IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 12, 2018, accepted October 4, 2018, date of publication October 16, 2018, date of current version November 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2876427

Saliency Detection With Features From Compressed HEVC

WEI ZHOU, RUI BAI, AND HENGLU WEI
School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China

Corresponding author: Wei Zhou (zhouwei@nwpu.edu.cn)

This work was supported by the National Natural Science Foundation of China (61602383 and 61772424) and the Fundamental Natural Science Research Funds of Shaanxi Province (2018JQ6016 and 2017JQ6019).

ABSTRACT In this paper, a saliency detection algorithm with features from compressed high-efficiency video coding (HEVC) is proposed. The proposed algorithm consists of three parts: static saliency detection, dynamic saliency detection, and competitive fusion. Static features are generated by downsampling and discrete cosine transform, and dynamic features are extracted from compressed HEVC, specifically motion vector. A Gaussian kernel is used to extract the data structure in static feature maps. For dynamic feature map, a coding unit depth and bits combined mask is designed to filter out the dynamic background. Finally, competitive fusion is designed to adaptively fuse the static and dynamic saliency maps. Experimental results show that the proposed method is superior to classic methods by up to 0.1223 area under curve gaining and 0.8362 Kullback–Leibler divergence decreasing on average. The average detection speed is 2.3 s per frame.

INDEX TERMS HEVC, saliency, static feature, dynamic feature.

I. INTRODUCTION

Visual saliency model aims to detect areas of concern to human eyes and filter out unimportant areas [1]. Visual saliency detection is used in various areas, such as object detection [2], object recognition [3], image re-targeting [4], image quality assessment and image/frame compression [5].

Various saliency detection models have been developed in the literature. These models are separated according to two mechanisms: bottom-up and top-down. The bottom-up refers to low level visual features and data-driven fast processing. Koch and Ullman [6] put forward a very influential biological inspiration model. Itti *et al.* [7] extracted low-level features of intensity and color to detect static image saliency regions. Harel *et al.* [8] formed activation maps on certain feature channels, and then normalized them in a way which highlighted conspicuity and admitted combination with other maps. Itti and Baldi [9] proposed a formal Bayesian definition of surprise to capture subjective aspects of sensory information and implemented a simple computational model. The top-down refers to slow processing based on task-driven and conscious control. Existing top-down models are designed to learn prior knowledge firstly, and then use prior knowledge to guide saliency detection. Hou and Zhang [10] presented a fast Fourier spectrum residual method. Marchesotti *et al.* [11] used Bayesian framework to calculate image saliency. Most of top-down saliency detection models need to learn large database of images, and the computation is huge.

Feature extraction plays a critical role in saliency detection. The performance of a saliency detection model mainly relies on how well the extracted features coincide with the human visual system (HVS). Traditionally, only static features are needed to detect saliency in images. But to detect saliency in videos, not only static features but also dynamic features should be extracted. The necessity of dynamic features in videos comes from that the interesting areas in the scene changes along with the movement of foreground [12], [13]. All the above saliency detection models extract features from uncompressed images or videos. However, almost all images and videos are stored in coding standard compatible format, such as JPEG, H.264, MPEG4 and HEVC. To extract features from such compressed images or videos, it is necessary to decompress them firstly, which burdens the saliency detection system a lot as both decompression and saliency detection are time consuming. Several studies tried to extract features from compressed images [14] and videos [15]. For the latest video coding standard, very few saliency models are designed. Xu *et al.* [16] established eye tracking data sets and detected video saliency with HEVC features. However, many top-level features were used in their algorithm, which introduced huge

computing complexity and caused unsuitability for real-time system.

In this paper, a saliency detection algorithm for compressed HEVC videos is proposed. Our method includes static saliency detection, dynamic saliency detection and competitive fusion. Firstly, the static features which include chroma, luminance and texture channels are extracted by down-sampling of color components and the DCT coefficients of Y component. Then, static features are filtered by Gaussian filter to detect static saliency map. Next, the dynamic feature is extracted using motion vector (MV). The dynamic feature is then filtered with a mask to filter out the dynamic background and the dynamic saliency map is detected. Finally, competitive fusion method is proposed to fuse the static saliency map and dynamic saliency map into one map.

The rest of this paper is organized as follows. Section II presents the proposed saliency detection algorithm. Experimental results are presented and discussed in Section III. Section IV concludes the works in this paper.

II. THE PROPOSED ALGORITHM

In this section, the proposed saliency detection algorithm with features from compressed HEVC is introduced. The framework of the proposed algorithm is shown in Fig. 1. Static saliency detection and dynamic saliency detection performed separately, and the detected static saliency map and dynamic saliency map are finally fused into one map by adaptive fusion.

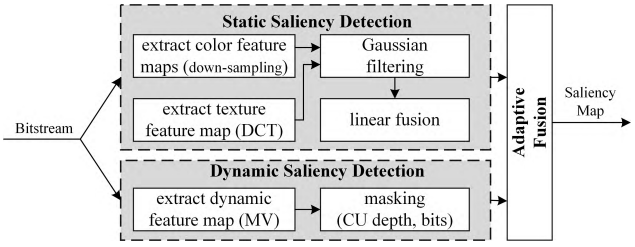


FIGURE 1. Framework of detecting saliency from HEVC compressed bitstream.

A. STATIC SALIENCY DETECTION

In HEVC, the format of input video sequence is YCbCr which contains three color channels Y, Cb and Cr. For each color channel, a feature map is extracted. In addition, a texture feature map is also extracted. The final static saliency map is obtained by filtering and fusing these four static feature maps.

1) EXTRACT STATIC FEATURES

The static features include one luminance feature map, two chroma feature maps and one texture feature map. The luminance feature map and chroma feature maps are extracted by down-sampling each color channel directly. Firstly, divide the frame into 8×8 blocks which is the same as the smallest coding unit (CU) size in HEVC. Each block contains a 8×8 luminance component Y and two 4×4 chroma components Cb and Cr. Then, three color features can be obtained by averaging the corresponding color component in an 8×8 block, as the following,

$$\begin{cases} \mathbf{FL}\,(i,j) = mean\left(\mathbf{Y}_{(i,j)}\right) \\ \mathbf{FCb}\,(i,j) = mean\left(\mathbf{Cb}_{(i,j)}\right) \\ \mathbf{FCr}\,(i,j) = mean\left(\mathbf{Cr}_{(i,j)}\right) \end{cases} \tag{1}$$

where $\mathbf{Y}_{(i,j)}$ is the Y color component of (i,j)-th 8×8 block in the frame, and *mean* is averaging operation which outputs a real-value. Finally, three color feature maps, i.e., **FL**, **FCb** and **FCr**, are extracted.

