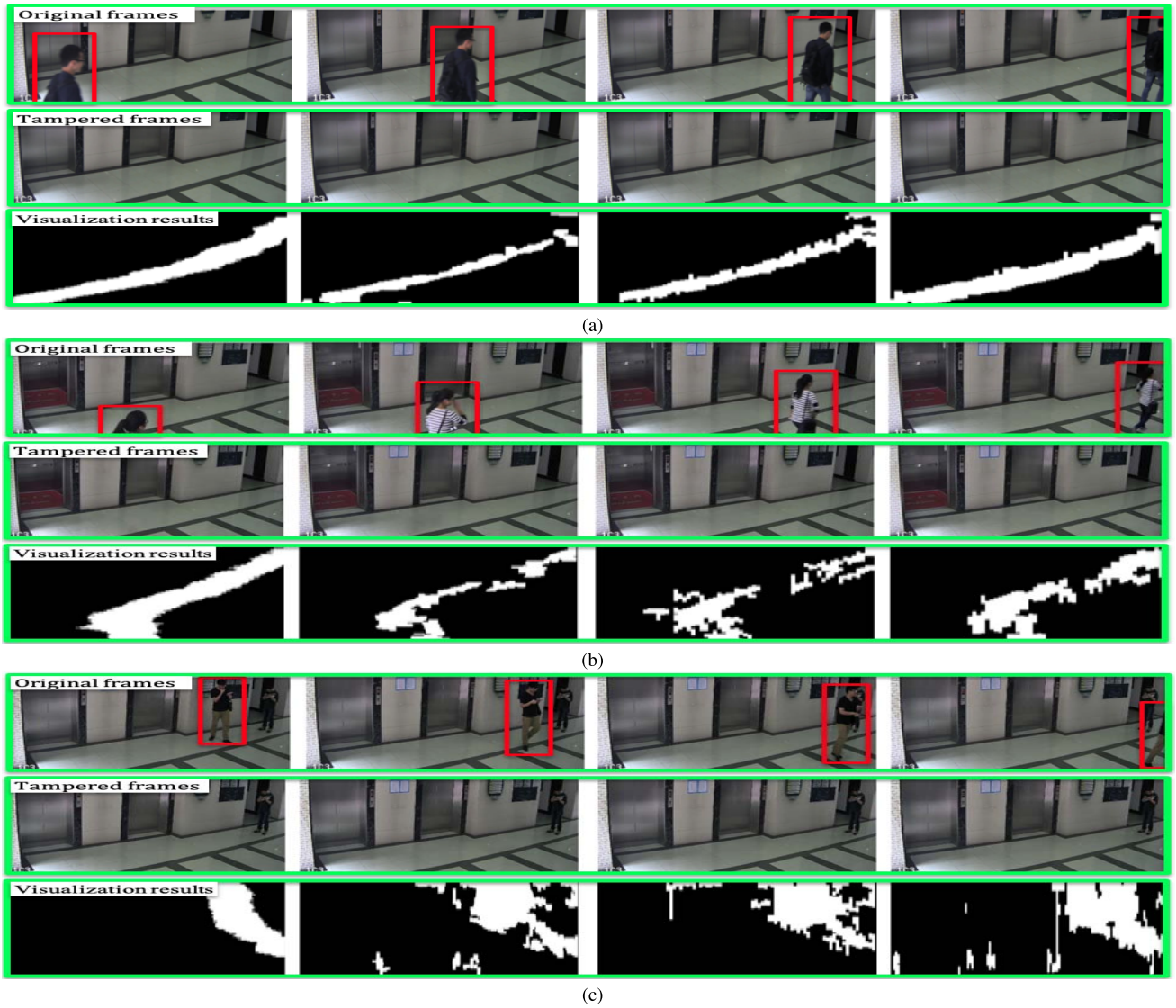
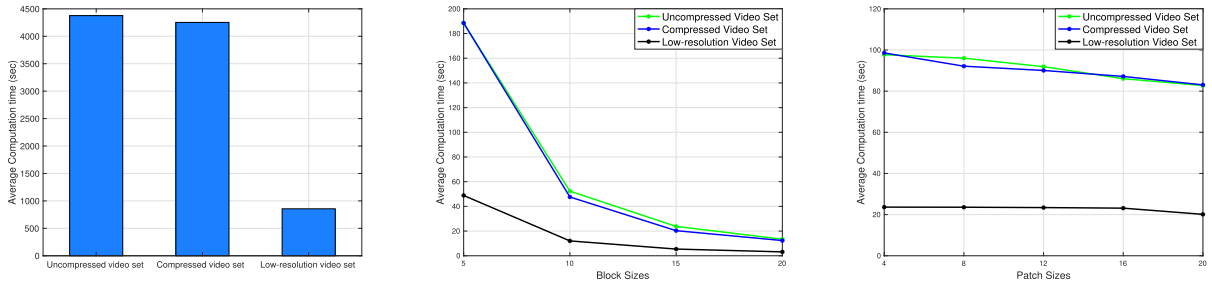


