This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2929855, IEEE Transactions on Circuits and Systems for Video Technology



# Efficient Privacy-Preserving Anomaly Detection and Localization in Bitstream Video

Jianting Guo, *Student Member, IEEE,* Peijia Zheng, *Member, IEEE,* and Jiwu Huang, *Fellow, IEEE*

***Abstract*—In cloud computing, videos may be in an encrypted format to protect privacy. Therefore, encrypted video processing is an important application in secure cloud computing. In this paper, we focus on parameter estimation and anomaly detection in an encrypted video bitstream. By analyzing the common properties of video encoding frameworks and the format-compliant encryption schemes, we propose an anomaly detection scheme for encrypted video bitstream with format-compliant encryption. From the encrypted bitstream, we extract three types of complementary features, i.e., the macroblock sizes, the macroblock partitions, and the motion vector difference magnitude, and then propose a method to combine these three features. The proposed detection and localization scheme does not involve video decryption, full decompression, or an interactive protocol, which makes it efficient. Our scheme is also compatible with different video encryption methods. To accelerate the running time, we develop a parallel implementation for our scheme. The experimental results show that our method achieves good running time and detection rate performance.**

***Index Terms*—Signal Processing in the Encrypted Domain, Encrypted Video Processing, Privacy-preserving, Anomaly Detection, Partial Encryption, Cloud Computing.**

## I. Introduction

With the advent of the big data era, large-scale multimedia data are being generated rapidly, particularly video data, such as surveillance videos, sport video, and video databases. For a common resource-constrained data owner, storing and processing the enormous amount of video data is becoming unaffordable. Fortunately, with cloud computing, the data owner can outsource the expensive data storage as well as the sophisticated video data processing to the cloud server and enjoy conveniently personalized computing services. Due to security and privacy concerns, video data in the cloud are very likely to be encrypted [1]. The employment of cryptographic tools on video data makes the subsequent video processing approaches complicated. Signal processing in the encrypted domain (SPED) [2], which is able to manipulate data without decrypting, will be helpful to the development of privacy-preserving computing in the cloud. For example, in a mobile media cloud system, the server can perform processing and analysis directly on ciphertext data in response to user requests by relying on SPED techniques.

Many reports on SPED have recently been published to develop privacy-preserving applications [3]–[6]. Moreover, work on encrypted video processing has also been reported, i.e., encrypted video bitstream data hiding [7] and encrypted video motion detection [8]. By relying on some prior assumptions, [8] extracted features from motion vector difference (MVD) and performed motion detection on the encrypted H.264 video in serial mode. However, to the best of our knowledge, there are no reports on anomaly detection in an encrypted video bitstream. Considering the importance of anomaly detection in video surveillance, it is meaningful to perform anomaly detection on encrypted video for privacy protection.

Anomalous event detection is an important component of intelligent video surveillance. The anomalies that we focus on here are abnormal motion with sudden actions compared to most of the other motion in the video, e.g., running, throwing, and dispersing. Generally, the input of conventional video anomaly detection schemes is plaintext videos (i.e., videos that are not encrypted). The existing schemes can be grouped into two categories, frame-based schemes and bitstream-based schemes. For the frame-based schemes, there are various traditional methods such as object trajectory of normal events [9], spatio-temporal gradients [10], mixture of dynamic textures [11], hierarchical Bayesian model [12], sparse representations of events [13], scene-parsing approach [14], etc. Recently, motivated by the impressive success of deep learning, researchers began to propose new video anomaly detection schemes with a deep network, including Deep-cascade [15], AVID [16], ALOCC [17], CVPRW2015 [18], Deep-anomaly [19], and GMFC-VAE [20]. However, it is still very difficult to make the deep learning methods work for the encrypted video. Other restrictions on deep learning are that the severe demands on hardware devices, the heavy relying on a large amount of labeled data, and the high probability of overfitting/underfitting for the model [21]. As for the bitstream-based schemes, they generally have higher efficiency. By using the magnitudes of the motion vectors, Biswas *et al*. [22] proposed an anomaly detection scheme in H.264 video bitstreams. In [23], the authors improved the work by adding the orientation information of the motion vectors into the feature set. In [24], Kiryati *et al*. proposed an abnormal motion detection algorithm in compressed video

Jianting Guo is with the School of Data Science and Computer, Sun Yat-Sen University, Guangzhou 510006, China (guojting@mail2.sysu.edu.cn)

Peijia Zheng (*corresponding author*) is with the School of Data Science and Computer and the Guangdong Key Lab of Information Security, Sun Yat-Sen University, Guangzhou 510006, China (zhpj@mail.sysu.edu.cn)

Jiwu Huang is with the Guangdong Key Laboratory of Intelligent Information Processing and Key Laboratory of Media Security, Shenzhen University, Shenzhen 518060, China, and also with the Shenzhen Institute of Artificial Intelligence and Robotics for Society and Peng Cheng Laboratory, Shenzhen, China. (e-mail: jwhuang@szu.edu.cn)

This work was supported in part by the NSFC (61502547, U1636202), Guangdong Natural Science Foundation under Grant 2015A030310319, Opening Project of GuangDong Province Key Laboratory of Information Security Technology (Grant No. 2017B030314131), Guangdong R&D Program in Key Areas (2019B010139003), and Shenzhen R&D Program (JCYJ20160328144421330).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2929855, IEEE Transactions on Circuits and Systems for Video Technology



bitstreams by deriving the total motion, regional information, and directional information from the motion vectors. In [25], an anomaly detection scheme is proposed in H.264 videos by using the sparse represented histogram of oriented motion vectors. To perform traffic abnormal event detection in HEVC, Li *et al*. [26] proposed a compressed-domain feature that uses motion vectors, coding unit, and prediction unit modes.

The effectiveness and feasibility of the existing conventional anomaly detection schemes generally rely on the available pixel values or bitstream parameters. However, this is not the case in privacy-preserving anomaly detection. The necessary inputs of the existing anomaly detection algorithms are encrypted in most of the practical video encryption schemes. For example, the required pixel values in the scheme [27] and the MV directions employed in the scheme [23] are encrypted via practical video encryption [7]. Therefore, the design of a practical privacy-preserving anomaly detection scheme under the constraints that many important input data or parameters are unavailable is challenging. The problem becomes even more difficult when the input is an encrypted video bitstream rather than an encrypted frame sequence.

In this paper, we attempt to solve this problem by performing anomaly detection in encrypted video bitstreams. Compared with the frame-based methods, bitstream-based algorithms generally have the advantages of low storage, high efficiency and wide flexibility. Based on format-compliant video encryption, we propose a privacy-preserving anomaly detection protocol that can be run non-interactively with the video owner. Specifically, in the proposed protocol, we provide estimates of the macroblock (MB) size (in bits), the MB partition, and the magnitude of the motion vector difference (MVD) based on the video structure information. We design the feature-extraction algorithms for the frame-level detector and the intra-frame localization method. For applications that require rapid response, we can use only the frame-level detector. If higher detection accuracy is required, we can combine the frame-level detector and the intra-frame localization method to obtain exact localization information of the abnormal regions. After feature extraction, we train a model to conduct anomaly detection. To reduce the running time, we propose several parallel strategies to accelerate the computations of feature extraction, detection, and localization. The contributions of this paper are as follows.

1. We propose a novel non-interactive privacy-preserving video anomaly detection and localization scheme, which is not restricted to a specified video encryption and can work with many types of format-compliant video encryptions.
2. We combine three types of complementary feature information from the video bitstream, which makes our detection performance on encrypted video bitstreams achieve a satisfactory performance in our experiments.
3. The proposed algorithms are very efficient because all the feature extraction, model training, and anomaly detection are performed directly in the video bitstream. We also propose parallel implementations with a high speedup ratio.

The remainder of this paper is organized as follows. In Section II, we provide the problem statement. Section III presents our features information in encrypted video bitstreams. We describe the details of the proposed scheme, as well as its parallelization implementation, in Section IV. In Section V, we conduct experiments of the proposed scheme. Section VI provides more discussions. Finally, we conclude our paper in Section VII.

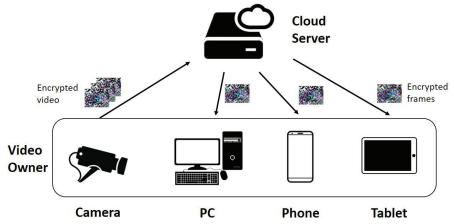



Fig. 1: System model.

## II. Problem Statement

### A. System Model

In this paper, we consider a non-interactive video anomaly detection system by outsourcing data storage and computation to the cloud. Two parties are included in the considered system, i.e., the video data owner $\mathfrak{O}$ and the cloud server $\mathfrak{S}$. $\mathfrak{O}$ may simultaneously use many devices, such as surveillance cameras, personal computers, mobile phones, and tablets. A simple sketch of our system model is shown in Fig. 1. In this application scenario, we assume that the data owner $\mathfrak{O}$ has a large volume of sensitive video data but is resource-constrained with respect to both storage and computation resources. $\mathfrak{O}$ would like to outsource both video data storage and the task of anomaly detection to the cloud server to be relieved from maintaining a local video database and interacting with the database users online.

Almost all video data are transmitted and stored in the form of a compressed bitstream. To avoid private or sensitive content leakage to the cloud, $\mathfrak{O}$ prefers to store encrypted video, rather than clear video data, on the cloud storage server. Therefore, in this paper, we adopt partial video encryption and focus on the privacy protection of video content rather than the security of every bit.

When an anomaly event is detected, $\mathfrak{S}$ extracts the encrypted anomaly frames, locates the anomaly regions in the encrypted frames, and sends these encrypted frames to $\mathfrak{O}$ with some network abstraction layer unit (NALU) parameters. $\mathfrak{O}$ decrypts the received encrypted frames with the private decryption key and obtains the abnormal frames with the located abnormal regions.