Texture feature is extracted from low frequency Discrete Cosine Transform (DCT) coefficients. Studies have shown that alternating current (AC) DCT coefficients can well represent texture information for image blocks [17]. In fact, most of the energy is concentrating in the left upper corner of the DCT coefficients matrix, and high frequency coefficients contain little texture information compared with low frequency coefficients. What's more, HVS is not sensitive to the high-frequency image [15]. Therefore, only 5 low frequency AC coefficients are used in the proposed algorithm to reduce complexity, as shown in Fig. 2. In addition, as Y color channel contains more texture information and HVS is more sensitive to Y color channel, texture feature is extracted only from Y color channel in the proposed algorithm, as the following,

$$\begin{aligned} \mathbf{FT}\,(i,j) = \{&\mathbf{B}_{(i,j)}\,(0,1)\,,\mathbf{B}_{(i,j)}\,(1,0)\,, \\ &\mathbf{B}_{(i,j)}\,(2,0)\,,\mathbf{B}_{(i,j)}\,(1,1)\,,\mathbf{B}_{(i,j)}\,(0,2)\} \end{aligned} \tag{2}$$

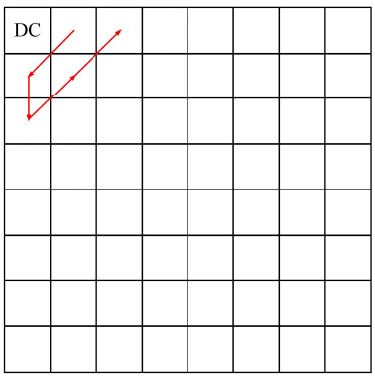


FIGURE 2. The chosen AC coefficients.

where $\mathbf{B}_{(i,j)}$ is the DCT coefficients matrix of the Y component of the (i,j)-th 8×8 block. Different from the above three color features which are real-values, texture feature is the real-valued vector. All texture feature vectors in a frame constitute the texture feature map, i.e., **FT**.

2) GAUSSIAN FILTERING

In bottom-up driven saliency detection, salient objects are detected from the data without any background, i.e., stimulus-driven. In this sense, signals stand out its surrounds are more

likely to attract more attention, i.e., the similarity of one point to its surrounds can be regarded as the likelihood of saliency. Therefore, static saliency in this paper for each feature map is detected by comparing the similarity of one feature to its surrounds. Just like in many other works, we use Gaussian kernel here to extract data structure. Specifically, each point in the static feature maps is considered as the center point and influenced by its surrounding points. The influence of each surrounding point to the center point is evaluated by Gaussian kernel, as the following,

$$\alpha_{(x_s,y_s)}=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x_s-x_c)^2+(y_s-y_c)^2}{2\sigma^2}\right) \quad (3)$$

where σ is constant and set as 40; (x_c, y_c) is the coordinate of center point; (x_s, y_s) is the coordinate of surrounding point. The influence of all surrounding points represent the static saliency value of center point. The static saliency for each center point is calculated by (4),

$$\mathbf{S_F}(x_c, y_c) = \sum_{x_s=0}^{w}\sum_{y_s=0}^{h}\left(\alpha_{(x_s,y_s)}|\mathbf{F}(x_s, y_s) - \mathbf{F}(x_c, y_c)|\right) \quad (4)$$

where $\mathbf{F} \in \{\mathbf{FL}, \mathbf{FCb}, \mathbf{FCr}, \mathbf{FT}\}$ is the static feature map; w and h are the width and height of the static feature map.

Then, final static saliency map can be obtained by linear weighting four static saliency maps, as following,

$$\mathbf{S_S} = \frac{\mathbf{S_{FL}} + \mathbf{S_{FCb}} + \mathbf{S_{FCr}} + \mathbf{S_{FT}}}{4} \quad (5)$$

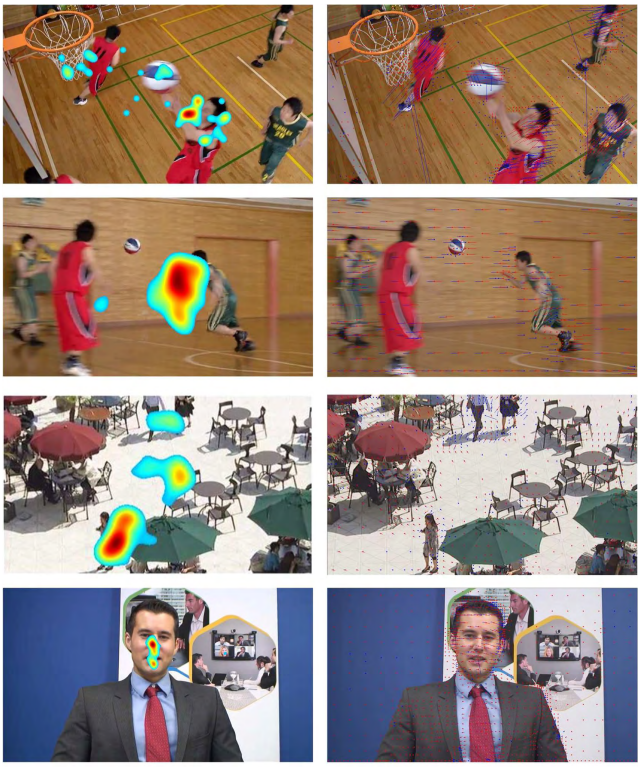
B. DYNAMIC SALIENCY DETECTION

The moving parts of videos are mainly concerns for HVS, therefore the moving information of objects in the frame is extracted as the dynamic feature. We use such dynamic feature to detect dynamic saliency. In addition, a mask based on coding information is designed to filter out dynamic background in dynamic feature map.

1) EXTRACT DYNAMIC FEATURE

When people look at the video, most attention are paid to the part of the movement [15]. In video coding, inter prediction is used to predict the current frame using the coded frame. The displacement frame reference block to current block is called MV. For video with static background, the background contains only few codirectional MVs, while the moving objects contain various unidirectional MVs. An example is shown in Fig. 3 where Fig. 3 (a) shows the human fixations extracted by eye tracker and Fig. 3 (a) shows the distribution of MVs. It can be seen from Fig. 3 that audiences tend to pay more attention to areas with complex movement. Therefore, it is efficient to distinguish the salient moving objects and non-salient background by the relative value of MV. In this paper, MV is used as the dynamic feature.

After coding, each block of 4×4 corresponds to a MV, all points in one 4×4 block share the same MV. Therefore, MVs are extracted per 4×4 block, and the length of every



(a) (b)

FIGURE 3. The relationship between human fixations and MVs: (a) human fixations (b) MVs.

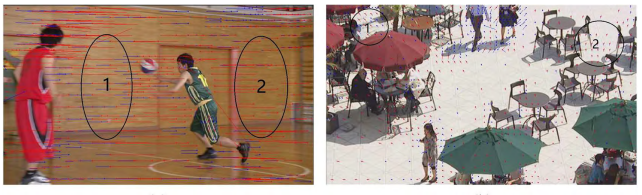
MV constitutes the dynamic feature map, as following,

$$\mathbf{FD}(i, j) = \left|\overrightarrow{V_{(i,j)}}\right| \quad (6)$$

where (i, j) is the coordinate of 4×4 block, and $\overrightarrow{V_{(i,j)}}$ is the MV of the (i, j)-th 4×4 block.