Fig. 7. Three examples of visualization results of a removed object movement using the univariate, multivariate, and patch analyses. In each example, images on the top row indicate frames from the original video; images on the middle row indicate the corresponding frames from the tampered video where the object in the red box has been removed from the scene; images on the bottom row (from left to right) indicate the ground truth of the removed object movement, the movement visualization using the univariate analysis, multivariate analysis, and patch analysis, respectively.



(a) (b) (c)

Fig. 8. Average computation time per video in seconds for the three video sets using (a) univariate analysis, (b) multivariate analysis with different block sizes, and (c) patch analysis with different patch sizes.

when using the uncompressed video set, and then it slightly decreases throughout the compressed and low-resolution video sets. The overall detection performance (F1 score) of the three video sets is further improved when using our patch analysis approach, which is another major contribution of this work. We believe the reason for this improvement is that object removal forgery always happens in temporally adjacent regions, hence applying patch analysis, which is based on

TABLE III

COMPARISON RESULTS OF OBJECT REMOVAL FORGERY DETECTION AT VIDEO LEVEL FOR OUR PATCH ANALYSIS APPROACH AND OTHER APPROACHES USING DIFFERENT VIDEO SETS

Approach	Precision (%)			Recall (%)			F1 (%)			IoU(%)		
	Uncompressed	Compressed	Low-resolution	Uncompressed	Compressed	Low-resolution	Uncompressed	Compressed	Low-resolution	Uncompressed	Compressed	Low-resolution
StatFeat [41]	82.61	72.34	78.49	84.44	75.56	81.11	83.52	73.91	79.78	71.70	58.62	66.36
STCA [44]	89.66	77.78	80.00	86.67	75.27	79.12	88.14	76.50	79.56	78.79	61.95	66.06
CompSen [43]	90.80	79.51	81.91	87.78	73.34	77.78	89.21	76.30	79.79	80.52	61.68	66.37
StegFeat [42]	**97.80**	95.45	**96.59**	**98.89**	93.34	94.45	**98.34**	94.38	**95.50**	**96.73**	89.36	**91.39**
UniSeq [4]	93.33	92.22	92.86	93.33	92.22	86.67	93.33	92.22	89.66	87.49	85.56	81.26
Our approach	95.65	**95.51**	93.48	97.78	**94.44**	**95.56**	96.70	**94.97**	94.51	93.62	**90.43**	89.58

temporally adjacent blocks, increases detection results throughout the three video sets.

3) Comparison Between Sequential and Patch Analyses: We set the block size to 10 and the patch size to 12 in this comparison because these values are optimal for block and patch sizes based on the results in Table I and Table II.

We compare forgery detection between sequential and patch analyses by plotting receiver operating characteristic (ROC) curves. Fig.6a shows ROC curves for different change thresholds (h_u, h_m, h_p) using univariate, multivariate, and patch analyses. We observe that our patch analysis outperforms univariate and multivariate analyses throughout different change thresholds, which is expected because detection results in Table II are better than detection results in Table I. We believe that our patch analysis outperforms sequential analysis because our patch analysis is based on spatiotemporal analysis, whereas sequential analysis is based on spatial analysis. Hence, our patch analysis detects object removal forgery, which is always created using temporally adjacent frames, better than sequential analysis.