### B. Threat Model

The proposed protocol is implemented with a single server and is easy to deploy in practice. Throughout this paper, we adopt the semi-honest security setting. Considering the original data are videos, we focus on privacy-protection issues. Specifically, we assume that the cloud server $\mathfrak{S}$ follows the protocol but attempts to learn additional private or sensitive content, such as human faces and card security codes, from the encrypted video and the exchanged messages. As discussed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2929855, IEEE Transactions on Circuits and Systems for Video Technology



above, we use partial video encryption to ensure the privacy protection of the video data. Generally, video data are compressed with a specified format. To the best of our knowledge, nearly all practical video bitstream encryption methods are partial encryptions [28], which are popular in multimedia encryption due to their advantage of maintaining a balance between privacy and convenience. Therefore, the adoption of partial video encryption to protect video data is a reasonable choice. More details about privacy protection with partial video encryption can be found in [28], [29]. With the proposed protocols, $\mathfrak{S}$ can detect anomaly frames and locate anomaly regions in encrypted frames. However, without the decryption keys, $\mathfrak{S}$ cannot deduce sensitive and private information about the individuals in the video, for example, learning the exact human faces or obtaining exact card security codes.

## III. Abnormal Motion Information Estimation from Encrypted Video Bitstreams

We consider the extraction of abnormal information from encrypted bitstreams using the format-compliant video encryption. In our scheme, we use three types of estimated values obtained from the bitstream structure and codeword structure, i.e., the data size of the macroblock (in bits), the macroblock (MB) partition mode, and the magnitude of motion vector difference (MVD).

### A. Preliminary of H.264 Syntax

In video compression, a video frame is compressed using different algorithms. An I-frame (intra-coded picture), is a self-containing frame that does not need references to other frames. A P-frame (predicted picture) uses the reference to previous I- or P-frame, and will hold only the changes in the image from the previous frame. In H.264/AVC compression, the video frame is divided into several non-overlapping slices. A slice is a spatially distinct region of a frame that is encoded separately from any other region in the same frame. The entropy encoded data of the slices will constitute the bitstream of the compressed video bitstream. We can consider the entropy encoded data of the slice as a bit sequence. For any bit in the bit sequence, we can calculate the offset of this bit from the beginning of the bit sequence. For convenience, we call this offset bit_offset, which is denoted by $\mathcal{O}(\cdot)$. The entropy data of one slice begins with the slice head (denoted by **SH**), and then followed by some encoded macroblock data and some skip indication that denoting the number of skipped macroblocks. A skipped macroblock is a macroblock for which no information is sent to the decoder - i.e. no coded coefficients, no header and no prediction information. In the reference software JM, the skip indication is denoted by the syntactic element skip_run, which is denoted by $\mathbf{SR}_i$. For example, in the original video bitstream, a slice data may be represented as

$$\{\mathbf{SH}, \mathbf{MB}_1, \mathbf{MB}_2, \mathbf{SR}_3, \cdots, \mathbf{MB}_i, \cdots, \mathbf{MB}_n\},$$

where $\mathbf{MB}_i$ denotes the entropy encoded data of the $i$-th macroblock in the slice. We refer to [30] for more details of H.264 syntax.

### B. Macroblock Size

Video encoding frameworks usually adopt prediction and compensation for compression. Most of the background and the normal contents are predicted accurately, resulting in MBs with a small size (a few bits). Compared to normal motion, anomalous motion requires more bits in the video bitstream because it is "unexpected" and usually implies rapid motion. We make full use of this observation in our anomaly detection scheme.

In Fig. 2, we show an example of the comparison of normal and anomaly frames. We can see that the anomaly regions are brighter than the normal regions, which means that an anomaly uses a larger number of bits within the bitstream. The influence of an anomaly on the data size is shown in Fig.3. The bar graphs on the left show the data size of the P-frames in two encrypted videos. We can see that the data size changes rapidly when an anomaly occurs. In the encrypted bitstream, the slice data can be represented as

$$\{\mathbf{SH}, \mathbf{MB}_1^*, \mathbf{MB}_2^*, \mathbf{SR}_3, \cdots, \mathbf{MB}_i^*, \cdots, \mathbf{MB}_n^*\},$$

where $\mathbf{MB}_1^*$ denotes the corresponding ciphertexts of the original $\mathbf{MB}_i$. We calculate the data size of $\mathbf{MB}_i$ as

$$s_i = \begin{cases} 0, & \mathbf{MB}_i \text{ is skipped}, \\ \mathcal{O}(\mathbf{SR}_{i+1}) - \mathcal{O}(\mathbf{MB}_i^*), & \mathbf{MB}_{i+1} \text{ is skipped}, \\ \mathcal{O}(\mathbf{MB}_{i+1}^*) - \mathcal{O}(\mathbf{MB}_i^*), & \text{otherwise}, \end{cases} \quad (1)$$

where $\mathcal{O}(\mathbf{MB}_i)$ denotes the offset of the first bit of $\mathcal{O}(\mathbf{MB}_i)$ in the bit sequence of the slice. Note that $\mathbf{SR}_i$ can represent more than one skipped MB; the data sizes of these skipped MBs are set to 0.

### C. Macroblock Partitions

Relying on only the MB size $\{s_i\}$ is not sufficient for good abnormal detection in different situations. It is difficult to distinguish normal and abnormal in the case of I-frames since the data sizes of MBs in I-frames are generally larger than those in P-frames. Moreover, in Fig. 3, we can see that not all the rapid increases in MB size are caused by an anomaly, especially in the situation where many objects undergo normal motion in a single frame.

Macroblock partition information can also be extracted from the encrypted video bitstream. In H.264/AVC, a $16 \times 16$ MB can be divided into several sub-MBs. MBs that contain more details are divided into more sub-MBs. According to the characteristics of video coding, the sub-MBs of an abnormal region in the P-frame are more likely to have a small size. We show the MB partitions of an example frame in Fig. 4. We can see that the anomaly motion has a positive effect on the number of partitions.

In our scheme, the partition level of a $16 \times 16$ MB corresponds to the number of its sub-MBs. For example, the partition level is 16 for an MB that is segmented into sixteen $4 \times 4$ sub-MBs. If a $16 \times 16$ MB is divided into two $16 \times 8$ sub-MBs, the partition level is 2. Specifically, the partition level of $\mathbf{MB}_i$ is given as

$$l_i = \mathcal{P}(\mathbf{MT}_i, \mathbf{ST}_i), \quad (2)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2929855, IEEE Transactions on Circuits and Systems for Video Technology



(a) Normal frame (b) Abnormal frame (c) Abnormal frame

Fig. 2: Estimation of MB size (bits). (a) Example of a normal frame. (b) Abnormal frame of a man running. (c) Abnormal frame of a man walking on a wrong path (on the grass). The image on the left is the encrypted frame, and the corresponding estimation image from the MB size is on the right.

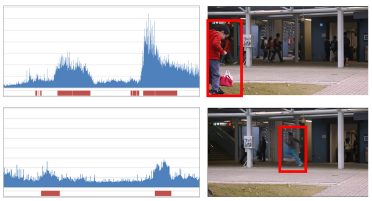


Fig. 3: Influence of anomalies on data size. The figures in the left column are the data sizes of several consecutive frames of the encrypted video and the corresponding annotation of an anomaly (marked in red on the bottom). The examples of the anomalies are shown on the right.

TABLE I: PL(**MT**, **ST**): Partition level and mb_type. I4 × 4 and I16 × 16 represent a 4 × 4 macroblock and a 16 × 16 macroblock that are encoded with in-frame prediction mode, respectively.

| MB Type (**MT**) | Slice Type (**ST**) | Partition | Level |
|---|---|---|---|
| - | P | Skip | 0 |
| 0 | P | 16 × 16 | 1 |
| 1 | P | 16 × 8 | 2 |
| 2 | P | 8 × 16 | 2 |
| 3,4 | P | 8 × 8 | 4-16 |
| 5 | P | I4 × 4 | 20 |
| 6-29 | P | I16 × 16 | 18 |
| 0 | I | I4 × 4 | 20 |
| 1-24 | I | I16 × 16 | 18 |

where $\mathbf{MT}_i$ is the MB type of $\mathbf{MB}_i$, $\mathbf{ST}_i$ is the slice type, and $\mathcal{P}(\cdot,\cdot)$ is the map outputting the partition level, which is defined as shown in Table I. Note that for $\mathbf{MT}_i = 3, 4$ and $\mathbf{ST}_i =$"P", the partition mode is "8×8". In this case, $\mathbf{MB}_i$ can be further divided into several sub-blocks. The value ranges from 4 to 16 according to the number of sub-blocks.

### *D. Motion Vector Difference*

MV is a feature that is widely used in many anomaly detection schemes [22], [24]. However, since MVs are encrypted in the encrypted video, conventional MV feature extraction algorithms are not suitable for our scheme. We propose a new MV feature extraction method via parameter estimation.

In the video bitstream, the MV is composed of two parts, the predicted MV and the MVD. The MVD is the difference between the current MV and the predicted MV. The magnitude of the MVD reflects a degree of motion information. In our application scenario, the predicted MV in the encrypted bitstream is scrambled. Although the MVD is also encrypted, the length of the codeword is kept the same to make the encrypted bitstream format compliant with the video decoder. According to the entropy coding used in the video coding framework, the codeword length corresponds to the size of the values (as shown in Table II).

Hence, we extract the MV-related feature by estimating the magnitude of the encrypted MVDs from the codeword length.

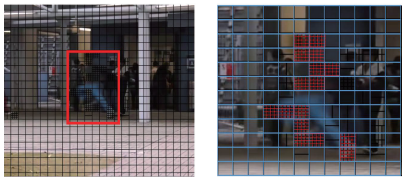


Fig. 4: Estimation from the MB partition. An example to show that anomaly motion region has more macroblock partitions. The explanation of this figure is detailed in Section VI-B.

