Moving Object Detection in HEVC Video

LieLin Pang* and KokSheik Wong†
* Faculty of Computer Science and Information Technology, Universiti Malaya, Malaysia.
E-mail: adpangll@siswa.um.edu.my
† School of Information Technology, Monash University Malaysia, Malaysia.
E-mail: wong.koksheik@monash.edu

***Abstract*—Video surveillance has drawn much interest in monitoring physical assets, spaces and events over time for detection of threats as well as business and process monitoring purposes. However, the rising number of recorded videos has significantly increased the time and effort in manual event analysis and video content management. Therefore, automatic moving object detection is of great importance. Nowadays, for storage and transmission purposes, video usually appears in the compressed form. Therefore, in this paper, an automatic moving object detection method is proposed for HEVC video. Specifically, the number of bits spent on coding a frame, which can be extracted during encoding or retrieved from an encoded video bit stream, is exploited as the key feature for moving object detection. In addition, temporal sub-layering feature of HEVC is utilized to reduce the number of frame to be processed, which in turn magnifies the energy of the coded video frames without losing most of the predicted information. A coarse background / foreground mask is then formed based on bit consumption, and it is further refined via post processing to remove noise and to smooth the mask image. The proposed method achieves encouraging results in detecting slow moving objects, even with dynamic background.**

I. Introduction

Moving object detection in video can be utilized for purposes of event analysis and content management. These applications include abnormal event detection [1], action recognition and classification [2], traffic monitoring [3], passenger counting [4], video indexing, content searching and retrieving [5], to name a few. The increased crime rate and security threat all around the globe have driven a continuous growth of video surveillance in various places including public areas, community infrastructures, commercial or private buildings. The videos recorded in Closed-Circuit TeleVision (CCTV) and Internet of Things (IoTs) for monitoring daily activities have resulted in a large volume of visual-based data. This ever-increasing volume of visual data has resulted in significant increase in time and effort required to search and retrieve the desired video contents. Moving object detection plays an important role to enable more effective surveillance processes, better content management, as well as more accurate and credible analysis.

Without loss of generality, moving object detection can be performed by using the raw video sequence (i.e., uncompressed) or the encoded video bit stream with partial decoding. For the former, Guo et al. propose a multi-layer adaptive block-based background subtraction and pixel-based classification method to detect moving object in the spatial domain [6]. The proposed method is capable of removing most of the dynamic background and solving the deficiency of blocking effect. Cuevas et al. apply a spatio-temporal non-parametric background model for moving object detection in video sequence recorded by a moving camera [7]. Lee et al. propose a background-subtraction method by using background sets to detect objects from dynamic background with the idea of image- and color-space reduction [8]. In another work, Huang et al. propose an optical flow based motion detection framework for real-time motion detection in non-stationary scenes by treating the distribution of the optical flow field for background as a quadratic function of the point coordinates [9]. While the performance is better in general, the methods designed for raw video sequence require higher computational power because of the large number of pixels to handle, as well as the extra space (even temporary) needed to store the decoded video frames.

On the other hand, many videos, if not all, are encoded in a compressed form these days by following certain standards such as MPEG-2 [10], H.264/AVC [11] and HEVC [12], which achieve a compression ratio of $31:1$, $500:1$ and $1000:1$, respectively. As such, researchers investigate into different techniques to achieve moving object detection by analyzing the encoded video bit stream directly. In this context, the syntax elements of the coding standard are commonly exploited to detect moving object. These syntax elements include motion vector (MV), block structure, prediction mode as well as transformed coefficient. Among them, MV is often analyzed for moving object detection due to its very purpose in video coding, i.e., motion estimation. For example, Li et al. derive motion intensity count from MVs to serve as an indicator for detecting any abnormal events [13]. Moriyama et al. propose to amplify the MVs by sub-sampling HEVC video sequence in the temporal axis and separate the objects from background by using adaptive thresholding [14]. Samaiya et al. utilize MV clustering and block partitioning modes to achieve foreground segmentation in surveillance HEVC video [15]. Jaballah et al. use syntax elements including MVs, block types and transform coefficients for object detection in video encoded in the H.264/AVC and HEVC formats [16]. Instead of relying on MV, Laumer et al. analyze the type of macroblock in H.264/AVC encoded video then assign weights to the macroblock types and partition modes to determine whether a block is classified as part of a moving object [17]. They also exploit temporal dependencies between frames to improve the