2) FILTER OUT DYNAMIC BACKGROUND

In general, audiences mainly concern foreground in the movie, e.g., figures, cars, animals, and such foreground usually moves with relatively still background. But for dynamic background scenarios, the movement of background is comparable to foreground, which may result in pseudo salient areas. Take the 56th frame of BasketballPass for example, as shown in Fig. 4 (a), there exists a large amount of MVs in background 1 and background 2. These MVs in dynamic



(a) (b)

FIGURE 4. The MVs of background: (a) BasketballPass (b) BQSquare.

background may introduce pseudo salient areas. To get a more accurate dynamic saliency map, dynamic background must be filtered out. In this paper, the CU depth and coding bits are used to filter out pseudo salient areas introduced by dynamic background.

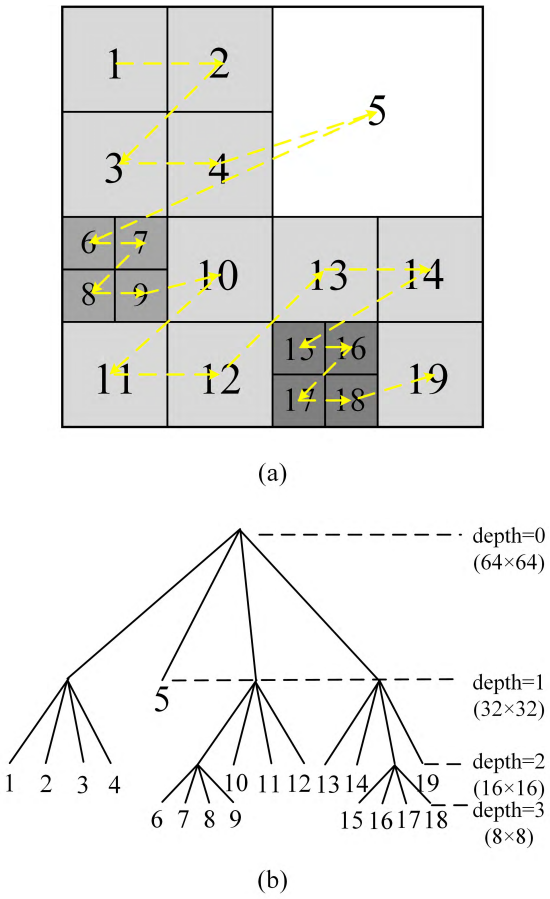


FIGURE 5. Quad-tree structure in HEVC: (a) the structure of a CTU (b) quad-tree of the CTU.

HEVC takes quad-tree as basic coding structure, the root of which is called coding tree unit (CTU). A CTU can be recursively split into multi depth CUs, as shown in Fig. 5 (a). The quad-tree representation of the CTU in Fig. 5 (a) is shown in Fig. 5 (b). HEVC defines the smallest CU as 8×8, which means the 64×64 CTU has 3 depth layers at most. As shown in Fig. 5, depth 0 corresponds to the root of quad-tree, i.e., CTU, depth 1 corresponds to 32×32 CU, depth 2 corresponds to 16×16 CU, and depth 3 corresponds to 8×8 CU, i.e., the smallest CU. The structure of CTU is tightly related with the video content. In general, areas of rich movement tend to be encoded by small CUs to improve the accuracy of motion estimation. In contrast, smooth background tends to be encoded by large CU to improve the coding efficiency. For dynamic background, the movement in it is almost directional and simple, while the movement in foreground is unidirectional and complexity. Therefore, dynamic background also tends to be coded by large CUs.

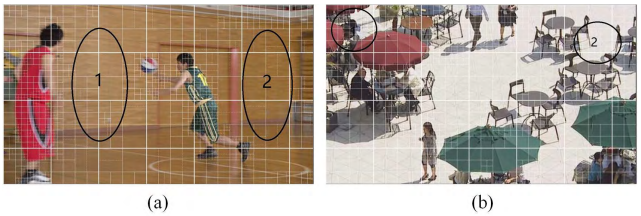


FIGURE 6. Comparison of CU sizes between background and foreground: (a) BasketballPass (b) BQSquare.

An example is shown in Fig. 6, where the circled parts are background. Although the MVs in background are comparable to those in foreground, the CU size in background is obviously larger than that in foreground. As there is a one-to-one corresponding relation between CU size and CU depth, we use CU depth to filter out the dynamic background in dynamic feature map.

Besides CU depth, coding bit is another feature to distinguish foreground and background. In recent years, some researches try to model saliency as the conditional entropy [18], [19]. From an uncertainty or informativeness point of view, the conditional entropy measures the remaining uncertainty of the center once its surrounds are known, or the amount of information of the center given the knowledge of its surrounds. This model can be easily explained by video coding language. In HEVC, Inter frames are estimated from several previously reconstructed frames, and only the difference between the original frame and the estimated frame is coded. Areas with complex movement are hard to be accurately estimated and will consume more bits to be encoded. In contrast, background, even dynamic background contains only few movement or just simple unidirectional movement. Such background is very similar to that in the previous frames, and few bits are required to encode it. Therefore, coding bits, i.e., the conditional entropy, can represent the relative saliency. The heat map of bits distribution is shown in Fig. 7, where the circled part 1 and part 2 are background and part 3 is foreground. As the background parts contain relatively little information, the consumed bits are relatively few, while the foreground consumes more bits. In the proposed algorithm, we use coding bits to filter out background in dynamic feature map.

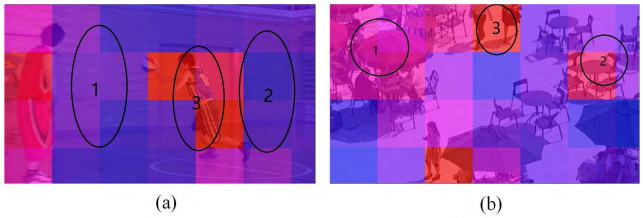


FIGURE 7. Heat map of bits: (a) BasketballPass (b) BQSquare.

Combined CU depth with coding bits, a binary mask is designed to filter the dynamic background, as

the following,

$$\mathbf{M}(i,j) = \begin{cases} 0, & d_{(i,j)} \times b_{(i,j)} < Th \\ 1, & d_{(i,j)} \times b_{(i,j)} > Th \end{cases} \tag{7}$$

$$Th = \frac{p_0 \times \left(\sum_{(i,j) \in frame} d_{(i,j)} \times b_{(i,j)} \right)}{N} \tag{8}$$

where $d_{(i,j)}$ is the CU depth of the (i,j)-th pixel; $b_{(i,j)}$ are the consumed bits; N is the total pixel number in the frame; p_0 is a parameter indicating filtering strength.

With the binary mask, the dynamic saliency map can be detected as the following,

$$\mathbf{S_D} = Norm(\mathbf{M} \circ \mathbf{FD}) \tag{9}$$

where $\circ$ is Hadamard product; *Norm* is the normalization operation.