We compare forgery localization between sequential and patch analyses by visualizing a movement of removed objects. Fig.7 shows three examples for visualization results of a removed object movement using univariate, multivariate, and patch analyses. We observe that our patch analysis localizes the removed object movement more accurately compared to univariate and multivariate analyses, which is expected because detection results using patch analysis at the pixel level are improved as discussed in Sec.V-E2. However, the false positive rate of patch analysis is higher than the false positive rates of univariate and multivariate analyses as shown in Fig.7c, which is expected because the largest Precision value in Table II(a) is less than the largest Precision value in Table I(a).

F. Computational Complexity

We will use the following notation throughout this section. N is the number of frames in a video, M is the number of pixels in each frame, B is the number of pixels in each block, and P is the number of blocks in each patch.

Univariate analysis detects additive changes that are associated with a pixel sequence in a video using the binary segmentation algorithm. We know that the computational complexity of the binary segmentation is $O(N\log(N))$ [51]. Therefore, the computational complexity of univariate analysis is $O(MN\log(N))$ because univariate analysis detects changes in each pixel. We also show the average computation time per video in seconds for the three video sets using univariate analysis in Fig.8a. We observe that the average computation time for low-resolution video set is much less than the average computation time for uncompressed and compressed video sets, which is expected because the resolution of this video set is reduced by half compared to the other video sets.

Multivariate analysis detects additive changes associated with feature vectors that are extracted from dividing video frames into non overlapping blocks. Each block requires $O(NB)$ computations to extract feature vectors throughout video frames and $O(N\log(N))$ computations to detect changes using the binary segmentation algorithm. As a result, the computational complexity of multivariate analysis is $O(M/B(NB + N\log(N)))$. We also show the average computation time per video in seconds for the three video sets using multivariate analysis in Fig.8b. We observe that the average computation time is exponentially reduced for the three video sets. The reason is that the multivariate analysis is performed for each block instead of each pixel, hence the average computation time is dramatically decreased when the block size increases.

Patch analysis detects anomalous patches, which are temporally adjacent blocks, throughout video frames by examining each patch in an overlapping manner with overlapping step equals one. The computational complexity of patch analysis is similar to multivariate analysis. The only difference is that calculating the log-likelihood of overlapping patches throughout video frames requires $O((N-P)N)$ computations. As a result, the computational complexity of patch analysis is $O(M/B(NB + (N - P)N))$. We also show the average computation time per video in seconds for the three video sets using patch analysis in Fig.8c. We observe that the average computation time does not significantly change when the patch size increases because our patch analysis approach is performed based on overlapping patches instead of non overlapping patches.

We conclude that by using either the multivariate or patch analysis not only improves the detection performance compared to univariate analysis as discussed in Sec.V-E, but also results in much less computational time, which is another major contribution of this work.

G. Comparison Results With Other Approaches

We compare our approach with five recent approaches [4], [41]–[44]. We refer to [4] as UniSeq, [41] as StatFeat, [42] as StegFeat, [43] as CompSen, and [44] as STCA throughout the comparison results.

Detection results at video level for our patch analysis approach and the other approaches using different video sets