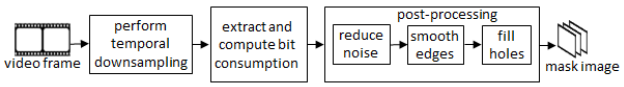


Fig. 1: Flow of moving object detection process.

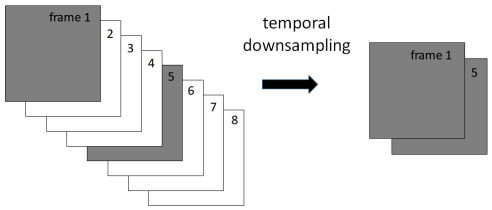


Fig. 2: Illustration of temporal downsampling with $\alpha = 2$.

detection accuracy. Recently, Alizadeh et al. propose a moving object detection method based on Conditional Random Field (CRF) for HEVC video [18]. In their work, MV, partitioning mode and the number bits spent on encoding a given block are extracted from the HEVC bit stream. After removing outlier MVs, the remaining MVs are copied to the I-blocks based on their neighboring blocks. The MV, partitioning mode and the number of bits spent are used as the input variables to the CRF model, which is updated for every frame in order to detect object.

Although many methods for moving object detection have been proposed, they are either (i) designed for the previous generation of video coding standards, or; (ii) complex in nature. HEVC achieves higher compression ratio in comparison to the previous video coding standards via effective removal of redundant information, both spatially and temporally. As a result, moving object detection is more challenging in HEVC since most of the information is removed by the more sophisticated prediction and motion estimation techniques introduced in HEVC. In particular, there are limited methods focusing on fast feature extraction for moving object detection in HEVC video, which is a format expected to gain more market share in the near future. Hence, this paper proposes a light-weight moving object detection method in HEVC video, where the output can be subsequently used in various computer vision applications. Specifically, the number of bits spent on coding the video is extracted as the feature to form a coarse background / foreground mask. Subsequently, post processing (i.e., morphological operations) is performed to remove noise and to smooth the mask image. The proposed method involves lightweight processes which only need to compute the number of bits spent on coding each coding units (CUs), which is straightforward to implement.