TABLE II: MVDs and codewords.

| MVD | codeword | Group ID | MVD | codeword | Group ID |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 5 | 0001010 | 3 |
| 1 | 010 | 1 | -5 | 0001011 | |
| -1 | 011 | | 6 | 0001100 | |
| 2 | 00100 | 2 | -6 | 0001101 | |
| -2 | 00101 | | 7 | 0001110 | |
| 3 | 00110 | | -7 | 0001111 | |
| -3 | 00111 | | ... | ... | |
| 4 | 0001000 | 3 | | | |
| -4 | 0001001 | | | | |

For example, if the codeword length of an encrypted MVD is 7, then the possible absolute value of this MVD ranges from 4 to 7, as shown in Table II. Suppose that all possible values appear with equal probability; then, the MVD can be estimated as the mathematical expectation, i.e., 5.5. Since 5.5 is not included in Table II, we can use 5 as the estimated value. Therefore, the estimated value for detection would be "5". We can formulate this estimation as

$$\mathcal{D}(M) = \begin{cases} 0, & M = 1 \\ \sum\limits_{m=3,5,\ldots}^{M-2} 2^{\frac{m-3}{2}} + \left\lfloor 2^{\frac{M-5}{2}} \right\rfloor + 1, & \text{otherwise}, \end{cases} \tag{3}$$

where $\lfloor\cdot\rfloor$ is the floor function, and $M$ denotes the input of $\mathcal{D}(\cdot)$, i.e., the bit length of the codeword. The first term $\sum_{m=3,5,\ldots}^{M-2} 2^{\frac{m-3}{2}}$ denotes the initial value of the current group where $M$ locates. The second term $\lfloor 2^{\frac{M-5}{2}} \rfloor$ means the expected value of current group. The third term 1 is added to keep the consistence with the EG0 coding of signed numbers. Taking the same example that the codeword length is 7, we can also obtain that $\mathcal{D}(7) = 3 + 1 + 1 = 5$.

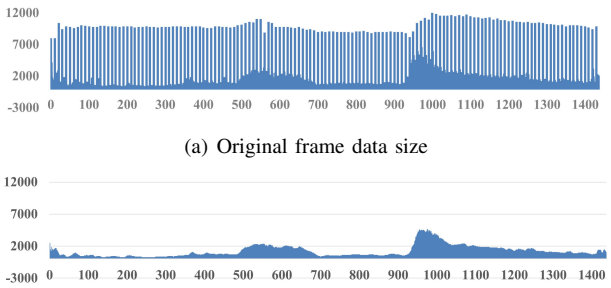
If we denote the encrypted MVD as $\mathbf{MVD}^*$, then we have $\mathbf{MVD}^* = \left[\mathbf{MVD}^*_x, \mathbf{MVD}^*_y\right]$, where $\mathbf{MVD}^*_{x_i}$ and $\mathbf{MVD}^*_{y_i}$ are the $x$-axis and $y$-axis components of $\mathbf{MVD}^*$, respectively. The energy of $\mathbf{MVD}^*_i$ can then be evaluated as

$$v_i = \mathcal{D}\left(\mathcal{L}\left(\mathbf{MVD}^*_{x_i}\right)\right)^2 + \mathcal{D}\left(\mathcal{L}\left(\mathbf{MVD}^*_{y_i}\right)\right)^2, \tag{4}$$

where $\mathcal{L}(\cdot)$ is the bit length of the component.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2929855, IEEE Transactions on Circuits and Systems for Video Technology

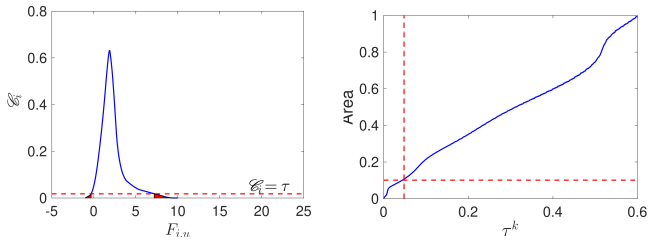





(a) Original frame data size

(b) After median filtering

Fig. 5: Preprocessing of the frame data size.




(a) (b)

Fig. 6: (a) An example of anomaly detection with only one feature. (b) An optional strategy: determining $\tau^k$ with $\theta$. The blue curve denotes the function of $\tau^k$, i.e., $\texttt{Area}\left(\{\mathscr{C}_i < \tau\}\right) \leq \theta$.

## IV. Proposed Anomaly Detection and Localization Schemes

The proposed anomaly detection scheme consists of two levels, the frame-level method and the localization method. We can use the frame-level method to detect abnormal frames and then utilize the localization method to locate abnormal regions.

### A. Anomaly Detection at the Frame Level

*1) Feature Extraction:* In frame-level detection, we first extract the MB data size, the partition level, and the MVD magnitude at the frame level.

Suppose that $s_{ij}$ is the MB data size of $\mathbf{MB}_j$ in the $i$-th frame, and $\mathbf{s}_i = \{s_{i0}, s_{i1}, \ldots\}$ denotes the set of $s_{ij}$. We use $\mathbb{J}_i$ to denote the set of all MB addresses in the $i$-th frame. In each frame, we obtain the frame data size by computing the energy of $\mathbf{s}_i$ as

$$\mathbb{E}\left(\mathbf{s}_i\right) = \left\|\mathbf{s}_i\right\|_1 = \sum_{j \in \mathbb{J}_i} s_{ij} \triangleq \lambda_{s,i}, \tag{5}$$

where $j$ is the MB address, and $\mathbb{E}(\cdot)$ and $\|\cdot\|_1$ are the energy operator and $\ell_1$ norm operator, respectively. In Fig. 5(a), we show an example of $\{\lambda_{s,i}\}_{i=1}^n$ in a sequence, from which we can see that the value of $\lambda_{s,i}$ is volatile due to the I-frames. To eliminate the negative effect of the I-frames, we perform median filtering on the frame data size sequence $\{\lambda_{s,i}\}_{i=1}^n$. Let us denote the frame rate of the video as $\rho$, which represents the number of frames displayed per second. We use $\frac{\rho}{5}$ frames as the size of the median filter window. After performing median filtering, we obtain the new frame data sizes from the filtered sequence. For convenience, we use $\lambda_{s,i}$ to denote the new frame data size. We present the values of $\lambda_{s,i}$ after filtering in Fig. 5(b). We can see that noise has less influence and the curve is much smoother.

For the MB partition information, we use different partition levels to identify different partition modes. Suppose that $l_{ij}$ is the partition level of $\mathbf{MB}_j$ in the $i$-th frame, and $\mathbf{l}_i$ is defined as $\{l_{i0}, l_{i1}, \ldots\}$ in the $i$-th frame. Similarly, we can obtain the partition level of each frame as

$$\mathbb{E}\left(\mathbf{l}_i\right) = \left\|\mathbf{l}_i\right\|_1 = \sum_{j \in \mathbb{J}_i} l_{ij} \triangleq \lambda_{l,i}. \tag{6}$$

After performing median filtering, we obtain a new sequence of $\{\lambda_{l,i}\}_{i=1}^n$.

The MVD energy of $\mathbf{MB}_j$ reflects the motion information in $\mathbf{MB}_j$. Thus, all the MVD energies in a frame approximately indicate the general information of the frame. Let us use $v_{ij}$ to denote the MVD energy of $\mathbf{MB}_j$ in the $i$-th frame and $\mathbf{v}_i = \{v_{i0}, v_{i1}, \ldots\}$ to denote the vector of $v_{ij}$. The MVD energy of the $i$-th frame can then be computed as

$$\mathbb{E}\left(\mathbf{v}_i\right) = \left\|v_i\right\|_1 = \sum_{j \in \mathbb{J}_i} v_{ij} \triangleq \lambda_{v,i}. \tag{7}$$

We then perform median filtering to obtain the new sequence of $\{\lambda_{v,i}\}_{i=1}^n$. In the following, for convenience, we use $\lambda_{\chi,i}$ to represent any element in $\{\lambda_{s,i}, \lambda_{l,i}, \lambda_{v,i}\}$ and use $x_{ij}$ to denote any element in $\{s_{ij}, l_{ij}, v_{ij}\}$.

The second type of extracted feature in the proposed frame-level method is the range of the data variation. Let us use $\mathbf{x}_i$ to denote $\{x_{i0}, x_{i1}, \ldots\}$. Since a rapid change in $x_{ij}$ indicates a possible anomaly, we compute the variance of $\mathbf{x}_i$ as

$$\sigma_{\chi,i}^2 = \underset{j \in \mathbb{J}_i}{\mathrm{Var}}\left(\mathbf{x}_i\right) = \frac{1}{|\mathbb{J}_i| - 1} \sum_{j \in \mathbb{J}_i} \left(x_{ij} - \bar{x}_i\right)^2 \triangleq \delta_{\chi,i}, \tag{8}$$

where $\mathrm{Var}\left(\cdot\right)$ is the sample variance operator.

The third type of extracted feature is related to the metric of the significant data. From the perspective of data, an anomaly is conspicuous. Thus, we extract information from the maximum $x_{ij}$ as the feature of the $i$-th frame. We acquire the top $\alpha$ percentage of $\mathbf{x}_i$ in the $i$-th frame, denoted by $\mathbf{x}_{i,\alpha}$, and use $\mathbb{J}_\alpha$ to denote the set of all the indexes of $j$ in $\mathbf{x}_{i,\alpha}$. We approximately evaluate the conspicuous tendency of the $i$-th frame as

$$\mathbb{E}\left(\mathbf{x}_{i,\alpha}\right) = \left\|\mathbf{x}_{i,\alpha}\right\|_1 = \sum_{j \in \mathbb{J}_\alpha} x_{ij} \triangleq \mu_{\chi,i}. \tag{9}$$

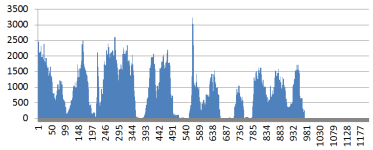
Finally, the features obtained by the frame-level method are $[S, L, V]$, where $S = [\lambda_{s,i}, \delta_{s,i}, \mu_{s,i}]_i$, $L = [\lambda_{l,i}, \delta_{l,i}, \mu_{l,i}]_i$, and $V = [\lambda_{v,i}, \delta_{v,i}, \mu_{v,i}]_i$.

*2) Anomaly Detection Algorithm:* In video abnormal detection, the distribution of normal sample can be well approximated by Gaussian mixture model [17]. Compared with other common learning algorithms, adaptive kernel density estimator [31] does not need any prior knowledge of the sample distribution and can make full use of the sample data to produce a particularly effective approximation of the distribution. Hence, in this paper, we use the adaptive kernel density estimator to estimate the probability density function

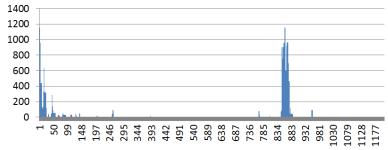
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2929855, IEEE Transactions on Circuits and Systems for Video Technology




(a) Example frame



(b) Block 1



(c) Block 2

Fig. 7: Examples of different regions.