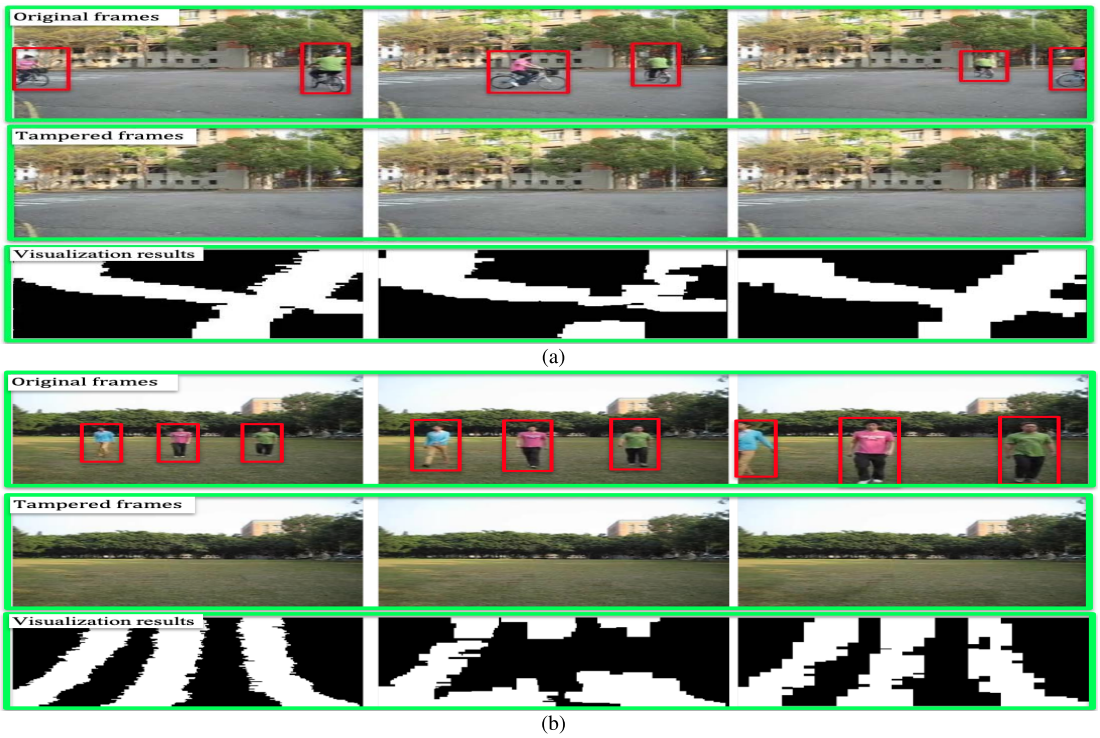


Fig. 9. Two examples of visualization results of removed objects' movement using our patch analysis approach and STCA approach. In each example, images on the top row indicate frames from the original video; images on the middle row indicate the corresponding frames from the tampered video where the objects in the red box have been removed from the scene; images on the bottom row (from left to right) indicate the ground truth of the removed objects' movement, the movement visualization using the STCA approach, and our patch analysis approach, respectively.

are shown in Table III. We set the patch size to 12 across the three video sets to have a fair comparison with the other approaches. We observe that detection results of our approach are consistent across the three video sets. We also observe that StegFeat achieves slightly better performance compared to our approach throughout uncompressed and low-resolution video sets. However, our approach outperforms all five approaches in compressed video set. This result indicates that our approach is more practical because most of the online videos are compressed. Moreover, our approach not only detects forgery but also localizes forged regions, unlike other approaches [4], [41]–[43]

Detection results at pixel level for our patch analysis approach and the STCA approach using different video sets are shown in Tables IV(a) to IV(c). We compare with the STCA approach only because the other approaches are not able to detect pixel level forgery. We observe that our approach outperforms the STCA approach throughout the three video sets. We also observe that detection results of our approach are consistent across the three video sets.

TABLE IV

COMPARISON RESULTS OF OBJECT REMOVAL FORGERY DETECTION AT PIXEL LEVEL FOR OUR PATCH ANALYSIS APPROACH AND THE STCA APPROACH USING DIFFERENT VIDEO SETS

(A) DETECTION RESULTS OF UNCOMPRESSED VIDEO SET.

Approach	Precision (%)	Recall (%)	F1 (%)	IoU (%)
STCA [44]	**78.68**	75.35	76.98	62.57
Our approach	74.49	**88.90**	**81.06**	**68.15**

(B) DETECTION RESULTS OF COMPRESSED VIDEO SET.