C. COMPETITIVE FUSION ALGORITHM

The static saliency and dynamic saliency map are combined by fusion algorithm. In this paper, an adaptive fusion algorithm based on competition is presented as the following,

$$\mathbf{S} = Norm(a_1 \cdot \mathbf{S_S} + a_2 \cdot \mathbf{S_D} + a_3 \cdot \mathbf{S_{SD}}) \tag{10}$$

where $\mathbf{S_{SD}} = \mathbf{S_S} \circ \mathbf{S_D}$ is the mixed map; a_1, a_2 and a_3 are the parameters to control the weight of static, dynamic and mixed map, respectively. The parameters a_1, a_2 indicate the weight of static saliency map, dynamic saliency map and fused saliency map. As static saliency map is detected only by physical stimulation, the distribution of salient pixels in static salient map is dispersive and some background information can not be successfully filtered, as shown in Fig. 10. In contrast, dynamic saliency map is detected by moving objects and the distribution of salient pixels in dynamic saliency map is more concentrated, as shown in Fig. 11, which is similar as that of human fixations, as shown in Fig. 3. Therefore, dynamic saliency map plays a more important role to construct the final saliency map. In this paper, we use the standard deviation of single saliency map to highlight the importance of dynamic saliency map, as following,

$$\begin{cases} a_1 = 1 \\ a_2 = p_1 \cdot \left(\dfrac{\sigma^{\mathbf{S_S}}}{\sigma^{\mathbf{S_D}}} \right)^{\frac{1}{2}} \\ a_3 = p_2 \cdot \left(\dfrac{\sigma^{\mathbf{S_S}}}{\sigma^{\mathbf{S_{SD}}}} \cdot \dfrac{\sigma^{\mathbf{S_D}}}{\sigma^{\mathbf{S_{SD}}}} \right)^{\frac{1}{2}} \end{cases} \tag{11}$$

where σ is the standard deviation of corresponding saliency map, p_1 and p_2 are constants.

III. EXPERIMENT RESULTS

A. SETTING ON ENCODING

We implement the proposed algorithm into HEVC test model HM-16.0 [20]. Fifteen sequences including CIF (352 × 288), 240P (416 × 240), 480P (832 × 480), 720P (1280 × 720) and 1080P (1920 × 1080) (each with 300 frames) are chosen to evaluate the performance of proposed saliency model. These sequences are from the databases SFU [21] and Xu *et al.* [16]. The performance of the proposed algorithm is tested under lowdelay_main encoder configurations [22]. Quantization parameter (QP) is set as 22 if not explicitly presented in the following. Other encoding parameters are set by default.

B. EVALUATION METRICS

In the experiment, KullbackLeibler divergence (KL), Receiver operating characteristic curve (ROC) and Area under the curve (AUC) values [23] are used to evaluate saliency detection accuracy.

ROC measures the tradeoff between true and false positives at various discrimination thresholds [24], [25]. The area under ROC is called AUC which is the most widely used metric for evaluating saliency detection accuracy. Many methods to calculate AUC have been proposed, in the experiment we use the method proposed by Judd *et al.* [23]. For a given threshold, true positive rate (TP rate) is the ratio of true positives to the total number of fixations, where true positives are saliency map values above threshold at fixated pixels. False positive rate (FP rate) is the ratio of false positives to the total number of saliency map pixels at a given threshold, where false positives are saliency map values above threshold at unfixated pixels [23]. AUC is obtained by plotting the TP rate and the FP rate at various thresholds of saliency map. The larger AUC is, the more accurate the saliency detection model is.

KL is a distribution based metric and can be used to evaluate the loss of information between saliency and fixation maps. As mentioned in [23], the loss information can be evaluated when distribution p (the saliency map) is used to approximate distribution QD (the ground truth fixation map). KL can be calculated by the following,

$$KL\left(P, Q^D\right) = \sum_i Q_i^D \log\left(\varepsilon + \frac{Q_i^D}{\varepsilon + P_i}\right) \tag{12}$$

where ε is a regularization constant. The smaller KL indicates the better approximation of the ground truth.

C. PARAMETERS DETERMINATION

Firstly, we explore the impact of p_0 in (8) on the overall performance. Saliency maps with different p_0 are detected and shown in Fig. 8 where the circled parts in Fig. 8 (a) are fixation areas. In the experiment, p_1 is set as 1, and p_2 is set as 2. Comparing Fig. 8 (b), (c) and (d), it can be found that filtering strength of dynamic ground is in direct proportion to p_0. When $p_0 = 1$, the dynamic background is not fully filtered; when $p_0 = 3$, the dynamic background is over-filtered, i.e., some details are missing. Therefore, $p_0 = 2$ is more reasonable.

To further explore the impact of p_0 on the overall performance, AUC and KL are tested with different p_0, as shown

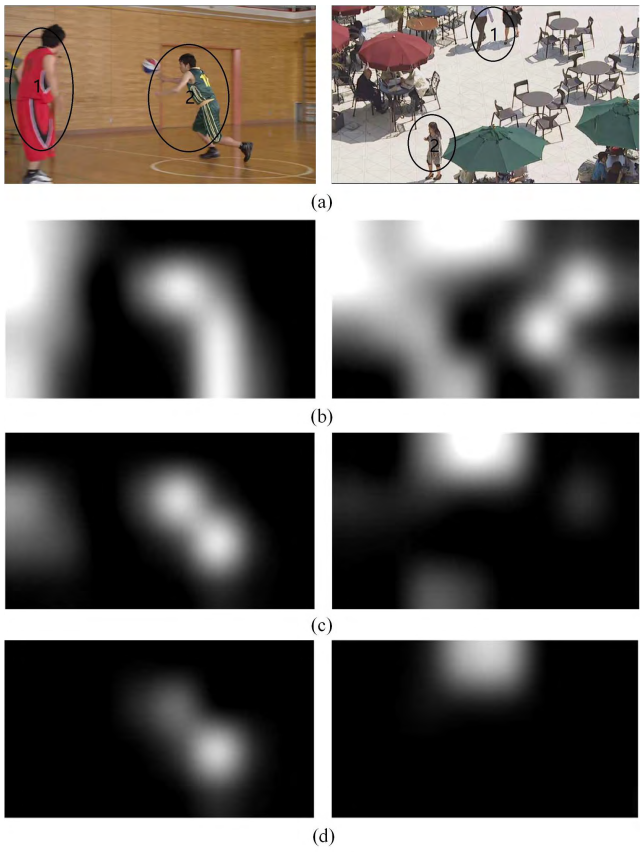


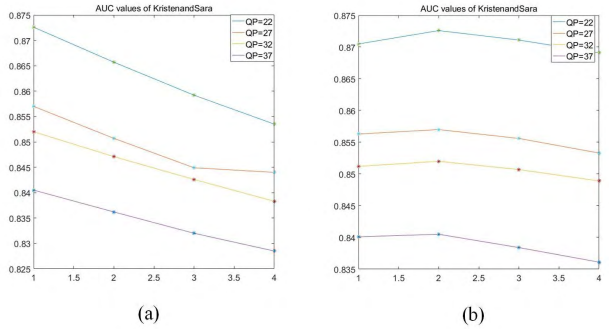
FIGURE 8. Saliency map comparison with different p_0: (a) Original frame (b) Saliency map with $p_0 = 1$ (c) Saliency map with $p_0 = 2$ (d) Saliency map with $p_0 = 3$.

