Learning to Detect Video Saliency With HEVC Features

Mai Xu, *Member, IEEE*, Lai Jiang, Xiaoyan Sun, *Senior Member, IEEE*, Zhaoting Ye, and Zulin Wang, *Member, IEEE*

***Abstract*— Saliency detection has been widely studied to predict human fixations, with various applications in computer vision and image processing. For saliency detection, we argue in this paper that the state-of-the-art High Efficiency Video Coding (HEVC) standard can be used to generate the useful features in compressed domain. Therefore, this paper proposes to learn the video saliency model, with regard to HEVC features. First, we establish an eye tracking database for video saliency detection, which can be downloaded from https://github.com/remega/video_database. Through the statistical analysis on our eye tracking database, we find out that human fixations tend to fall into the regions with large-valued HEVC features on splitting depth, bit allocation, and motion vector (MV). In addition, three observations are obtained with the further analysis on our eye tracking database. Accordingly, several features in HEVC domain are proposed on the basis of splitting depth, bit allocation, and MV. Next, a kind of support vector machine is learned to integrate those HEVC features together, for video saliency detection. Since almost all video data are stored in the compressed form, our method is able to avoid both the computational cost on decoding and the storage cost on raw data. More importantly, experimental results show that the proposed method is superior to other state-of-the-art saliency detection methods, either in compressed or uncompressed domain.**

***Index Terms*— Saliency detection, compressed domain, HEVC, machine learning, SVM.**

I. Introduction

According to the study on the human visual system (HVS) [1], when a person looks at a scene, he/she may pay much visual attention on a small region (the fovea) around a point of eye fixation at high resolutions. The other regions, namely the peripheral regions, are captured with little attention at low resolutions. As such, humans are able to avoid the processing of tremendous visual data. Visual attention is therefore a key to perceive the world around humans, and it has been extensively studied in psychophysics, neurophysiology, and even computer vision societies [2]. Saliency detection is an effective way to predict the amount of human visual attention attracted by different regions in images/videos, with computation on their features. Most recently, saliency detection has been widely applied in object detection [3], [4], object recognition [5], image retargeting [6], image quality assessment [7], and image/video compression [8], [9].

In earlier time, some heuristic saliency detection methods are developed according to the understanding of the HVS. Specifically, in light of the HVS, Itti et al. [10] found out that the low level features of intensity, color, and orientation are efficient in detecting saliency of still images. In their method, center-surround responses in those feature channels are established to yield the conspicuity maps. Then, the final saliency map can be obtained by linearly integrating conspicuity maps of all three features. For detecting saliency in videos, Itti *et al.* [11] proposed to add two dynamic features (i.e., motion and flicker contrast) into Itti's image saliency model [10]. Later, other advanced heuristic methods [12]–[18] have been proposed for modeling video saliency.

Recently, data-driven methods [19]–[24] have emerged to learn the visual attention models from the ground-truth eye tracking data. Specifically, Judd *et al.* [19] proposed to learn a linear classifier of support vector machine (SVM) from training data for image saliency detection, based on several low, middle, and high level features. For video saliency detection, most recently, Rudoy *et al.* [23] have proposed a novel method to predict saliency by learning the conditional saliency map from human fixations over a few consecutive video frames. This way, the inter-frame correlation of visual attention is taken into account, such that the accuracy of video saliency detection can be significantly improved. Rather than free-view saliency detection, a probabilistic multi-task learning method was developed in [21] for the task-driven video saliency detection, in which the "stimulus-saliency" functions are learned from the eye tracking data as the top-down attention models.

High efficiency video coding (HEVC) [25] was formally approved as the state-of-the-art video coding standard in April 2013. It achieves double coding efficiency improvement over the preceding H.264/AVC standard. Interestingly, we found out that the state-of-the-art HEVC encoder can be explored as a feature extractor to efficiently predict video saliency. As shown in Figure 1, the HEVC domain features on splitting depth, bit allocation and motion vector (MV)

Manuscript received February 1, 2016; revised August 5, 2016; accepted November 5, 2016. Date of publication November 14, 2016; date of current version November 29, 2016. This work was supported in part by the China 973 Program under Grant 2013CB329006, in part by the National Nature Science Foundation of China projects under Grant 61573037 and Grant 61202139, in part by the Fok Ying Tung Education Foundation under Grant 151061, and in part by the MSRA Visiting Young Faculty Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Catarina Brites.