Approach	Precision (%)	Recall (%)	F1 (%)	IoU (%)
STCA [44]	72.70	68.26	70.41	54.33
Our approach	**75.17**	**87.08**	**80.69**	**67.63**

(C) DETECTION RESULTS OF LOW-RESOLUTION VIDEO SET.

Approach	Precision (%)	Recall (%)	F1 (%)	IoU (%)
STCA [44]	73.18	71.40	72.28	56.59
Our approach	**78.58**	**82.38**	**80.44**	**67.27**

(D) DETECTION RESULTS OF LIN'S VIDEO SET.

Approach	Precision (%)	Recall (%)	F1 (%)	IoU (%)
STCA [44]	**78.40**	72.10	75.12	60.15
Our approach	77.28	**91.87**	**83.95**	**72.33**

H. Generalization

To evaluate the generalization of our approach using different data sets and different inpainting algorithms, we use the data set that is introduced by Lin and Tsay [44]. This data set consists of 26 object removal forged videos that are generated using two inpainting algorithms: temporal copy-and-paste [59] and exemplar-based texture synthesis [60]. This data set contains forged videos with multiple removed objects as shown in Fig.9. All videos in this dataset are compressed using MPEG-4 with 3Mbps and a resolution of 320×240. We refer to this dataset as Lin's video set throughout the comparison results.

Detection results at pixel level for our patch analysis approach and the STCA approach using Lin's video set are shown in Table IV(d). We observe that STCA achieves a slightly better Precision score compared to our approach. However, our approach outperforms STCA in terms of Recall, F1, and IoU scores. This result confirms that our approach can detect and localize object removal forgery in forged videos with multiple removed objects for different data sets and inpainting algorithms.

Two examples of localization results for our patch analysis approach and the STCA approach are shown in Fig.9. We observe that our patch analysis localizes the removed objects' movement more accurately compared to STCA. In fact, our patch analysis correctly localizes three removed objects in Fig.9b. However, the STCA localizes only one removed object in Fig.9b. We believe that our patch analysis can detect and localize multiple removed objects because patch analysis detects all anomalous patches of a particular block by investigating all overlapping patches of this block as shown in Fig.5. For example, if there are two removed objects that pass through a block in video frames, then patch analysis would detect two anomalous segments for this block.

VI. Conclusion

We investigated the object removal video forgery problem, and proposed a novel approach based on sequential and patch analyses to detect video forgery and localize forged regions by visualizing a movement of removed objects. We modeled video sequences as stochastic processes, where changes in the parameters of these processes indicate a video forgery. We also modeled video sequences as a mixture model of normal and anomalous patches, with the aim to separate these patches by identifying the distribution of each patch. We evaluated detection performance at pixel and video levels, unlike most of the existing approaches that evaluated detection performance at video level only without localizing forged regions. The experimental results show that the detection performance is improved by using multivariate sequential analysis compared to univariate sequential analysis. Furthermore, our patch analysis approach not only achieves excellent detection performance with low computational complexity, but also leads to robust results against compressed and lower resolution videos.

In the future, we plan to investigate non-additive change models such as changes in covariance or correlations using the asymptotic local hypotheses. In the sequential analysis, we modeled video sequences as an additive change in scalar and multidimensional parameters. The detection results at the video level are superior, but the detection results at the pixel level can be further improved. Hence, using non-additive change models may lead to better detection performance. We also plan to extend our work to be able to detect object removal forged videos with moving backgrounds.

References

[1] M. C. Stamm, M. Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, vol. 1, pp. 167–200, 2013.

[2] S. Milani *et al.*, "An overview on video forensics," *APSIPA Trans. Signal Inf. Process.*, vol. 1, no. e2, pp. 1–18, Aug. 2012.

[3] K. Sitara and B. M. Mehtre, "Digital video tampering detection: An overview of passive techniques," *Digit. Invest.*, vol. 18, pp. 8–22, Sep. 2016.

[4] M. Aloraini, M. Sharifzadeh, C. Agarwal, and D. Schonfeld, "Statistical sequential analysis for object-based video forgery detection," *Electron. Imag.*, vol. 2019, no. 5, pp. 543-1–543-7, Jan. 2019.