II. Proposed method

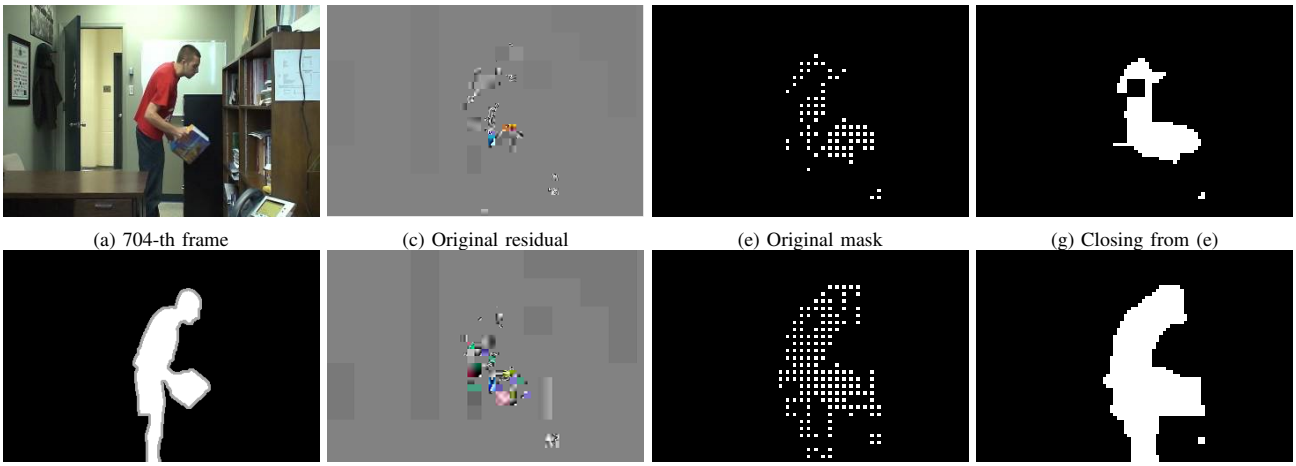
In this work, we put forward a moving object detection method based on the number of bits spent (i.e., bit consumption) on coding a HEVC video [12]. The general flow of processes in the proposed method is as illustrated in Fig. 1. Recall that HEVC adopts quadtree-based variable block size block partitioning structure, where each video frame is partitioned into multiple blocks called coding tree units (CTUs). Specifically, CTU is the largest coding block used for prediction. Each CTU is subsequently split into multiple CUs, where the size ranges from 64×64 to 8×8. In general, the amount of bits consumed by a block is highly correlated to the predicted information and the prediction error/residual energy. The cost of prediction and coding the residual depends on the video content and it varies from block to block. Basically, a block with higher bit consumption indicates that it has more energy, and vice versa. Therefore, bit consumption for each coded block can be extracted to infer some information about the energy due to activity, detail, texture and motion in the video frame, which can be subsequently utilized for moving object detection. However, there is a situation where information is missing or insufficient (e.g., prediction error/residual) in the compressed video, especially for video coded at high frame rate. Specifically, when most CUs are coded with a small number of bits or even no bits at all, only a portion of the object or nothing at all can be detected. Since HEVC standard supports the temporal sub-layering feature, temporal downsampling can be performed during encoding to reduce the frame rate, which in turn magnify the energy of the coded video frames without losing most of the predicted information.

The temporal down-sampling factor α reduces the resolution of the time axis t through $\Delta t = t/2^{\alpha}$. An illustration of temporal down-sampling in reducing temporal resolution of an input stream is depicted in Fig. 2, where every 4-th frame is processed. The prediction error/residual is formed by computing the difference between the original block and its predicted information. To address the challenge of limited information in the compressed domain, we examine and compare the pixels reconstructed from the residual. Unlike the pixel reconstructed from the original residual as shown in Fig. 3(c), the residual after down-sampling with $\alpha = 2$ as shown in Fig. 3(d) can provide more detail. The energy from the residual is further enriched by information extracted from the prediction parameters, which in turn results in more detail to form a better mask image as shown in Fig. 3(f). The morphological closing operation is then performed to obtain the final mask, where Fig. 3(g) and (h) are produced from Fig. 3(e) and (f), respectively.

Next, the number of bits consumed for encoding the (i, j)-th CU, i.e., $b_{i,j}$, is computed. Here, the number of bits consumed for encoding a CU over the entire frame is non-linearly scaled by using the logarithmic function. The logarithmic values have a wider range and they are more distinguishable for further analysis based on energy consumption per CU. In particular, the ratio $\varepsilon_{i,j}$ is computed as

$$\varepsilon_{i,j} = \frac{255 \times \log_2(b_{i,j})}{\max\{\log_2(b_{i,j})\}}. \quad (1)$$

For our objective of detecting moving object, the (i, j)-th block in the k-th frame $f^k_{i,j}$ is set as either foreground ('1')



(a) 704-th frame (c) Original residual (e) Original mask (g) Closing from (e)

(b) Ground truth (d) Residual after down-sampling (f) Mask after down-sampling, f_b^k (h) Closing from (f), $f_{b,c}^k$

Fig. 3: Comparison of residual energy, intermediate mask and final output mask before and after down-sampling.

when $\varepsilon_{i,j} \neq 0$ or background ('0') when $\varepsilon_{i,j} = 0$. In essence, f^k is a mask (binary image) of dimension $M \times N$ when the dimension of the input HEVC video is $4M \times 4N$.