(pdf) of the extracted feature, i.e.,

$$f_{\text{pdf}}(z) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{K}_h(z - z_i) = \frac{1}{mh}\sum_{i=1}^{m}\mathcal{K}\left(\frac{z - z_i}{h}\right), \quad (10)$$

where $\mathcal{K}(\cdot)$ is the kernel function, $z$ is the data, $m$ is the number of samples, and $h$ is the adaptive bandwidth. Specifically, we use a Gaussian kernel in our adaptive kernel density estimator. For convenience, we use $F_{i,u}$ to denote the unified extracted feature, i.e.,

$$[F_{i,u}]_{u=1}^{9} = [\lambda_{s,i}, \lambda_{\ell,i}, \lambda_{v,i}, \sigma^2_{s,i}, \sigma^2_{\ell,i}, \sigma^2_{v,i}, \mu_{s,i}, \mu_{\ell,i}, \mu_{v,i}], \quad i = 1, 2, \ldots, n, \quad (11)$$

where $i$ indicates the frame number, and $u$ is the feature index. For every feature in the $i$-th frame, we can obtain the probability $\Pr(F_{i,u})$ $(u = 1, 2, ...)$ from the estimated probability density function. The score of the $i$-th frame can then be evaluated as

$$\mathscr{C}_i = \prod_{u=1}^{9}\Pr(F_{i,u}). \quad (12)$$

To address the case of a continuous anomaly, we apply median filtering to the frame score sequence $\{\mathscr{C}_i\}_i$. We consider a frame to be an anomaly when $\mathscr{C}_i < \tau$, where $\tau$ is the threshold. This consideration is based on the fact that events with smaller possibilities are more likely to be abnormal events than normal events. We show an example of anomaly detection with only one feature in Fig. 6(a). The samples whose $\mathscr{C}_i$s are less than $\tau$ (shown in the red area) are considered to be the anomalies.

### B. Anomaly Region Localization

*1) Feature Extraction:* Different anomalies have different areas of influence. Some have global influences on a frame, while others affect only local regions. For example, in Fig. 2, we can see that one anomaly event (i.e., the running man) influences only a small region compared to another anomaly

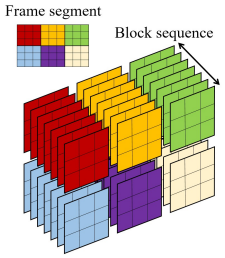



Fig. 8: Block segment method.

event (i.e., the man walking on the wrong path). In the situation that the anomaly influences only a local region, detection at the MB level would be more useful than that at the frame level. As consequence, more computation time is required since there is a large number of MBs in a single video. Meanwhile, the MBs of one frame are very likely to share the same statistical properties within the same region and to have different characteristics when located in different regions. For example, in Fig. 7, the mean MB size (in bits) of the regions in the middle of the road ("Block 1") is larger than the mean MB size of the regions in the grass ("Block 2"). As a compromise, we divide a frame into several blocks to reduce the computation cost and to make use of the statistical characteristics of MBs. The original frame sequence of the video is then divided into several block sequences. For example, in Fig. 8, we divide each frame into six non-overlapping blocks and obtain six block sequences from the original video. Feature extraction is then performed individually on each block sequence.

Let us use $\hat{\lambda}^k_{s,i}, \hat{\lambda}^k_{\ell,i}, \hat{\lambda}^k_{v,i}$ to denote the features extracted from the MB size, partition, and motion vector information in the $k$-th block of the $i$-th frame. For convenience, the symbol $\varkappa$ is used to denote any one of $s$, $\ell$, or $v$. We denote the set of $x_{ij}$ in the $k$-th block of the $i$-th frame by $\mathbf{x}^k_i$. The value of $\hat{\lambda}^k_{\varkappa,i}$ is then computed as

$$\hat{\lambda}^k_{\varkappa,i} = \mathbb{E}\left(\mathbf{x}^k_i\right) = \left\|\mathbf{x}^k_i\right\|_1 = \sum_{j\in\mathbb{b}_{ik}} x_{ij}, \quad (13)$$

where $\mathbb{b}_{ik}$ is the set of all the indexes of $j$ of $x_{ij}$ in $\mathbf{x}^k_i$. Note that in the proposed localization method, median filtering is performed at the MB level. In addition, the intra-block variance is adopted to measure the change inside the block, i.e.,

$$\text{Var}\left(\mathbf{x}^k_i\right) = \frac{1}{|\mathbb{b}_{ik}| - 1}\sum_{j\in\mathbb{b}_{ik}}\left(x^k_{ij} - \bar{x}^k_i\right)^2 \triangleq \hat{\delta}^k_{\varkappa,i} \quad (14)$$

Considering that an anomaly is a peak in the data through time and space, we use the block variances in the time and spatial domains to represent these statistical characteristics. Assume that $\beta$ and $\gamma$ are two predefined thresholds. In the $k$-th block of the $i$-th frame, we compute the sample variances in the time and spatial domains, i.e.,

$$\hat{\phi}^k_{\varkappa,i} = \text{Var}\left(\left\{\hat{\lambda}^k_{\varkappa,i+m} : |m| \le \beta, m \in \mathbb{Z}\right\}\right), \quad (15)$$

$$\hat{\varphi}^k_{\varkappa,i} = \text{Var}\left(\left\{\hat{\lambda}^{k+m}_{\varkappa,i} : |m| \le \gamma, m \in \mathbb{Z}\right\}\right), \quad (16)$$

where $\hat{\phi}^k_{\varkappa,i}$ is the block variance in the time domain, and $\hat{\varphi}^k_{\varkappa,i}$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2929855, IEEE Transactions on Circuits and Systems for Video Technology



is the inter-block variance in the spatial domain.

From the block sequence of the video, we obtain the anomaly features $[\hat{S}, \hat{L}, \hat{V}]$, where $\hat{S} = [\hat{\lambda}^k_{s,i}, \hat{\delta}^k_{s,i}, \hat{\phi}^k_{s,i}, \hat{\varphi}^k_{s,i}]_{i,k}$, $\hat{L} = [\hat{\lambda}^k_{\ell,i}, \hat{\delta}^k_{\ell,i}, \hat{\phi}^k_{\ell,i}, \hat{\varphi}^k_{\ell,i}]_{i,k}$, and $\hat{V} = [\hat{\lambda}^k_{v,i}, \hat{\delta}^k_{v,i}, \hat{\phi}^k_{v,i}, \hat{\varphi}^k_{v,i}]_{i,k}$.

*2) Anomaly Localization Algorithm:* Based on the features $[\hat{S}, \hat{L}, \hat{V}]$, we propose an anomaly localization algorithm that is similar to the frame-level anomaly detection algorithm. We use $\hat{F}_{i,u}$ to denote the unified extracted feature, i.e.,

$$[\hat{F}^k_{i,u}]^{12}_{u=1} = [\hat{\lambda}^k_{s,i}, \hat{\lambda}^k_{\ell,i}, \hat{\lambda}^k_{v,i}, \hat{\delta}^k_{s,i}, \hat{\delta}^k_{\ell,i}, \hat{\delta}^k_{v,i}, \hat{\phi}^k_{s,i}, \hat{\phi}^k_{\ell,i}, \hat{\phi}^k_{v,i}, \hat{\varphi}^k_{s,i}, \hat{\varphi}^k_{\ell,i}, \hat{\varphi}^k_{v,i}], i = 1, 2, \ldots, n, \quad (17)$$

where $i$ indicates the frame number, $k$ indicates the block number, and $u$ is the feature index. For every feature in the $k$-th block of the $i$-th frame, we can obtain the probability $\Pr(\hat{F}^k_{i,u})(u = 1, 2, ...)$ from the estimated pdf, and then compute the score in the $k$-th block of the $i$-th frame as

$$\mathscr{C}^k_i = \prod_{u=1}^{12} \Pr\left(\hat{F}^k_{i,u}\right). \quad (18)$$

A block is considered to be an anomaly when $\mathscr{C}^k_i < \tau^k$. We can search the optimum values of $\tau^k$ in every block during the training stage, i.e., the adaptive kernel density estimation. However, searching the optimum values of $\tau^k$ in every block may sometimes time-consuming, since there are many blocks in a frame. To solve this problem, we suggest using a parameter $\theta$ to generate $\tau^k$s in different blocks.

In Fig. 6(a), there is a red dotted line $y = \tau$ that is parallel to the $x$-axis. We can see that the red area under the dotted line increases as the value of $\tau$ increases. Hence, the red area $\texttt{Area}\left(\{\mathscr{C}_i < \tau\}\right) \leq \theta$. Similarly, in the case of anomaly localization, in every block sequence, we have $\texttt{Area}\left(\{\mathscr{C}^k_i < \tau^k\}\right) \leq \theta$. Since the curves of $\texttt{Area}\left(\{\mathscr{C}^k_i < \tau^k\}\right)$ in different blocks are different, $\texttt{Area}\left(\{\mathscr{C}^k_i < \tau^k\}\right) = \theta$ will produce different values of $\tau^k$ in different blocks. We show an example of determining $\tau^k$ for a fixed value $\theta = 0.01$ in Fig. 6(b). The $x$-axis indicates the value of $\tau^k$, and the blue curve corresponds to the value of $\texttt{Area}\left(\{\mathscr{C}^k_i < \tau^k\}\right)$. We also plot the red dotted line $y = \theta$, which intersects the blue line at a single point. The value of $\tau^k$ is the horizontal coordinate of the intersection.

Furthermore, to avoid falsely detecting the normal event that located in the $\theta$ lowest percentile of the obtained score, we perform anomaly localization algorithm only on the anomaly frames detected by the frame-level method. Since the previous frame-level detection method can effectively detect the abnormal frames and filter most of the normal frames, we can succeed to reduce the false localization rate of the proposed abnormal localization algorithm.