[5] R. C. Pandey, S. K. Singh, and K. K. Shukla, "Passive forensics in image and video using noise features: A review," *Digit. Invest.*, vol. 19, pp. 1–28, Dec. 2016.

[6] M. A. Mizher, M. C. Ang, A. A. Mazhar, and M. A. Mizher, "A review of video falsifying techniques and video forgery detection techniques," *Int. J. Electron. Secur. Digit. Forensics*, vol. 9, no. 3, pp. 191–208, 2017.

[7] P. Johnston and E. Elyan, "A review of digital video tampering: From simple editing to full synthesis," *Digit. Invest.*, vol. 29, pp. 67–81, Jun. 2019.

[8] H. Sharma, N. Kanwal, and R. S. Batth, "An ontology of digital video forensics: Classification, research gaps & datasets," in *Proc. Int. Conf. Comput. Intell. Knowl. Economy (ICCIKE)*, Dec. 2019, pp. 485–491.

[9] H. Kaur and N. Jindal, "Image and video forensics: A critical survey," *Wireless Pers. Commun.*, vol. 112, no. 2, pp. 1281–1302, May 2020.

[10] A. Wary and A. Neelima, "A review on robust video copy detection," *Int. J. Multimedia Inf. Retr.*, vol. 8, no. 2, pp. 61–78, Jun. 2019.

[11] J. Chao, X. Jiang, and T. Sun, "A novel video inter-frame forgery model detection scheme based on optical flow consistency," in *Proc. Int. Workshop Digit. Forensics Watermarking 2012*. Berlin, Germany: Springer, 2013, pp. 267–281.

[12] Y. Wu, X. Jiang, T. Sun, and W. Wang, "Exposing video inter-frame forgery based on velocity field consistency," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 2674–2678.

[13] P. Bestagini, S. Battaglia, S. Milani, M. Tagliasacchi, and S. Tubaro, "Detection of temporal interpolation in video sequences," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 3033–3037.

[14] M. C. Stamm, W. S. Lin, and K. J. R. Liu, "Temporal forensics and anti-forensics for motion compensated video," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 4, pp. 1315–1329, Aug. 2012.

[15] Y. Su, J. Zhang, and J. Liu, "Exposing digital video forgery by detecting motion-compensated edge artifact," in *Proc. Int. Conf. Comput. Intell. Softw. Eng.*, Dec. 2009, pp. 1–4.

[16] J. Yang, T. Huang, and L. Su, "Using similarity analysis to detect frame duplication forgery in videos," *Multimedia Tools Appl.*, vol. 75, no. 4, pp. 1793–1811, Feb. 2016.

[17] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting duplication," in *Proc. 9th Workshop Multimedia Secur.*, 2007, pp. 35–42.

[18] V. K. Singh, P. Pant, and R. C. Tripathi, "Detection of frame duplication type of forgery in digital video using sub-block based features," in *Proc. Int. Conf. Digital Forensics Cyber Crime*. Seoul, South Korea: Springer, 2015, pp. 29–38.

[19] Z. Zhang, J. Hou, Q. Ma, and Z. Li, "Efficient video frame insertion and deletion detection based on inconsistency of correlations between local binary pattern coded frames," *Secur. Commun. Netw.*, vol. 8, no. 2, pp. 311–320, Jan. 2015.

[20] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double quantization," in *Proc. 11th ACM Workshop Multimedia Secur.*, 2009, pp. 39–48.

[21] D. Liao, R. Yang, H. Liu, J. Li, and J. Huang, "Double H. 264/AVC compression detection using quantized nonzero AC coefficients," *Proc. SPIE*, vol. 7880, Feb. 2011, Art. no. 78800Q.

[22] P. He, X. Jiang, T. Sun, and S. Wang, "Detection of double compression in MPEG-4 videos based on block artifact measurement," *Neurocomputing*, vol. 228, pp. 84–96, Mar. 2017.