Subsequently, the post-processing tasks are performed to smooth the edges, fill the holes of the detected objects and remove noise in order to improve the detection accuracy of the moving object. Specifically, the gray level of each pixel is replaced by the median of the pixels based on the 3×3 neighborhood. Subsequently, a bilateral filter [19] is utilized to suppress the variation of intensity value from one pixel to another in f^k. Specifically, it replaces the intensity of each pixel with a weighted average of the intensity values from its adjacent pixels, where the weights are inversely proportional to the distance from the center of the neighborhood. Finally, the closing operation (i.e., dilation followed by erosion) of the image $f_b^k(i,j)$ is performed by a 3×3 rectangular structuring element (denoted by s) on the detected object to smooth the contour of the object and to fill small gaps within the foreground objects. Here, the closing of the image $f_b^k(i,j)$ is defined as:

$$f_{b,c}^k(i,j) \cdot s = (f_b^k(i,j) \oplus s) \ominus s, \tag{2}$$

where '$\cdot$', '$\oplus$', and '$\ominus$' denote the closing, dilation and erosion operations, respectively. Figure 4 shows the transition of the mask image for the 28-th frame in the test video *foreman*, starting from its first appearance f^{28} to the final output $f_{b,c}^{28}$. It can be observed that the mask image f_b^{28} (i.e., after applying filtering) has smoother edges. Subsequently, the small holes in the object is filled in the mask image $f_{b,c}^{28}$, i.e., after applying the closing operation.

The proposed method overcomes the challenge of lack of information (e.g., prediction error/residual) in the encoded video bit stream. Our method is also computational simple, and hence the process is greatly beneficial for energy-constrained devices such as those used in smart home appliances. Note that although both the proposed method and Alizadeh et al.'s method [18] consider the number of bits spent on coding a CU, our approaches are different in the following manner: 1. we apply temporal down-sampling to increase the variation of bits consumed in encoding CUs in addition to reducing the number of frames to be processed, while [18] applies partitioning mode and MV to improve the detection accuracy; 2. we utilize median and bilateral filters to reduce noise and to smooth the edge, while [18] removes the global motion based on the difference between MV in a block and the average MVs in the frame and it also utilizes iterated conditional mode to optimize the classification results, and; 3. we perform morphological operation to construct the output mask image, while [18] calculates the weighted average MVs of the neighboring blocks in order to assign a MV to the intra-coded block.

III. Experimental Results

The proposed method is implemented by modifying the HEVC reference software HM-16.22 [20]. For experiment purposes, the following setting is used: Low-delay P (LDP) main configuration settings with group of pictures (GOP) structure of 4, frame rate of 30 fps, and the quantization parameter (QP) set to 30. The remaining parameters are set to the HM default configuration. In addition, the temporal down-sampling factor $\alpha = 2$ is set. The experiments are conducted by using a PC with AMD Ryzen 5 3500U 2.10 GHz CPU and 8GB of RAM running on a 64-bit Windows 10 operating system. The video sequences from the CDNET 2014 video dataset [21], [22] and [23] are utilized to evaluate the performance. These sequences include *BusStation*, *Office*, *Pedestrians*, *PeopleInShade*, *PETS2006*, *Foreman*, *BasketballDrill* and *BasketballDrive*. The results are measured in term of

(a) Original 28-th frame (b) Mask image, f^{28} (c) Smoothing filter, f_b^{28} (d) Closing, $f_{b,c}^{28}$ (e) Ground truth

Fig. 4: Transition from the original video frame to the final output mask and ground truth for the 28-th frame in the Foreman test video sequence by applying median filter, bilateral filter and closing.