### C. Parallel Implementation

To improve the computational efficiency of our scheme, we can in parallel implement the proposed frame-level abnormal detection method and abnormal localization method. In the H.264/AVC, the video data are packed in NALUs. These NALUs can be considered to be independent video fragments, represented as $[\mathbf{VF}_1, \mathbf{VF}_2, \ldots]$. We can therefore directly perform frame-level and localization feature extractions in these video fragments. Considering the independence of these video fragments in the compressed video bitstream, the proposed anomaly detection and localization schemes can be implemented in parallel in the video bitstream. We provide more detailed experimental results in Section V-G.

TABLE III: Detailed descriptions of the experimental dataset and settings

| Dataset | Number of Videos | Train/test splits | Scenarios | Resolution | Number of abnormal events |
|---|---|---|---|---|---|
| *Avenue* | 37 | 15328/15324 | CUHK campus *Avenue* | 640360 | 14 |
| *Subway* | 2 | 27000/56155 | *Subway* | 512384 | 19 |
| *UMN* | 11 | 3456/4283 | Several surveillance scenarios | 320240 | 11 |
| *UCSD* | 70 | 6800/7200 | Walkways | 238158 | 10 |

## V. Experiments

### A. Experimental Settings

The proposed scheme has been tested on the videos of *Avenue*, *Subway*, *UMN*, and *UCSD*. These videos are encoded using the H.264/AVC reference software version JM-18.6 with variable bit rate mode. The frame rate is 30 frames/s. We use a window length of 5 to calculate the variance and a maximum of $10\%$ of the elements to calculate $\mu_{\chi,i}$ in the frame-level method. The window length in the time domain is 5, and it is 7 in the space domain of the localization method. We provide more detailed descriptions of the experimental dataset and settings in Table III. The train/test splits are all the same as that of the compared works. The parallelization results were generated on a 64-bit Windows Server 2008 server with four Intel Xeon E7-L8867 CPUs @2.13GHz and 256GB memory. All the other results were generated on a 64-bit Windows 7 PC with Intel Core i7-4790 CPU @3.60GHz and 16GB memory.

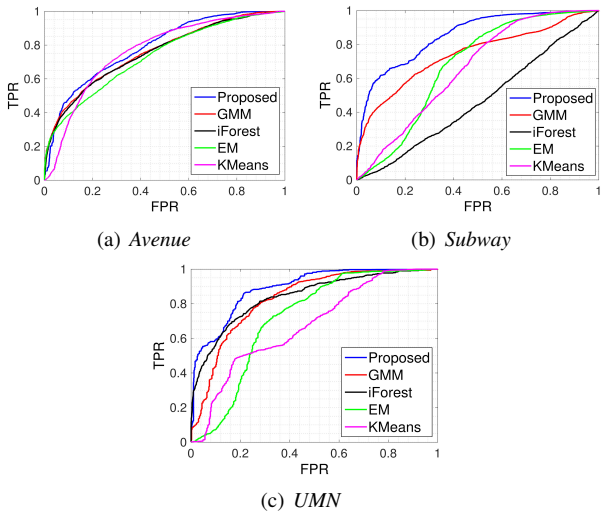
We adopt format-compliant partial video bitstream encryption, due to the constraints of the format-compliance requirements of the video decoder and the heavy computational cost of video processing. Specifically, we modify the H.264/AVC (baseline profile) bitstream encryption in [7] and use it in the proposed privacy-preserving anomaly detection protocol. In our modified video encryption, we additionally encrypt the values of the MVDs, the residual data, their signs, to strength privacy protection. Note that our video anomaly detection protocol is not designed to rely on this specific video encryption scheme. Other format-compliant video encryption schemes are also applicable in our protocol. We discuss this issue further in Section VI.

### B. Abnormal Frame Detection

We have also conducted experiments to evaluate the proposed scheme using the ROC (receiver operating characteristic) curve. The ROC curve is obtained by varying the value of $\tau$. The AUC (area under the curve) is $0.85$ for *Subway*. The corresponding EER (equal error rate) is $0.26$. By using the features in the proposed frame-level method, we have tested different techniques that are widely used in anomaly detection,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2929855, IEEE Transactions on Circuits and Systems for Video Technology

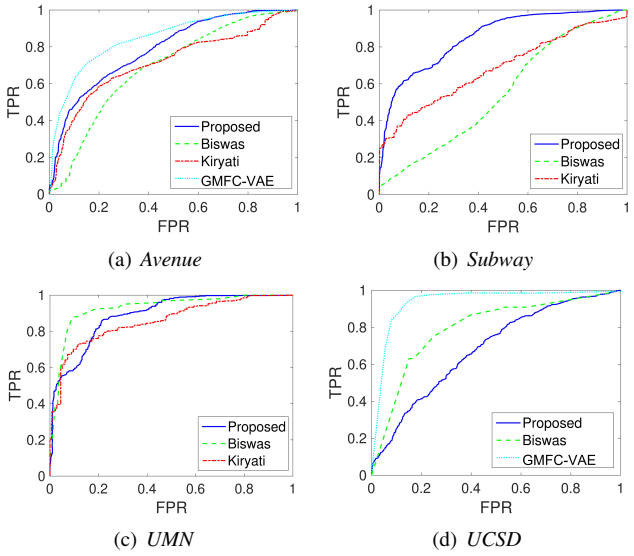





(a) *Avenue* (b) *Subway*

(c) *UMN*

Fig. 9: Comparison of the ROC curves of different detection methods (using features in the proposed frame-level method).




(a) *Avenue* (b) *Subway*

(c) *UMN* (d) *UCSD*

Fig. 10: Comparison of the ROC curves of the frame-level methods on four dataset in the plaintext domain.

i.e., the proposed method with the GMM [32], iForest [33], EM [34], and KMeans [35]. The results are shown in Table IV and Fig. 9. The ROC curves of our scheme on the *Subway* and *UMN* datasets are to the upper left of the other curves, which means that the proposed frame-level features and detection algorithm is generally better than the other methods.

We have compared the performance of our scheme with that of some plaintext domain methods [20], [23], [24]. "Bitwas" [23] and "Kiryati" [24] are performed on the compressed video bitstream. "GMFC-VAE" [20] is a deep learning method performed on the video frame sequence. We have also compared the performance of these schemes in encrypted video. "Bitwas" and "Kiryati" are modified to adopt to this situation. Specifically, We replace the MV magnitude with the estimated MVD magnitude in the "Bitwas" method to obtain **MVDM**, and then remove the MV orientation-related features and replace the MV features with the MVD features in the

TABLE IV: Detection comparison of different methods.

| Dataset | *Avenue* | | *Subway* | | *UMN* | |
|---|---|---|---|---|---|---|
| Method | AUC | EER | AUC | EER | AUC | EER |
| Proposed | **0.79** | 0.29 | **0.85** | **0.26** | **0.89** | **0.18** |
| GMM | 0.76 | 0.31 | 0.74 | 0.32 | 0.83 | 0.24 |
| iForest | 0.75 | 0.32 | 0.47 | 0.53 | 0.84 | 0.23 |
| EM | 0.73 | 0.34 | 0.68 | 0.35 | 0.72 | 0.30 |
| Kmeans | 0.77 | **0.28** | 0.66 | 0.40 | 0.68 | 0.40 |

TABLE V: Performance comparison. The plaintext domain and the encrypted domain denote the abnormal detection algorithms are performed on the plaintext and the encrypted video, respectively. FLM denotes the frame-level method. CB and FS represent that the inputs are compressed bitstream and video frame sequence, respectively.

| The plaintext domain | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Method | Dataset | *Avenue* | | *Subway* | | *UMN* | | *UCSD* | |
| | Input | AUC | EER | AUC | EER | AUC | EER | AUC | EER |
| Proposed FLM | CB | 0.79 | 0.29 | **0.85** | **0.26** | 0.89 | 0.18 | 0.69 | 0.38 |
| Biswas [23] | CB | 0.69 | 0.34 | 0.55 | 0.49 | **0.94** | **0.11** | 0.79 | 0.24 |
| Kiryati [24] | CB | 0.72 | 0.34 | 0.69 | 0.38 | 0.86 | 0.23 | - | - |
| GMFC-VAE [20] | FS | **0.83** | **0.23** | - | - | - | - | **0.95** | **0.11** |
| The encrypted domain | | | | | | | | | |
| Method | Dataset | *Avenue* | | *Subway* | | *UMN* | | *UCSD* | |
| | Input | AUC | EER | AUC | EER | AUC | EER | AUC | EER |
| Proposed FLM | CB | **0.79** | **0.29** | **0.85** | **0.26** | **0.89** | **0.18** | **0.69** | **0.38** |
| MVDM | CB | 0.65 | 0.40 | 0.49 | 0.50 | 0.72 | 0.37 | 0.62 | 0.43 |
| MVDF | CB | 0.53 | 0.47 | 0.54 | 0.49 | 0.66 | 0.39 | 0.58 | 0.45 |
| GMFC-VAE [20] | - | - | - | - | - | - | - | - | |

"Kiryati" method to have **MVDF**.

The comparison results are shown in Table V and Fig. 10. Note that for the "Bitwas" comparison, we have to resize the *UMN* sequences to $720 \times 480$ (according to the requirements of "Bitwas"). The proposed frame-level method can be directly applied in the plaintext domain without modification. Since the adopted format-compliant encryption has no effect on the bit rate, i.e., the codeword length after encryption is not changed, our method achieves the same performance on plaintext and encrypted video. For plaintext videos, "GMFC-VAE" [20] has the best performance on *Avenue* and *UCSD* by taking advantage of applying the deep learning method to the video frame sequence directly. However, our performance is still comparable to or better than that of the same type of bitstream-based methods [23], [24]. Moreover, "GMFC-VAE" [20] is not appropriate for the encrypted videos. On the contrary, the proposed scheme achieves the highest performance on encrypted videos.

### C. Abnormal Region Localization

We show an example of the localization results in Fig. 11. This example is from *Avenue*, which is the anomaly activity of running. The detected regions are larger than the ground truth because we have applied median filtering to the results to detect the continuous anomaly. The fast running activity has an effect on the previous and subsequent frames. However, the anomaly regions are well covered. These continuous frames show that we can successfully locate anomaly regions with the proposed localization method.