TABLE 1. Comparison of AUC and KL with different p_0 (QP = 22).

Sequences	$p_0 = 1$		$p_0 = 2$		$p_0 = 3$	
	AUC	KL	AUC	KL	AUC	KL
HallMonitor	0.8429	1.5956	0.8820	1.4326	0.8455	1.6102
FOREMAN	0.6691	2.2745	0.8823	1.8814	0.5700	2.4056
HARBOUR	0.5796	1.7419	0.6844	1.6103	0.5304	1.7491
BUS	0.6828	1.9244	0.7242	1.8046	0.5744	2.0564
Flower	0.5097	2.1034	0.5398	2.0923	0.5360	2.1346
BQMall	0.6648	1.9261	0.6946	1.9171	0.6346	1.9791
BQSquare	0.4980	1.5160	0.6596	1.2662	0.4558	1.4227
BasketballPass	0.6581	1.6809	0.7557	1.4753	0.6297	1.5825
Johnny	0.8537	2.9641	0.9186	2.5347	0.8780	2.8291
FourPeople	0.8111	2.7512	0.8253	2.5599	0.7665	2.8483
KristenAndSara	0.7565	3.2866	0.9215	2.4798	0.8029	3.2155
Cactus	0.7234	3.2626	0.7696	3.1450	0.7336	3.2777
Average	0.6875	2.2523	0.7715	2.0166	0.6631	2.2592

in Table 1. In the experiment, p_1 is set as 1, and p_2 is set as 2. From Table 1, AUC gaining is up to 0.1 and KL decreasing is up to 0.2351 on average when $p_0 = 2$. Therefore, we set $p_0 = 2$ in the following experiment.

Then, we explore the impact of p_1 and p_2 in (11) on the overall performance. The result of KristenAndSara is shown in Fig. 9, where Fig. 9 (a) is tested with variable p_1 and constant p_2, and Fig. 9 (b) is tested with constant p_1 and variable p_2. From Fig. 9 (a), the best performance is achieved at point $p_1 = 1$; from Fig. 9 (b), the best performance is



(a) (b)

FIGURE 9. AUC performance of KristenAndSara with different p_1 and p_2: (a) p_1 is variable, $p_2 = 2$ (b) $p_1 = 1$, p_2 is variable.

achieved at point $p_2 = 2$. Therefore, we set $p_1 = 1$ and $p_2 = 2$ in the following experiment.

D. PERFORMANCE OF STATIC SALIENCY DETECTION

To evaluate the performance of the proposed static saliency detection algorithm, saliency maps detected from single static feature map by (4) are detected. Fig. 10 shows the detected single channel static saliency maps and the fused static saliency map by (5). From Fig. 10, four static feature maps play different role in the fused static saliency map. Y-channel out-stands high brightness contrast areas which have the potential to attract more attention, as humans' eyes are sensitive to luminance change. Similarly, Cb-channel and Cr-channel features out-stand high chroma contrast areas. T-channel out-stands the outline of objects in pictures. In general, static features in the proposed algorithm are all low-level features which are inspired by human vision characters.

TABLE 2. AUC performance of single static channel.

Sequences	Y-channel	Cb-channel	Cr-channel	T-channel	fused
HallMonitor	0.5776	0.6555	0.6555	0.6097	0.6690
FOREMAN	0.4502	0.5590	0.5590	0.5546	0.4965
HARBOUR	0.5463	0.4444	0.4444	0.5278	0.4860
BUS	0.4819	0.4834	0.4934	0.7220	0.5878
Flower	0.5167	0.4639	0.4639	0.5053	0.5001
BQMall	0.5760	0.5508	0.5508	0.6414	0.6206
BQSquare	0.5569	0.4050	0.4050	0.5003	0.4520
BasketballPass	0.6052	0.6708	0.6708	0.6346	0.6736
Johnny	0.5559	0.8640	0.8640	0.8482	0.8653
FourPeople	0.4476	0.6184	0.6184	0.6913	0.5943
SlideEditing	0.5766	0.6874	0.6874	0.6028	0.6699
SlideShow	0.3101	0.8078	0.8078	0.7638	0.7790
KristenAndSara	0.5283	0.6098	0.6098	0.7791	0.6999
Cactus	0.5143	0.6224	0.6224	0.7632	0.6508
Average	0.5174	0.6030	0.6038	0.6532	0.6246

The AUC performance of each single channel is shown in Table 2. From Table 2, T-channel outperforms the other three channels in sense of AUC; performance of Cb-channel is similar as that of Cr-channel; performance of Y-channel is a little worse than that of the other three channels. Even so, the ratio between the average AUC of any two channels is approximately to 1. Therefore, the same weights are assigned when they are fused into one static saliency map, as shown in (5).

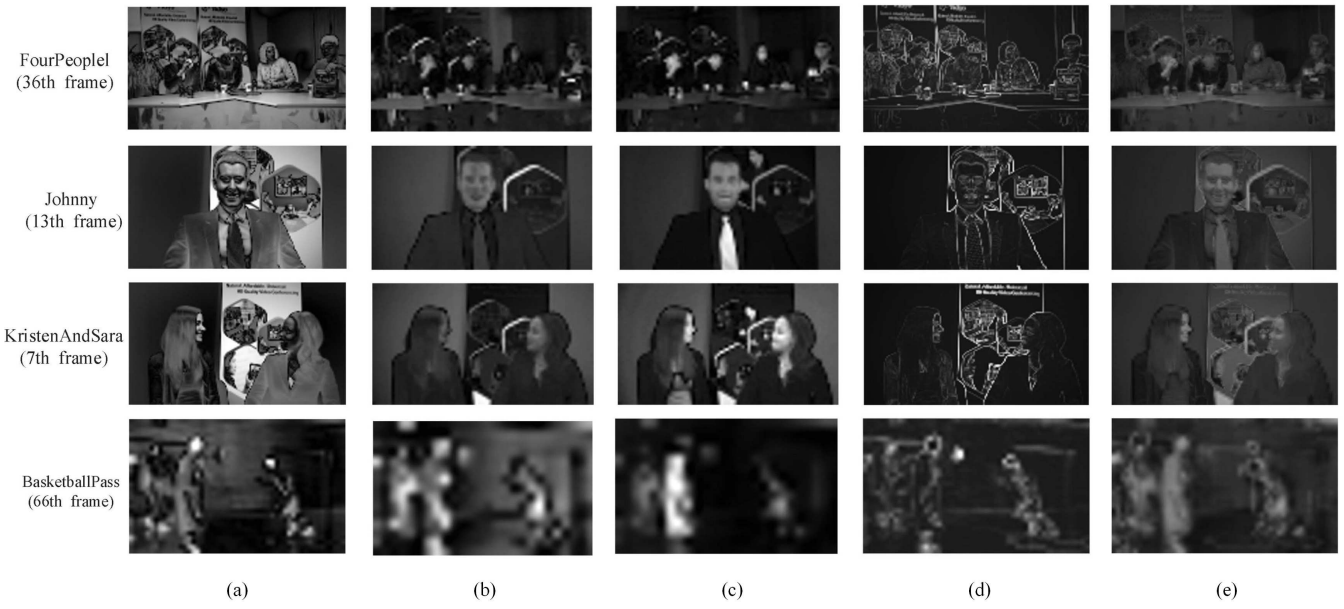


FIGURE 10. Visual comparison between saliency maps of four static channels and the fused static map: (a) Y-channel (b) Cb-channel (c) Cr-channel (d) T-channel (e) fused.