TABLE I: Comparison of different methods in the terms of F_1, Precision and Recall.

Video sequence	Method	Precision	Recall	F_1
Office	[17]	0.72	0.50	0.59
(360 × 240)	[15]	0.96	0.43	0.59
	Proposed	0.83	0.74	0.77
BusStation	[17]	0.69	0.63	0.66
(360 × 240)	[15]	0.81	0.29	0.42
	Proposed	0.80	0.83	0.81
Pedestrian	[17]	0.31	0.96	0.47
(360 × 240)	[15]	0.97	0.39	0.55
	Proposed	0.75	0.86	0.80
PeopleInShade	[17]	0.72	0.82	0.77
(360 × 240)	Proposed	0.70	0.86	0.77
PETS2006	[17]	0.59	0.65	0.63
(720 × 576)	[15]	0.94	0.48	0.63
	Proposed	0.72	0.74	0.73
Foreman	[17]	0.68	0.99	0.81
(352 × 288)	[18]	0.89	0.92	0.91
	Proposed	0.87	0.78	0.82
BaseketballDrill	[18]	0.81	0.88	0.84
(832 × 480)	Proposed	0.64	0.87	0.74
BasketballDrive	[18]	0.87	0.91	0.89
(1280 × 720)	Proposed	0.71	0.86	0.78

Precision, Recall and F_1, which are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

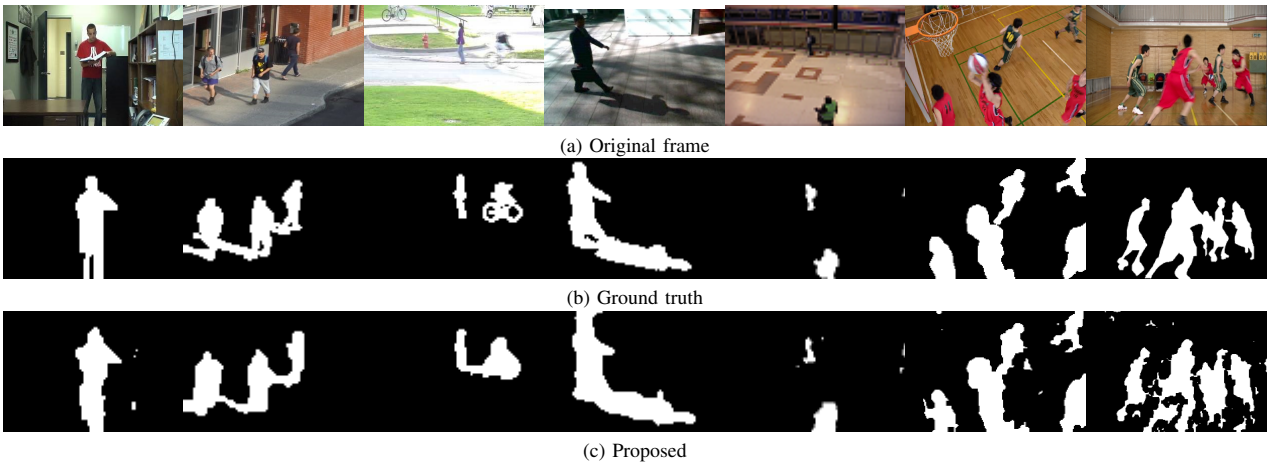
$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{5}$$

where TP, FP, and FN denote the true positive, false positive, and false negative, respectively.