To evaluate the performance of the proposed localization method, we use the ROC curve, AUC, and EER of the detection result on the *UCSD* sequences at the pixel level. In the evaluation of the plaintext domain, the proposed method takes the original video bitstream as input and then resize the resolution from $238 \times 158$ to $720 \times 480$. After compressing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2929855, IEEE Transactions on Circuits and Systems for Video Technology



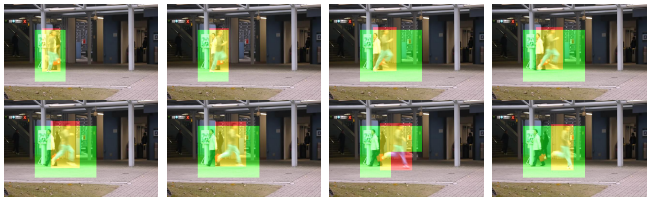


Fig. 11: Examples of the localization results. The yellow regions are the correctly detected regions. The green regions are the false positives. The red regions are the false negatives.

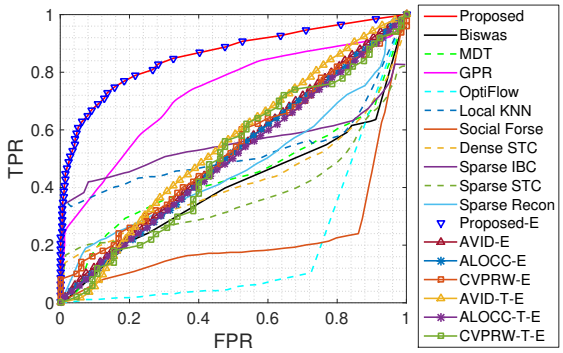



Fig. 12: Comparing ROC curves of different localization methods. The suffix "-E" denotes the that the method is performed on the encrypted video in *UCSD-E480P*. The suffix "-T-E" denotes that we train the deep networks directly from *UCSD-E480P*. All the other methods are performed on the plaintext video in *UCSD*.

the resized video frame sequences into H.264/AVC video bitstreams, we perform our localization algorithm. We compare the proposed localization method with some existing works in Fig. 12 and Table VI. The AUC and EER of the proposed localization method are 0.85 and 0.21, respectively. We have also compared our algorithm with other works in the plaintext domain, i.e., MDT [11], GPR [36], Optical Flow [37], Local KNN [38], Social [39], Dense STC [40], Sparse STC [40], Sparse IBC [41], and Sparse Recon [13]. Note that all the above compared works are performed on raw video (video frame sequence) in the plaintext domain. Fig. 12 and Table VI show that the performance of the proposed localization method is comparable to the frame-based methods. The method in [23] is a state-of-the-art bitstream-based method like ours. With the same evaluation, the AUC/EER of Biswas [23] is only 0.60/0.44, which is worse than that of our localization method. The ROC curve of our localization method is to the upper-left of the curve of Biswas, as shown in Fig. 12. Therefore, the proposed localization method achieve better performance than the compared bitstream-based method.

The localization algorithm does not perform well on the encrypted low-resolution video. However, it achieves a satisfactory performance on the encrypted video of common-sized resolution, e.g., 480P ($720 \times 480$). The original resolution of *UCSD* is too small, and there is no suitable video dataset of high resolution for anomaly localization. As a compromise, we took the encrypted resized *UCSD* video as input. Specifically, the resolution of the frames of the video in *UCSD* was resized to be $720 \times 480$. Then the resized video frame sequences were compressed into H.264/AVC video bitstreams. We performed format-compliant video encryption on the resized *UCSD*, and obtain the encrypted resized *UCSD* (denoted by *UCSD-E480P*). We then applied our localization algorithm on *UCSD-E480P*. For comparison, we conducted experiments to test the state-of-the-art deep learning methods AVID [16], ALOCC [17], and CVPRW2015 [18] on *UCSD-E480P*. We firstly perform the abnormal localization on *UCSD-E480P* directly with the deep learning networks [16]–[18]. We then adopt the training methods in [16]–[18] to obtain new deep networks from *UCSD-E480P*. The abnormal localization is then performed with the obtained deep networks. We show the experimental results in Fig. 12 and Table VI. We can see that all the deep learning methods not work on *UCSD-E480P*. In contrast, our performance on *UCSD-E480P* is the same as that of *UCSD*. From these observations, we can conclude that the proposed localization method achieves the best performance in the encrypted domain.

TABLE VI: Localization performance comparison. The plaintext domain and the encrypted domain denote the abnormal detection algorithms are performed on the plaintext and the encrypted video, respectively. LM denotes the localization method. CB and FS represent that the inputs are compressed bitstream and video frame sequence, respectively. CVPRW-T, AVID-T and ALOCC-T mean that the deep networks are obtained by training from the encrypted video dataset with CVPRW2015, AVID and ALOCC, respectively.

| The plaintext domain | | | | The encrypted domain | | | |
|---|---|---|---|---|---|---|---|
| Method | Input | AUC | EER | Method | Input | AUC | EER |
| Proposed LM | **CB** | 0.85 | 0.21 | Proposed LM | **CB** | **0.85** | **0.21** |
| Biswas [23] | **CB** | 0.60 | 0.44 | AVID [16] | FS | 0.51 | 0.49 |
| MDT [11] | FS | 0.44 | 0.58 | ALOCC [17] | FS | 0.52 | 0.50 |
| GPR [36] | FS | 0.72 | 0.32 | CVPRW [18] | FS | 0.53 | 0.49 |
| Optical Flow [37] | FS | 0.13 | 0.76 | AVID-T [16] | FS | 0.50 | 0.50 |
| Local KNN [38] | FS | 0.52 | 0.51 | ALOCC-T [17] | FS | 0.55 | 0.48 |
| Social [39] | FS | 0.18 | 0.78 | CVPRW-T [18] | FS | 0.50 | 0.46 |
| Dense STC [40] | FS | 0.42 | 0.58 | | | | |
| Sparse STC [40] | FS | 0.37 | 0.63 | | | | |
| Sparse IBC [41] | FS | 0.55 | 0.46 | | | | |
| Sparse Recon [9] | FS | 0.46 | 0.54 | | | | |

TABLE VII: Influence of different features in the proposed method. FLM: frame-level method.

| Dataset | *Avenue* | | *Subway* | | *UMN* | | *UCSD* | |
|---|---|---|---|---|---|---|---|---|
| Feature | AUC | EER | AUC | EER | AUC | EER | AUC | EER |
| $S$ | 0.71 | 0.37 | 0.61 | 0.42 | 0.70 | 0.33 | 0.60 | 0.40 |
| $L$ | 0.78 | 0.30 | 0.73 | 0.35 | 0.79 | 0.22 | 0.55 | 0.45 |
| $V$ | 0.70 | 0.36 | 0.79 | 0.28 | 0.89 | 0.2 | 0.47 | 0.53 |
| $SL$ | 0.78 | 0.30 | 0.72 | 0.33 | 0.79 | 0.28 | 0.61 | 0.39 |
| $SV$ | 0.74 | 0.32 | 0.80 | 0.28 | 0.89 | 0.21 | 0.61 | 0.39 |
| $LV$ | 0.78 | 0.31 | 0.80 | 0.28 | 0.89 | 0.20 | 0.60 | 0.40 |
| FLM | 0.79 | 0.29 | 0.85 | 0.26 | 0.89 | 0.18 | 0.69 | 0.38 |

### *D. Investigating the Influences of Different Features and the Sensitivity to Hyper-Parameters*

We have conducted experiments to investigate the influence of different features. The experiments use different subsets of the features $[S, L, V]$ for training and detection (e.g., only $S$ is used). We show the results in Table VII. For example, on the dataset *Subway*, the AUC is 0.79 when using feature $V$ extracted from the MVD information. We can see that the features from the motion information have a larger contribution. In the case of *Avenue*, feature $L$ has better performance. Thus, the proposed three types of features generally capture different abnormal information, and they are complementary in our scheme.

To investigate the sensitivity of the experimental results to the hyper-parameters, we adopt four median filter sizes, i.e.,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2929855, IEEE Transactions on Circuits and Systems for Video Technology



TABLE VIII: The experiment results on the dataset *Avenue* with the proposed localization methods, by using different median filter sizes and different kinds of non-overlapping blocks

| Median filter size | 3 | 5 | 7 | 9 |
|---|---|---|---|---|
| AUC | 0.7074 | 0.7002 | 0.691 | 0.6925 |
| EER | 0.3455 | 0.3546 | 0.3698 | 0.3784 |
| Non-overlapping block | $2\times2$ | $4\times3$ | $8\times6$ | $10\times8$ |
| AUC | 0.6176 | 0.7307 | 0.7002 | 0.6744 |
| EER | 0.4143 | 0.3416 | 0.3546 | 0.3695 |

TABLE IX: Performance on videos encrypted by other video encryption schemes [29], [42]

| | Scheme [29] | | Scheme [42] | |
|---|---|---|---|---|
| Dataset | AUC | EER | AUC | EER |
| *Avenue* | 0.80 | 0.29 | 0.77 | 0.31 |
| *UMN* | 0.90 | 0.19 | 0.88 | 0.20 |

3, 5, 7, and 9, and four kinds of non-overlapping blocks, i.e., $2\times2, 4\times3, 8\times6$, and $10\times8$, where $2\times2$ indicates the size of each block is consist of 4 macroblocks $(2\times2)$, and so on. We have conducted experiments on the dataset *Avenue* with the proposed localization methods. The results of AUC and EER are shown in Table VIII, from which we can see that the median filter size has less influence on the performance. However, either smaller or larger non-overlapping block will decrease the performance. A proper non-overlapping block, e.g., $4\times3$ or $8\times6$, will make our scheme have a better result.

### *E. Performance with Other Video Encryption Methods*

The proposed anomaly detection and localization algorithms are not restrained by the video encryption method [7]. Other format-compliant video encryption methods can be adopted. We have performed the proposed detection scheme on videos encrypted by another two video encryption schemes [29], [42]. In Table IX, we report the experimental results. The proposed frame-level method maintains its detection performance in this situation. More discussion is provided in Section VI-C.

Our detection algorithms can also be applied to plaintext video bitstreams. We have applied the proposed detection algorithm to unencrypted video bitstreams without any changes in the features. The detection performance is almost identical to that on encrypted video. Because our detection algorithm is based on the information extracted from the bitstream structure, we have extracted almost the same feature information from the encrypted and unencrypted bitstreams.