E. PERFORMANCE OF DYNAMIC SALIENCY DETECTION

In the proposed dynamic saliency detection algorithm, a mask is designed to filter out the dynamic background, as shown in equation (7). Here, we test the performance of the designed mask to show whether it can improve the performance of dynamic saliency detection. In the experiment, dynamic saliency map without filtering, dynamic saliency map filtered by CU depth only, dynamic saliency map filtered by bits only and dynamic saliency map filtered by the designed combined mask are detected, respectively. The result of FourPeople is shown in Fig. 11. FourPeople is a sequence without dynamic background. Therefore, the filtered maps by CU depth only, bits only and the combined mask are similar to that without filtering, as shown in Fig. 11. But for sequence BasketballPass, there is a lot of dynamic background in the frame, which introduced pseudo salient areas, as shown in Fig. 12 (a). The CU depth only and bits only masks cannot fully filter out dynamic background, as shown in Fig. 12 (b)-(c). In contrast, the combined mask can filter out almost all dynamic background, as shown in Fig. 12 (d).

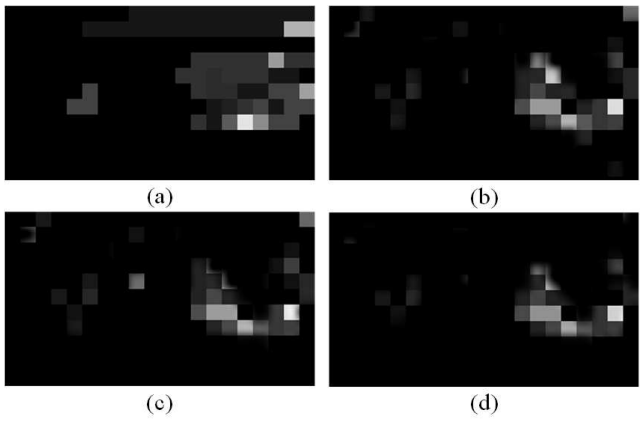


FIGURE 11. Dynamic saliency maps comparison of FourPeople with different filtering methods: (a) without filtering (b) CU depth only (c) bits only (d) combined mask.

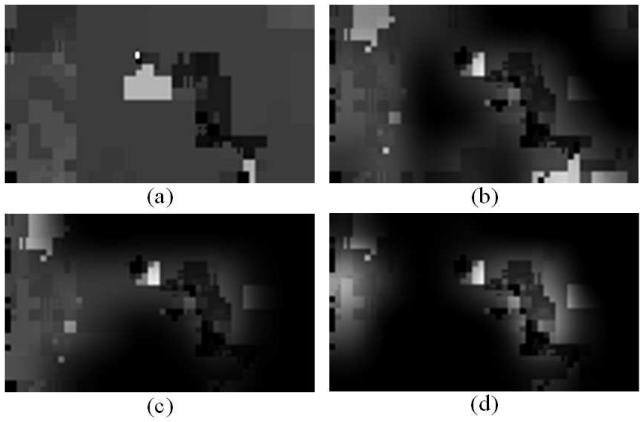


FIGURE 12. Dynamic saliency maps comparison of BasketballPass with different filtering methods: (a) without filtering (b) CU depth only (c) bits only (d) combined mask.

The AUC and KL performances of CU depth only, bits only and combined mask are tested and shown in Table 3. From Table 3, the AUC gaining by the combined mask is up to 0.1036 compared with CU depth only and bits only masks, and the KL decreasing is up to 0.2622. Therefore, the designed combined mask is effective in filtering dynamic background.

FIGURE 13. Comparison of saliency maps for different models: (a) Human fixations (b) proposed (c) SUN (d) Bayes (e) Seo (f) Hou (g) Itti.

TABLE 3. Performance comparison of different masks.

Sequences	CU Depth Only		Bits Only		Combined	
	AUC	KL	AUC	KL	AUC	KL
HallMonitor	0.8250	1.6581	0.8377	1.6098	0.8820	1.4326
FOREMAN	0.6110	2.2308	0.6640	2.2849	0.8823	1.8814
HARBOUR	0.5437	1.7705	0.6177	1.7022	0.6844	1.6103
BUS	0.6160	2.0541	0.6685	1.9436	0.7242	1.8046
Flower	0.5006	2.1188	0.5153	2.1039	0.5398	2.0923
BQMall	0.6489	1.9630	0.6800	1.9000	0.6946	1.9171
BQSquare	0.4987	1.4561	0.4871	1.5381	0.6596	1.2662
BasketballPass	0.5821	1.7429	0.6548	1.6868	0.7557	1.4753
Johnny	0.8341	3.0769	0.8522	2.9676	0.9186	2.5347
FourPeople	0.8076	2.7645	0.8233	2.7287	0.8253	2.5599
KristenAndSara	0.7390	3.3118	0.7586	3.2879	0.9215	2.4798
Cactus	0.7078	3.2883	0.7293	3.2396	0.7696	3.1450
Average	0.6595	2.2863	0.6907	2.2494	0.7715	2.0166

F. OVERALL PERFORMANCE ANALYSIS

To evaluate the performance of the proposed algorithm, saliency maps detected by the proposed algorithm, SUN [27], Bayes [9], Seo [28], Hou [29] and Itti [7] are presented in Fig. 13 (b)-(g) respective. As a comparison, the human fixations are also presented, as shown in Fig. 13 (a). Comparing these saliency maps with the human fixation data in Fig. 13, almost all the human fixation areas can be effectively marked by each of these six saliency detection algorithms. However, there also exist many pseudo salient areas in maps detected by SUN [27], Seo [28], Hou [29] and Itti [7]. Comparing saliency maps by the proposed algorithm and Bayes [9], salient areas detected by the proposed algorithm are more accurate, as shown in Fig. 13 (a), (b) and (d). Therefore, the proposed saliency detection algorithm is visually better than the other five classic algorithms.

Then, the ROC curves are plotted to further evaluate the performance, as shown in Fig. 14. From Fig. 14, the proposed algorithm generally has higher true positive rates than others at the same false positive rates, which indicates relatively better performance than the other five classic algorithms.