The results are recorded in Table I. On average, our precision, recall and F_1 are 0.76, 0.83, and 0.78, respectively. It is observed that the proposed method achieves better performance for video sequence containing small to medium moving objects. A potential reason is that the background in these videos is coded by using relatively larger blocks and hence less bits are consumed in coding the residual of the motion estimated background regions. In other words, blocks with lower number of bits are treated as background. In contrast, the moving object is more complex in term of texture when it is smaller in size. More importantly, the energy of the moving object is magnified after the temporal down-sampling is performed. Therefore, the distribution of the number of bits consumed for coding CUs is concentrated in the moving object. It should be noted that α can take a different value (e.g., 1, 2 and 3) but the results do not change significantly (viz., $\pm 3\%$). In fact, when α increases, the proposed method can be executed in a shorter period of time since there are less frames to process, and vice versa. However, it is suggested to set α to process $< 50\%$ of the frame.

For completion of discussion, we also compared our results against those achieved by [15], [17] and [18]. Results suggest that the proposed method outperforms Samaiya et al.'s method [15] and Laumer et al.'s method [17] in term of accuracy. Our performance is inferior in comparison to Alizadeh et al.'s method [18], where our performance is lower by 0.14, 0.09 and 0.12 for precision, recall and F_1, respectively. However, it is noteworthy that our method is computationally simpler and straightforward, which can beneficial for applications requiring low turn-around time as well as devices with constrained computational power or small battery capacity. In contrast, Alizadeh et al.'s method [18] considers more features including partitioning mode, bit consumption, and motion vector. Furthermore, more complex processes are utilized to remove outliers. As an example, our proposed method requires an average of 0.18s while [18] requires 0.75s for processing a HD frame. Moreover, to improve the detection accuracy, the CTU size in [18] is set to 32×32 when experimenting with low resolution videos (e.g., SD and CIF formats), because a larger CTU allows larger CUs to be coded, which will reduce the detection accuracy for any object smaller than a CU. On the other hand, in our experiments, the CTU size remains the same, i.e., 64×64, and our proposed method still performs well. It should be noted that the ground truth for the video sequences *foreman*, *BasketballDrill* and *BasketballDrive* are not available and hence we created our own ground truth (available for download at [24]), which could be a potential factor for our method being inferior. All in all, although the proposed method exhibits mixed outcomes when compared to the conventional methods, our method is based on simple operations which are light-weight in nature.



(a) Original frame

(b) Ground truth

(c) Proposed

Fig. 5: Video sequences considered for experiments in this work. From left to right column: *Office*, *BusStation*, *Pedestrian*, *PeopleInShade*, *PETS2006*, *BasketballDrill* and *BasketballDrive*).

IV. Conclusion

In this work, temporal sub-layering feature and bit size consumption in encoding the coding units are exploited to detect moving object in HEVC video. The proposed method overcomes the challenge of lack of information in the encoded HEVC video bit stream. Specifically, the energy of the coded video frames is magnified by reducing the number of video frames to be processed by using temporal sub-layering.

The feature, i.e., number of bits consumed in coding CU, is expanded by using a logarithmic function to highlight the differences of bit consumption per CU. After the coarse mask is constructed, post-processing operations are performed to remove noise and to smooth the extracted objects. The proposed method has simple computations and operations, and hence it is overall a light-weight method. Experimental results demonstrate that the proposed method can effectively detect moving object, and it performs well in video with moving object of small to medium sizes. In addition, the proposed method exhibits comparable performance when it is benchmarked against the conventional methods.

For future work, we aim to improve the precision of the proposed method and explore potential deployment of the proposed method for real-time or time-critical applications. We also want to explore moving object detection in other video coding standards such as VVC and AV2 by using the bit consumption feature as a starting point.

Acknowledgment

This research is supported by the Fundamental Research Grant Scheme (FRGS) MoHE Grant under project - Background/foreground separation in compressed domain (Grant No. FP063-2014A).

References

[1] Y. S. Chong and Y. H. Tay, "Abnormal event detection in videos using spatiotemporal autoencoder," in *Advances in Neural Networks - ISNN 2017*, F. Cong, A. Leung, and Q. Wei, Eds. Cham: Springer International Publishing, 2017, pp. 189–196.