[23] T. Sun, W. Wang, and X. Jiang, "Exposing video forgeries by detecting MPEG double compression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 1389–1392.

[24] H. Ravi, A. V. Subramanyam, G. Gupta, and B. A. Kumar, "Compression noise based video forgery detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 5352–5356.

[25] K. Sowmya and H. Chennamma, "A survey on video forgery detection," *Int. J. Comput. Eng. Appl.*, vol. 9, no. 2, pp. 17–27, 2015.

[26] M. A. Bagiwa, A. W. A. Wahab, M. Y. I. Idris, S. Khan, and K.-K.-R. Choo, "Chroma key background detection for digital video using statistical correlation of blurring artifact," *Digit. Invest.*, vol. 19, pp. 29–43, Dec. 2016.

[27] Y. Su, Y. Han, and C. Zhang, "Detection of blue screen based on edge features," in *Proc. 6th IEEE Joint Int. Inf. Technol. Artif. Intell. Conf.*, Aug. 2011, pp. 469–472.

[28] J. Xu, Y. Yu, Y. Su, B. Dong, and X. You, "Detection of blue screen special effects in videos," *Phys. Procedia*, vol. 33, pp. 1316–1322, 2012.

[29] L. D'Amiano, D. Cozzolino, G. Poggi, and L. Verdoliva, "A patchmatch-based dense-field algorithm for video copy-move detection and localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 3, pp. 669–682, Mar. 2019.
[30] Y. Liu, T. Huang, and Y. Liu, "A novel video forgery detection algorithm for blue screen compositing based on 3-stage foreground analysis and tracking," *Multimedia Tools Appl.*, vol. 77, no. 6, pp. 7405–7427, Mar. 2018.
[31] V. Conotter, J. F. O'Brien, and H. Farid, "Exposing digital forgeries in ballistic motion," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 1, pp. 283–296, Feb. 2012.
[32] D. D'Avino, D. Cozzolino, G. Poggi, and L. Verdoliva, "Autoencoder with recurrent neural networks for video forgery detection," *Electron. Imag.*, vol. 2017, no. 7, pp. 92–99, Jan. 2017.
[33] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, "Video inpainting of complex scenes," *SIAM J. Imag. Sci.*, vol. 7, no. 4, pp. 1993–2019, Jan. 2014.
[34] M. Ebdelli, O. Le Meur, and C. Guillemot, "Video inpainting with short-term windows: Application to object removal and error concealment," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3034–3047, Oct. 2015.
[35] R. Xu, X. Li, B. Zhou, and C. C. Loy, "Deep flow-guided video inpainting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3723–3732.
[36] J. Zhang, Y. Su, and M. Zhang, "Exposing digital video forgery by ghost shadow artifact," in *Proc. 1st ACM Workshop Multimedia Forensics*, 2009, pp. 49–54.
[37] C.-C. Hsu, T.-Y. Hung, C.-W. Lin, and C.-T. Hsu, "Video forgery detection using correlation of noise residue," in *Proc. IEEE 10th Workshop Multimedia Signal Process.*, Oct. 2008, pp. 170–174.
[38] S. Saxena, A. V. Subramanyam, and H. Ravi, "Video inpainting detection and localization using inconsistencies in optical flow," in *Proc. IEEE Region 10 Conf. (TENCON)*, Nov. 2016, pp. 1361–1365.
[39] M. A. Bagiwa, A. W. A. Wahab, M. Y. I. Idris, and S. Khan, "Digital video inpainting detection using correlation of hessian matrix," *Malaysian J. Comput. Sci.*, vol. 29, no. 3, pp. 179–195, 2016.
[40] Y. Yao, Y. Cheng, and X. Li, "Video objects removal forgery detection and localization," in *Proc. Nicograph Int. (NicoInt)*, Jul. 2016, p. 137.
[41] C. Richao, Y. Gaobo, and Z. Ningbo, "Detection of object-based manipulation by the statistical features of object contour," *Forensic Sci. Int.*, vol. 236, pp. 164–169, Mar. 2014.
[42] S. Chen, S. Tan, B. Li, and J. Huang, "Automatic detection of object-based forgery in advanced video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 11, pp. 2138–2151, Nov. 2016.
[43] L. Su, T. Huang, and J. Yang, "A video forgery detection algorithm based on compressive sensing," *Multimedia Tools Appl.*, vol. 74, no. 17, pp. 6641–6656, Sep. 2015.
[44] C.-S. Lin and J.-J. Tsay, "A passive approach for effective detection and localization of region-level video forgery with spatio-temporal coherence analysis," *Digit. Invest.*, vol. 11, no. 2, pp. 120–140, Jun. 2014.
[45] M. J. Veth, "Fusion of imaging and inertial sensors for navigation," Air Force Inst. Technol. School Eng. Manage., Air Force Base, OH, USA, Tech. Rep. AFIT/DS/ENG/06-09, 2006.
[46] E. Hecht, *Optics*. Reading, MA, USA: Addison-Wesley, 2002.
[47] A. J. Lingg, E. Zelnio, F. Garber, and B. D. Rigling, "A sequential framework for image change detection," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2405–2413, May 2014.
[48] D. Lelescu and D. Schonfeld, "Statistical sequential analysis for real-time video scene change detection on compressed multimedia bitstream," *IEEE Trans. Multimedia*, vol. 5, no. 1, pp. 106–117, Mar. 2003.
[49] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-3, no. 4, pp. 532–540, Apr. 1983.
[50] M. Basseville *et al.*, *Detection of Abrupt Changes: Theory and Application*, vol. 104. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
[51] J. Bai, "Estimating multiple breaks one at a time," *Econ. Theory*, vol. 13, no. 3, pp. 315–352, Jun. 1997.
[52] E. Eskin, "Anomaly detection over noisy data using learned probability distributions," in *Proc. Int. Conf. Mach. Learn.*, 2000, pp. 255–262.
[53] P.-N. Tan, *Introduction to Data Mining*. New Delhi, India: Pearson, 2018.
[54] G. Qadir, S. Yahaya, and A. T. Ho, "Surrey University library for forensic analysis (SULFA) of video content," in *Proc. IET Conf. Image Process.*, 2012, pp. 1–6.
[55] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 1150–1157.
[56] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2006, pp. 404–417.
[57] C. Alippi, G. Boracchi, D. Carrera, and M. Roveri, "Change detection in multivariate datastreams: Likelihood and detectability loss," 2015, *arXiv:1510.04850*. [Online]. Available: http://arxiv.org/abs/1510.04850
[58] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer, 2002.
[59] K. A. Patwardhan, G. Sapiro, and M. Bertalmío, "Video inpainting under constrained camera motion," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 545–553, Feb. 2007.
[60] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