Table 4 and Table 5 show the detail AUC and KL performance of six algorithms. From Table 4, the proposed

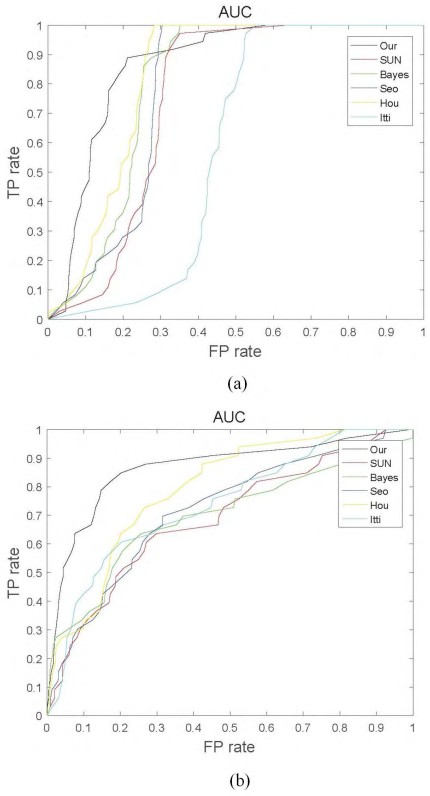


FIGURE 14. ROC curves comparison: (a) Johnny (b) BasketballPass.

algorithm achieves better AUC performance than the other five algorithms for more than half the sequences. The AUC gaining by the proposed algorithm is up to 0.1223 on average compared with the other five algorithms. From Table 5, the proposed algorithm achieves better KL performance than the other five algorithms for more than half the sequences. The KL decreasing by the proposed algorithm is up to 0.8362 on average compared with the other five algorithms. Therefore, the proposed saliency detection algorithm

TABLE 4. AUC performance comparison.

Sequences	SUN	Bayes	Seo	Hou	Itti	Proposed
HallMonit	0.7071	0.7999	0.8393	0.7912	0.7839	0.8820
FOREMAN	0.5285	0.6905	0.5696	0.6543	0.5095	0.8823
HARBOUR	0.5236	0.5773	0.4273	0.4832	0.4844	0.6844
BUS	0.7309	0.7202	0.7126	0.7462	0.6895	0.7242
Flower	0.4959	0.5166	0.5531	0.5262	0.6426	0.5398
BQMall	0.7129	0.7317	0.6652	0.7032	0.7270	0.6946
BQSquare	0.4693	0.5291	0.4935	0.5423	0.6600	0.6596
BasketballPass	0.6452	0.7905	0.6748	0.7252	0.7169	0.7557
BasketballDrill	0.5815	0.7025	0.6281	0.6390	0.7141	0.7209
Johnny	0.7532	0.8813	0.7915	0.8594	0.6115	0.9186
FourPeople	0.7350	0.6758	0.7323	0.8095	0.6928	0.8253
SlideEditing	0.5954	0.8559	0.6491	0.6956	0.6802	0.8428
SlideShow	0.7880	0.7892	0.7296	0.7284	0.6332	0.8142
KristenAndSara	0.8193	0.8163	0.8370	0.8643	0.8359	0.9215
Cactus	0.7158	0.7584	0.7256	0.7566	0.6414	0.7696
Average	0.6534	0.7223	0.6686	0.7016	0.6682	0.7757

TABLE 5. KL performance comparison.

Sequences	SUN	Bayes	Seo	Hou	Itti	Proposed
HallMonit	1.8691	1.6449	1.4449	1.7159	1.5896	1.4326
FOREMAN	2.3997	2.3633	3.3546	2.3962	2.5121	1.8814
HARBOUR	1.6730	1.9442	3.0244	1.8395	1.9248	1.6103
BUS	1.7545	3.7337	2.2277	1.8240	1.9191	1.8046
Flower	1.9633	5.3404	2.7645	1.9258	1.6062	2.0923
BQMall	1.9227	3.0857	2.0036	1.8600	1.7726	1.9171
BQSquare	1.6783	2.1565	2.7102	1.4812	1.2892	1.2662
BasketballPass	1.5692	1.9868	3.2289	1.4113	1.5118	1.4753
BasketballDrill	2.1652	2.1419	2.8392	2.0602	1.9995	1.9363
Johnny	3.4402	3.0371	3.0761	3.0933	3.6574	2.5347
FourPeople	2.9940	3.2652	2.8867	2.8302	3.1014	2.5599
SlideEditing	3.3471	2.4108	3.5077	2.8534	2.9414	2.6438
SlideShow	5.9667	3.8064	4.8856	2.6344	3.0069	2.4438
KristenAndSara	3.0774	3.0991	2.7087	2.9077	3.0863	2.4798
Cactus	3.2634	3.3286	3.1020	3.1354	3.4071	3.1450
Average	2.6056	2.8896	2.9177	2.2646	2.3550	2.0815

is superior to the other five classic algorithms in sense of AUC and KL.

TABLE 6. Saliency detection time per frame.

	SUN	Bayes	Seo	Hou	Itti	Proposed
Time (s)	1.6	18.7	40.6	1.8	0.12	1.9

Next, computation complexity of these six algorithms are tested. Computation complexity here is evaluated by saliency detection time per frame, as shown in Table 6. Saliency detection by Itti [7] can run at an amazing speed which is only 0.12s per frame. The complexity of SUN [27], Hou [29] and the proposed algorithm have the similar complexity which are 1.6s, 1,8s and 1.9s per frame respectively.

All the above results of the proposed algorithm are tested with QP $=$ 22. However, MV, CU depth and bits used in the proposed algorithm are associated with QP. To explore how QP can impact the performance, AUC and KL with four QPs are tested, as shown in Table 7 and Table 8. According to Table 7, the AUC difference with four QPs is very small. The maximum AUC difference among all sequences is less than 0.03 and the maximum average difference is only 0.0063. Similarly, the maximum KL difference among all sequences is less than 0.1, and the maximum average difference is only 0.0116. Therefore, QP has a small effect on the

TABLE 7. AUC performance with multiple QPs.

Sequences	QP=22	QP=27	QP=32	QP=37
HallMonit	0.8820	0.8858	0.8812	0.8771
FOREMAN	0.8823	0.8620	0.8471	0.8629
HARBOUR	0.6844	0.6935	0.6902	0.6911
BUS	0.7242	0.7413	0.7423	0.7270
Flower	0.5398	0.5348	0.5335	0.5321
BQMall	0.6946	0.6936	0.6925	0.6984
BQSquare	0.6596	0.6556	0.6550	0.6534
BasketballPass	0.7554	0.7523	0.7494	0.7541
FourPeople	0.8253	0.8178	0.8071	0.8047
Johnny	0.9186	0.9208	0.9212	0.9215
SlideEditing	0.8428	0.8423	0.8414	0.8422
SlideShow	0.8412	0.8283	0.8268	0.8374
KristenAndSara	0.9215	0.9204	0.9253	0.9286
Cactus	0.7696	0.7487	0.7396	0.7401
Average	0.7815	0.7784	0.7752	0.7765

TABLE 8. KL performance with multiple QPs.