[2] M. Sharif, M. A. Khan, T. Akram, M. Y. Javed, T. Saba, and A. Rehman, "A framework of human detection and action recognition based on uniform segmentation and combination of euclidean distance and joint entropy-based features selection," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, p. 89, 2017.

[3] J. Zhou and C. Kwan, "Anomaly detection in low quality traffic monitoring videos using optical flow," in *Pattern Recognition and Tracking XXIX*, vol. 10649. International Society for Optics and Photonics, 2018, p. 106490F.

[4] R. Sutopo, J. Lim, V. M. Baskaran, K. Wong, M. Tistarelli, and H. F. Liau, "Appearance-based passenger counting in cluttered scenes with lateral movement compensation," *Neural Computing and Applications*, vol. 93, pp. 583–595, 2021.

[5] L. F. Sikos, "Rdf-powered semantic video annotation tools with concept mapping to linked data for next-generation video indexing: a comprehensive review," *Multimedia Tools and Applications*, vol. 76, no. 12, pp. 14 437–14 460, 2017.

[6] J. Guo, C. Hsia, Y. Liu, M. Shih, C. Chang, and J. Wu, "Fast background subtraction based on a multilayer codebook model for moving object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 10, pp. 1809–1821, 2013.

[7] C. Cuevas, R. Mohedano, and N. García, "Statistical moving object detection for mobile devices with camera," in *2015 IEEE International Conference on Consumer Electronics (ICCE)*, 2015, pp. 15–16.

[8] H. Lee, H. Kim, and J. Kim, "Background subtraction using background sets with image- and color-space reduction," *IEEE Transactions on Multimedia*, vol. 18, no. 10, pp. 2093–2103, 2016.

[9] J. Huang, W. Zou, Z. Zheng, and J. Zhu, "An Efficient Optical Flow Based Motion Detection Method for Non-stationary Scenes," in *2019 Chinese Control And Decision Conference (CCDC)*, 06 2019, pp. 5272–5277.

[10] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital video: an introduction to MPEG-2*. Springer Science & Business Media, 1996.

[11] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[12] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[13] 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 *2014 IEEE International Conference on Multimedia and Expo (ICME)*, 2014, pp. 1–6.

[14] M. Moriyama, K. Minemura, and K. Wong, "Moving object detection in HEVC video by frame sub-sampling," in *2015 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2015, pp. 48–52.
[15] D. Samaiya and K. Gupta, "Intelligent video surveillance for real time energy savings in smart buildings using HEVC compressed domain features," *Multimedia Tools and Applications*, vol. 77, 11 2018.
[16] S. Jaballah and M. Larabi, "Fast object detection in H264/AVC and HEVC compressed domains for video surveillance," in *2019 8th European Workshop on Visual Information Processing (EUVIP)*, 2019, pp. 123–128.
[17] M. Laumer, P. Amon, A. Hutter, and A. Kaup, "Moving Object Detection in the H.264/AVC Compressed Domain," *APSIPA Transactions on Signal and Information Processing*, vol. 5, pp. 1–20, 2016.
[18] M. Alizadeh and M. Sharifkhani, "Compressed Domain Moving Object Detection Based on CRF," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 3, pp. 674–684, 2020.
[19] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 1998, pp. 839–846.
[20] HM-16.22, accessed Feb 1, 2017, https://hevc.hhi.fraunhofer.de/.
[21] CDW-2014, "Changedetection.net (cdnet)," http://jacarini.dinf.usherbrooke.ca/dataset2014/.
[22] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "A novel video dataset for change detection benchmarking," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4663–4679, 2014.
[23] Universität-Hannover, "Test sequence," ftp://ftp.tnt.uni-hannover.de/testsequences/.
[24] L. Pang, accessed Feb 25, 2021, https://github.com/LLPang/.