### *F. Detection Rate*

The proposed frame-level method directly addresses video bitstream data without full decoding and decryption in the detection phase. Thus, the proposed frame-level method can achieve high efficiency compared to other video processing techniques, e.g., motion detection in the encrypted domain [43]. In terms of the detection algorithm itself, the detection rate of our frame-level method is approximately 200 frames per second for $640\times480$ videos. For the same videos, the detection rates of [23], [24] are approximately 70 frames per second and 50 frames per second, respectively. Therefore, our scheme is more than 2 times faster than the other schemes [23], [24].

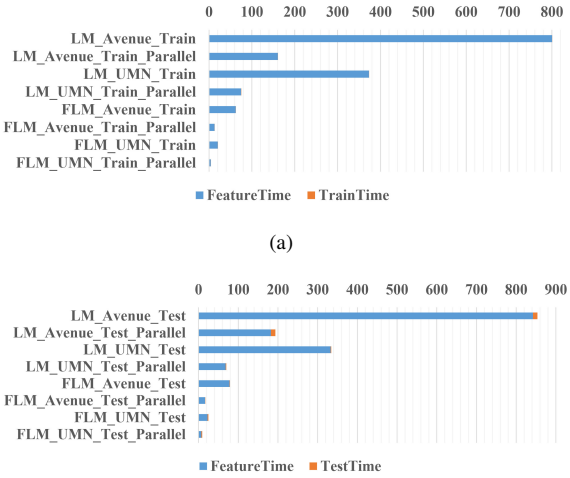



Fig. 13: Example of parallel acceleration, where FLM and LM represent the proposed frame level detection method and localization method, respectively.

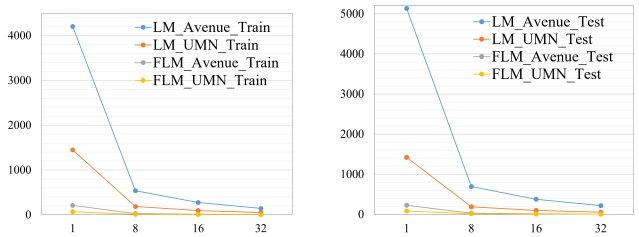



Fig. 14: Running time. FLM and LM represent the proposed frame level detection method and localization method, respectively.

### *G. Parallelization Performance*

The detection speed can be further increased by performing parallel acceleration. We show an example of parallel acceleration in Fig. 13. We can see that the feature extraction consumes most of the time. After applying parallel implementation with 16 cores, the running time is reduced substantially. Note that this acceleration is especially useful for the localization method. Take the proposed localization method on *Avenue* as an example. The detection rate is approximately 3.3 frames per second before parallelization and 51.1 frames per second after applying parallelization. We present additional results for different numbers of cores in Fig. 14. We can see that the detection scheme is highly efficient when using the proposed parallelization algorithm.

## VI. Discussions

### *A. Security Analysis*

Video encryption is used to protect privacy against cloud servers and attackers. The main goal of video encryption is to protect the video content from being revealed. In this security context, as discussed in [28], video security simply means that an adversary should not be able to reconstruct the plaintext from the encrypted video at a level of quality higher than allowed in the application scenario.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2929855, IEEE Transactions on Circuits and Systems for Video Technology



TABLE X: Several statistic performance results of video encryption with different encrypted keys. Five metrics [29] are used, including edge similarity score (ESS), luminance similarity score (LSS), neighborhood similarity degree (NSD), local entropy (LE), local feature based visual security (LFBVS).

| Evaluation | Key1 | key2 | Key3 |
|---|---|---|---|
| ESS | 0.3643 | 0.3641 | 0.3642 |
| LFBVS | 0.4419 | 0.4417 | 0.4417 |
| LSS | -1.4716 | -1.4697 | -1.4685 |
| LE | 0.0567 | 0.0567 | 0.0567 |
| NSD | 0.0703 | 0.0702 | 0.0701 |

We investigate two video encryption schemes [7], [29], in our experiments. The secure stream cipher is used to encrypt the bitstream in [7], and the block cipher AES (CFB mode) is utilized in [29]. The proven secure cryptographic primitive protects the encrypted video against cryptographic attacks. The video encryption [7] encrypts intra prediction mode (IPM), MVD, and residual coefficients, which makes the encrypted video unintelligible and thus protects the sensitive content. In the video encryption [29], the codewords of the nonzero DCT coefficients are encrypted, which destroys the relationship with the original video and protects the original content. We refer to [7], [29] for the detailed security analysis of these schemes.

We have conducted experiments to provide statistic performance results on video encryption with three random keys. As suggested in [29], we evaluated the performance of video encryption on the training sequence on *Avenue* with five metrics. The mean values of all the test video with the five metrics are provided in Table X. We can see that the performances of the video encryption with different encryption keys are the almost the same under the five metrics.

### B. Explanation of the Three Types of Feature Information

We provide a physical explanation for why the proposed combination of information of three features can capture anomalous motion in videos. In general, an object undergoing anomalous motion has greater acceleration. When the background is static, an object with anomalous motion generally has faster speed. The feature information of the motion vector difference can capture these characteristics of anomalous motion. Furthermore, the human body is elastic rather than a particle or rigid body. The movement of an elastic body is more complicated than that of a particle or rigid body, which means that we can capture more characteristics for anomaly detection. For example, when a person is running, his/her arms, legs, head, etc. are all moving. This physical fact is reflected in the video encoding. A slice/tile in a bitstream containing an anomalous moving body will be divided into more sub-MBs by the video compression standard. It then will produce more zero values in the small partitions after motion prediction, and achieves a higher compression rate. Moreover, many sub-MBs cannot find exactly the same sub-MBs as in the previous frames. For example, sub-MBs related to the arms of an anomalous moving person in different frames may vary greatly. Therefore, most of the residual data in these sub-MBs are not zeros, which results in an increase in the bits of the MB data. The feature information of MB partitions and MB size can capture these characteristics of the anomalous motion of elastic bodies. The three types of feature information reflect different properties of anomalous motion. By combining the three complementary features, our scheme can be applied to different scenarios. However, there are still some situations in which our scheme is not efficient. For example, if the video is recorded by a moving camera, our scheme may not work because the statistics are different than those of a fixed camera. Another situation where our method may not work is when an anomaly occurs among other rapidly moving objects. For example, an illegal road change in a traffic lane. Our scheme focuses on situations where most of the moving objects are humans.

### C. Adaptation to other Video Coding Standards and Video Bitstream Encryption Schemes

Although we use the H.264/AVC video encoding framework and three video encryption schemes [7], [29], [42] in this paper, the proposed anomaly detection scheme is not specifically designed for them. We can adapt the proposed scheme to other video encoding frameworks since the feature information can be extracted via the common techniques adopted in video compression. For example, motion compensation is also adopted in H.265/HEVC. Thus, we can extract the energy of MVDs from the H.265 bitstream.

We can also generalize the proposed abnormal detection scheme to other video bitstream encryption methods. For example, if there is a video encryption in which the coefficients are encrypted before encoding (which would change the MB size), then we can make an adaptation to use the motion information and MB partition information for abnormal detection from the encrypted video. In this case, the proposed abnormal detection framework is suitable for most format-compliant video bitstream encryptions.

### D. Investigation on Using Deep Learning Methods

We provide some investigation on the challenge of using deep learning methods in our application scenario.

*1) :* The video encryption is performed directly on the compressed video bitstream. After video decompression, most of the pixel values are changed and the original video frames will become like noise-like images, as shown in Fig. 2. To the best of our knowledge, the existing deep networks for video anomaly detection are performed on the decompressed video frame sequence, and heavily rely on the exact values of every pixel. Therefore, these schemes do not work in our application scenario. As shown in Section V-C, the performance of using AVID [16] and ALOCC [17] on the encrypted video dataset is not satisfactory.

*2) :* Another option using deep learning method is to train a new deep network directly from the encrypted video dataset. This approach may work in theory, as we have succeeded to do it with traditional feature extraction method. However, we will face a challenge to design this deep network in practice. Training a deep network directly on the encrypted video dataset is still an open problem. The encryption of the video data will undermine the possibility of training and reduce the accuracy of prediction. We have tested three training methods from deep learning methods AVID [16], and ALOCC [17]

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2929855, IEEE Transactions on Circuits and Systems for Video Technology



on the encrypted video dataset *UCSD*, respectively. As shown in Section V-C, the performance is not satisfactory. We have also tested the training methods AVID and ALOCC on a larger encrypted video dataset, which is the subset of *Subway* containing 64,999 frames. We use 34,999 frames for training and 30,000 frames for testing. The AUC and BER values are 0.51 and 0.50 for AVID, and 0.54 and 0.47 for ALOCC, respectively. The experimental results show that both the two deep learning methods are still not effective even with more training data.

*3) :* We have testing the deep-learning based approach CVPRW2015 [18] on the encrypted video dataset *UCSD-E480P*. As shown in Section V-C, the performance is also not satisfactory. In our application scenario, the video data is degenerated by the video encryption algorithm. The degenerated video data has affected the performance of the deep learning methods. Although the CVPRW2015 method has lower complexity compared with AVID and ALOCC, its parameters are still several orders of magnitude larger than our method. The suitable dataset for the CVPRW2015 method on the encrypted video may be also several orders of magnitude larger than the employed dataset. However, there is still not such public large-size dataset at this moment.

## VII. Conclusions

In this paper, we have proposed an anomaly detection and localization scheme for encrypted videos and demonstrated the parallel implementation. We exploit the residual information and codeword structure of format-compliant encrypted videos to estimate the probability of being an anomalous motion. The MB size in bits, the MB partition, and the magnitude of the MVD are extracted directly from the encrypted bitstream. Frame-level and localization features are derived from these parameters. The detection is performed directly on the encrypted video bitstream without decryption and full decoding. Moreover, we can select different video encryption algorithms. This characteristic makes the proposed scheme applicable to a wide range of fields. According to our experimental results, the detection and localization performance of our scheme is satisfactory. The proposed frame-level method outperforms the compared methods in terms of efficiency. Our algorithm also works for non-encrypted video bitstreams and maintains the same performance. The proposed parallel implementation demonstrated that we can accelerate the detection rate to improve the efficiency.