Mohammed Aloraini (Member, IEEE) received the B.S. degree in electrical engineering from Qassim University in 2011 and the M.S. degree in electrical and computer engineering from the University of Illinois at Chicago in 2014, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His current research interests include image processing, multimedia forensics, and information security.

Mehdi Sharifzadeh (Member, IEEE) received the B.S. degree in electrical engineering from the Sharif University of Technology in 2012. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Illinois at Chicago, where he is also a Researcher. His current research is in image steganography. Along with his main research topic, he works on deep neural networks and problems in machine learning and computer vision.

Dan Schonfeld (Senior Member, IEEE) received the B.S. degree in electrical engineering and computer science from the University of California at Berkeley in 1986 and the M.S. and Ph.D. degrees in electrical and computer engineering from Johns Hopkins University, Baltimore, MD, USA, in 1988 and 1990, respectively. In 1990, he joined the University of Illinois at Chicago, where he is currently a Professor with the Department of Electrical and Computer Engineering. He has authored over 120 technical articles in various journals and conferences. His current research interests are in multidimensional signal processing, image and video analysis, computer vision, and genomic signal processing. He was the co-author of articles that won the Best Student Paper Awards in Visual Communication and Image Processing in 2006 and the IEEE International Conference on Image Processing 2006 and 2007. He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He serves as an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.