Sequences	QP=22	QP=27	QP=32	QP=37
HallMonit	1.4326	1.4141	1.4145	1.4370
FOREMAN	1.8814	1.9018	1.9201	1.8468
HARBOUR	1.6103	1.5997	1.6038	1.6119
BUS	1.8046	1.7903	1.7802	1.8064
Flower	2.0923	2.0863	2.1053	2.0885
BQMall	1.9171	1.9116	1.9121	1.9159
BQSquare	1.2662	1.2638	1.2603	1.2656
BasketballPass	1.4753	1.4963	1.4979	1.4789
FourPeople	2.5599	2.6362	2.6797	2.6864
Johnny	2.5347	2.5368	2.5467	2.5577
SlideEditing	2.6438	2.6038	2.5776	2.6392
SlideShow	2.4438	2.4095	2.4149	2.3877
KristenAndSara	2.4798	2.4723	2.4634	2.4644
Cactus	3.1450	3.1937	3.1940	3.2330
Average	2.0919	2.0940	2.0979	2.1014

performance of the proposed algorithm, although the quality of compressed video is mainly related with QP. It is due to the difference between salient areas and non-salient areas which is independent of QP. As described in Section II, salient areas are usually more textured and contains more complex movement, therefore, salient areas tend to consume more bits and have larger MVs and more complex CU structure compared with non-salient areas regardless of the value of QP. As bits, MV and CU structure play the main role in the proposed algorithm, the proposed algorithm performs well with various QPs.

IV. CONCLUSION

In this paper, we propose a saliency detection algorithm for HEVC compressed videos. The proposed algorithm extracts features from compressed videos so that it is not necessary to decode the video. The proposed algorithm is composed of static saliency detection and dynamic saliency detection. Static features include three color-channel feature maps and one texture feature map. Gaussian kernel is used to extract data structure in static saliency map. Dynamic feature is composed of motion vector. A CU depth and bits combined mask

is designed to filter dynamic background. Finally, experiments show the advancement of the proposed algorithm.

REFERENCES

[1] E. Matin, "Saccadic suppression: A review and an analysis," *Psychol. Bull.*, vol. 81, no. 12, pp. 899–917, 1974.

[2] N. J. Butko and J. R. Movellan, "Optimal scanning for faster object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2751–2758.

[3] D. Gao, S. Han, and N. Vasconcelos, "Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 989–1005, Jun. 2009.

[4] M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir, "A comparative study of image retargeting," *ACM Trans. Graph.*, vol. 29, no. 5, pp. 160:1–160:10, 2010.

[5] Z. Li, S. Qin, and L. Itti, "Visual attention guided bit allocation in video compression," *Image Vis. Comput.*, vol. 29, no. 1, pp. 1–14, Jan. 2011.

[6] C. Koch and S. Ullman, *Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry*. Dordrecht, The Netherlands: Springer, 1987, pp. 115–141.

[7] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[8] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 545–552.

[9] L. Itti and P. Baldi, "Bayesian surprise attracts human attention," *Vis. Res.*, vol. 49, no. 10, pp. 1295–1306, Jun. 2009.

[10] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[11] L. Marchesotti, C. Cifarelli, and G. Csurka, "A framework for visual saliency detection with applications to image thumbnailing," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 2232–2239.

[12] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 185–198, Jan. 2010.

[13] Y. Zhai and M. Shah, "Visual attention detection in video sequences using spatiotemporal cues," in *Proc. 14th ACM Int. Conf. Multimedia*, 2006, pp. 815–824.

[14] Y. Fang, Z. Chen, W. Lin, and C.-W. Lin, "Saliency-based image retargeting in the compressed domain," in *Proc. 19th ACM Int. Conf. Multimedia*, 2011, pp. 1049–1052.

[15] Y. Fang, W. Lin, Z. Chen, C.-M. Tsai, and C.-W. Lin, "A video saliency detection model in compressed domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 27–38, Jan. 2014.

[16] M. Xu, L. Jiang, X. Sun, Z. Ye, and Z. Wang, "Learning to detect video saliency with HEVC features," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 369–385, Jan. 2017.

[17] Y.-L. Huang and R.-F. Chang, "Texture features for DCT-coded image retrieval and classification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 6, Mar. 1999, pp. 3013–3016.

[18] Y. Li, Y. Zhou, J. Yan, Z. Niu, and J. Yang, "Visual saliency based on conditional entropy," in *Computer Vision-ACCV*. Berlin, Germany: Springer, 2010, pp. 246–257.

[19] A. C. L. Ngo, G. Qiu, G. Underwood, L. Ang, and K. P. Seng, "Visual saliency based on fast nonparametric multidimensional entropy estimation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 1305–1308.

[20] K. McCann, C. Rosewarne, B. Bross, M. Naccari, K. Sharman, and G. Sullivan, *High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description*, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-R1002, Jul. 2014.

[21] H. Hadizadeh, M. J. Enriquez, and I. V. Bajic, "Eye-tracking database for a set of standard video sequences," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 898–903, Feb. 2012.

[22] K. Sharman, *Common Test Conditions*, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-Z1100, Jan. 2017.

[23] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/8315047

[24] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[25] D. M. Green and J. A. Swets, *Signal Detection Theory and Psychophysics*. Melbourne, FL, USA: Krieger, 1966.

[26] C. Guo, Q. Ma, and L. Zhang, "Spatio-temporal saliency detection using phase spectrum of quaternion Fourier transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Anchorage, AK, USA, Jun. 2008, pp. 1–8.

[27] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "SUN: A Bayesian framework for saliency using natural statistics," *J. Vis.*, vol. 8, no. 7, p. 32, Dec. 2008.

[28] H. J. Seo and P. Milanfar, "Static and space-time visual saliency detection by self-resemblance," *J. Vis.*, vol. 9, no. 12, p. 15, 2009.

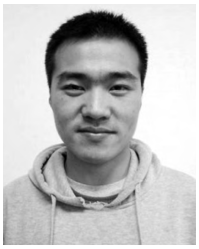
[29] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 194–201, Jan. 2012.

[30] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2106–2113.

[31] Z. Bylinskii *et al. MIT Saliency Benchmark*. [Online]. Available: http://saliency.mit.edu/

WEI ZHOU received the B.E., M.S., and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China, in 2001, 2004, and 2007, respectively. He is currently a Professor with Northwestern Polytechnical University. His research interests include video coding and associated very large-scale integration architecture design.

RUI BAI received the B.E. degree from Northwestern Polytechnical University, Xi'an, China, in 2017, where he is currently pursuing the M.S. degree. His research interests include video coding and computer vision.



HENGLU WEI received the B.E. and M.S. degrees from Northwestern Polytechnical University, Xi'an, China, in 2013 and 2015, respectively, where he is currently pursuing the Ph.D. degree. His research interests include video coding and image processing.