The proposed scheme can benefit many cloud applications with privacy protection requirements, such as cloud home care and segmentation and classification of encrypted videos in the cloud. Videos encrypted in the cloud do not have to be downloaded and decrypted for anomaly detection. The server can label normal and anomaly videos without decryption, and thus achieves a trade-off between privacy protection and service convenience.

## Acknowledge

The authors thank the reviewers and the editors for their valuable comments to improve this paper, and also thank Dr. Amit Adam for providing a new video dataset.


Fig. 15: Examples of the face detection and recognition results on the original and the encrypted video. (a) Detecting a human face at the region [74, 56, 110, 533]. (b) None human face detected. (c) Detecting a human face at the region [78, 60, 114, 569]. (d) None human face detected, (e) A known face. (f) An unknown face in a plaintext frame is recognized as the same face in (e). (g) The face recognition algorithm cannot recognize the encrypted face.

## Appendix

The proposed abnormal detection scheme can also protects the privacy of people in video. We have conducted experiment to perform an actual face detection algorithm [44] on the encrypted video and the corresponding original video in all the video datasets we used. Our experimental results show that we cannot detect any human faces from all the encrypted video with the same face detection algorithm. We show some examples of the experimental results in Fig. 15 (a)-(d). Furthermore, we assumed that we had already detected the human faces in the video, and performed an actual face recognition algorithm [45] on the encrypted video. We show an example of the experimental results in Fig. 15 (e)-(g). By comparing with a known face, we can see that the face recognition algorithm can correctly recognize the unknown face in the plaintext frame. However, by taking the corresponding encrypted frame as input, even manually locate the face region, the face recognition still cannot recognize who the person is.

## References

[1] Y. Zheng, X. Yuan, X. Wang, J. Jiang, C. Wang, and X. Gui, "Toward encrypted cloud media center with secure deduplication," *IEEE Trans. Multimedia*, vol. PP, no. 99, pp. 251–262, 2016.

[2] R. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 82–105, 2013.

[3] P. Zheng and J. Huang, "Discrete wavelet transform and data expansion reduction in homomorphic encrypted domain," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2455–2468, 2013.

[4] T. Bianchi and A. Piva, "Secure watermarking for multimedia content protection: A review of its benefits and open issues," *IEEE Signal Process. Mag.*, vol. 30, no. 2, pp. 87–96, 2013.

[5] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *Privacy Enhancing Technologies*. Springer, 2009, pp. 235–253.

[6] J. R. Troncoso-Pastoriza, D. Gonzalez-Jimenez, and F. Perez-Gonzalez, "Fully private noninteractive face verification," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 7, pp. 1101–1114, 2013.

[7] D. Xu, R.Wang, and Y. Q. Shi, "Data hiding in encrypted H.264/AVC video streams by codeword substitution." *IEEE Trans. Inf. Forensics Security*, pp. 596–606, 2014.

[8] J. Guo, P. Zheng, and J. Huang, "An efficient motion detection and tracking scheme for encrypted surveillance videos," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 13, no. 4, p. 61, 2017.

[9] C. Piciarelli, C. Micheloni, and G. L. Foresti, "Trajectory-based anomalous event detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1544–1554, 2008.

[10] L. Kratz and K. Nishino, "Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models," in *Proc. CVPR*. IEEE, 2009, pp. 1446–1453.

[11] V. B. V. Mahadevan, W. Li and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Proc. CVPR*, Jun. 2010, pp. 1975–1981.

[12] X. Wang, X. Ma, and E. Grimson, "Unsupervised activity perception by hierarchical bayesian models," in *Proc. CVPR*. IEEE, 2007, pp. 1–8.

[13] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Proc. CVPR*. IEEE, 2011, pp. 3449–3456.

[14] B. Antić and B. Ommer, "Video parsing for abnormality detection," in *Proc. ICCV*. IEEE, 2011, pp. 2415–2422.

[15] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, "Deep-cascade: Cascading 3D deep neural networks for fast anomaly detection and localization in crowded scenes," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1992–2004, 2017.

[16] M. Sabokrou, M. Pourreza, M. Fayyaz, R. Entezari, M. Fathy, J. Gall, and E. Adeli, "AVID: Adversarial visual irregularity detection," *Proc. Asian Conf. Comput. Vision*, pp. 488–505, 2019.

[17] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proc. CVPR*. IEEE, 2018, pp. 3379–3388.

[18] M. Sabokrou, M. Fathy, M. Hoseini, and R. Klette, "Real-time anomaly detection and localization in crowded scenes," in *Proc. CVPR Workshops*, 2015, pp. 56–62.

[19] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Comput. Vision Image Understanding*, vol. 172, pp. 88–97, 2018.

[20] Y. Fan, G. Wen, D. Li, S. Qiu, and M. D. Levine, "Video anomaly detection and localization via gaussian mixture fully convolutional variational autoencoder," *arXiv preprint arXiv:1805.11223*, 2018.

[21] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun, "Deep image: Scaling up image recognition," *arXiv preprint arXiv:1501.02876*, 2015.

[22] S. Biswas and R. Babu, "Real time anomaly detection in H.264 compressed videos," in *Proc. NCVPRIPG*, Dec 2013, pp. 1–4.

[23] S. Biswas and R. V. Babu, "Anomaly detection in compressed H.264/AVC video," *Multimedia Tools Appl.*, vol. 74, no. 24, pp. 11 099–11 115, 2015.

[24] N. Kiryati, T. Raviv, Y. Ivanchenko, and S. Rochel, "Real-time abnormal motion detection in surveillance video," in *Proc. ICPR*, 2008, pp. 1–4.

[25] S. Biswas and R. V. Babu, "Sparse representation based anomaly detection using homv in H. 264 compressed videos," in *Proc. SPCOM*. IEEE, 2014, pp. 1–6.

[26] H. Li, Y. Zhang, M. Yang, Y. Men, and H. Chao, "A rapid abnormal event detection method for surveillance video based on a novel feature in compressed domain of HEVC," in *Proc. ICME*. IEEE, 2014, pp. 1–6.

[27] R. Leyva, V. Sanchez, and C.-T. Li, "Video anomaly detection based on wake motion descriptors and perspective grids," in *Proc. WIFS*. IEEE, 2014, pp. 209–214.

[28] T. Stutz and A. Uhl, "A survey of H.264 AVC/SVC encryption," *IEEE Trans. Circuits Syst. Video Technol*, vol. 22, no. 3, pp. 325–339, March 2012.

[29] Z. Shahid, M. Chaumont, and W. Puech, "Fast protection of H.264/AVC by selective encryption of CAVLC and CABAC for I and P frames," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 5, pp. 565–576, 2011.

[30] I. E. Richardson, *The H.264 advanced video compression standard*. John Wiley & Sons, 2011.

[31] G. R. Terrell and D. W. Scott, "Variable kernel density estimation," *Ann. Statist.*, vol. 20, no. 3, pp. 1236–1265, 1992.

[32] G. McLachlan and D. Peel, *Finite mixture models*. John Wiley & Sons, 2004.

[33] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 1, pp. 3:1–3:39, 2012.

[34] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc. Series B.*, pp. 1–38, 1977.

[35] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Prob.*, vol. 1, no. 14. Berkeley, CA, USA., 1967, pp. 281–297.

[36] K. W. Cheng, Y. T. Chen, and W. H. Fang, "Gaussian process regression-based video anomaly detection and localization with hierarchical feature representation," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5288–5301, 2015.

[37] I. S. A. Adam, E. Rivlin and D. Reinitz, "Robust realtime unusual event detection using multiple fixed-location monitors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 555–560, Mar. 2008.

[38] V. Saligrama and Z. Chen, "Video anomaly detection based on local statistical aggregates," in *Proc. CVPR*, 2012, pp. 2112–2119.

[39] A. O. R. Mehran and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proc. CVPR*, Jun. 2009, pp. 935–942.

[40] M. J. Roshtkhari and M. D. Levine, "Online dominant and anomalous behavior detection in videos," in *Proc. CVPR*, Jun. 2013, pp. 2611–2618.

[41] O. Boiman and M. Irani, "Detecting irregularities in images and in video," *Int. J. Comput. Vision*, vol. 74, pp. 17–31, 2007.

[42] S. Lian, Z. Liu, Z. Ren, and H. Wang, "Secure advanced video coding based on selective encryption algorithms," *IEEE Trans. Consum. Electron.*, vol. 52, no. 2, pp. 621–629, 2006.

[43] K.-Y. Chu, Y.-H. Kuo, and W. H. Hsu, "Real-time privacy-preserving moving object detection in the cloud," in *Proc. ACM MM*, 2013, pp. 597–600.

[44] D. E. King, "Dlib-ml: A machine learning toolkit," *J. Mach. Learn. Res.*, vol. 10, no. Jul, pp. 1755–1758, 2009.

[45] ageitgey, "face recognition," 2019. [Online]. Available: https://github.com/ageitgey/face_recognition


**Jianting Guo** received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2013, where he is currently pursuing the Ph.D. degree. His current research interests include signal processing in the encrypted domain, video processing, and video encryption.

**Peijia Zheng** (S'10-M'14) received the B.S. degree in mathematics, in 2009, and the Ph.D degree in computer science, in 2014, from Sun Yat-sen University, Guangzhou, China.

He is currently a Lecturer with the School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China. He is also a member of the Guangdong Key Laboratory of Information Security and Technology. His research interests include multimedia security, image/video encryption, and encrypted image/video processing.

**Jiwu Huang** (M'98-SM'00-F'16) received the B.S. degree from Xidian University, Xian, China, in 1982, the M.S. degree from Tsinghua University, Beijing, China, in 1987, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1998. He was with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China. He is currently a Professor with the College of Information Engineering, Shenzhen University, Shenzhen, China.

His current research interests include multimedia forensics and security. He is also a member of the IEEE Circuits and Systems Society Multimedia Systems and Applications Technical Committee and the IEEE Signal Processing Society Information Forensics and Security Technical Committee. He was the General Co-Chair of the IEEE Workshop on Information Forensics and Security in 2013. He served as an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY from 2010 to 2014.