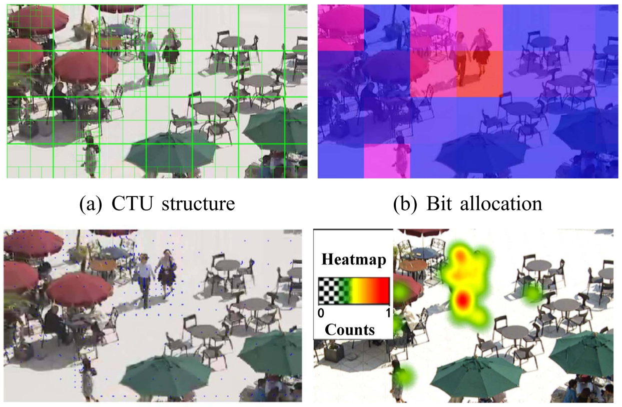
M. Xu, L. Jiang, and Z. Ye are with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China (e-mail: Maixu@buaa.edu.cn; jianglai.china@aliyun.com; ZhaotingYe@outlook.com).

X. Sun is with Microsoft Research Asia, Beijing 100080, China (e-mail: xysun@microsoft.com).

Z. Wang is with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China, and also with the Collaborative Innovation Center of Geospatial Technology, Wuhan 430079, China (e-mail: wzulin@buaa.edu.cn).

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

Digital Object Identifier 10.1109/TIP.2016.2628583



(a) CTU structure (b) Bit allocation

(c) MV (d) Heat map of human fixations

Fig. 1. An example of HEVC domain features and heat map of human fixations for one video frame. (a), (b), and (c) are extracted from the HEVC bitstream of video *BQSquare* (resolution: 416 × 240) at 130 Kbps. Note that in (c) only the MVs that are larger than 1 pixel are shown. (d) is the heat map convolved with a 2D Gaussian filter over fixations of 32 subjects.

for each coding tree unit (CTU), are highly correlated with the human fixations. The statistical analysis of Section III-B verifies such high correlation. Therefore, we develop several features in our method for video saliency detection, which are based on splitting depths, bit allocation and MVs in HEVC domain. It is worth pointing out that most videos exist in the form of encoded bitstreams, and the features related to entropy and motion have been well exploited by video coding at the encoder side. Since [2] has argued that entropy and motion are very effective in video saliency detection, our method utilizes these well-exploited HEVC features (splitting depth, bit allocation and MV) at the decoder side to achieve high accurate detection on video saliency.

Generally speaking, the main motivation of using HEVC features in our saliency detection method is two fold. (1) Our method takes advantage from sophisticated encoding of HEVC, to effectively extract features for video saliency detection. Our experimental results in this paper also show that the HEVC features are indeed very effective in video saliency detection. (2) Our method can efficiently detect video saliency from HEVC bitstreams without completely decoded the videos, thus avoiding both the computational time and storage. Consequently, our method is generally more efficient than the aforementioned video saliency detection methods at pixel domain (or called uncompressed domain), which have to decode the bitstreams into raw data. Such efficiency is also validated by our experiments.

There are only a few methods [26]–[28] proposed for detecting video saliency in compressed domain of previous video coding standards. Among these methods, the block-wise discrete cosine transform (DCT) coefficients and MVs are extracted in MPEG-2 [26] and MPEG-4 [27]. Bit allocation of H.264/AVC is exploited for saliency prediction in [28]. However, all above methods do not take full advantage of the sophisticated features of the modern HEVC encoder, such as CTU splitting [29] and r-λ bit allocation [30]. More importantly, all methods of [26]–[28] fail to find out the precise impact of each compressed domain feature on attracting visual attention. In fact, the relationship between compressed domain features and visual attention can be learned from the ground-truth eye tracking data. Thereby, this paper proposes to learn the visual attention model of videos with regard to the well explored HEVC features.

Similar in spirit, the latest work of [31] also makes use of HEVC features for saliency detection. Despite conceptually similar, our method is greatly different from [31] in two aspects. From the aspect of feature extraction, our method develops pixel-wise HEVC features, whilst [31] directly uses block-based HEVC features with deeper decoding (e.g., inverse DCT). Instead of going deeper, our method develops shallow decoded HEVC features with sophisticated design of temporal and spatial difference on these features, more unrestrictive than [31]. In addition, camera motion is detected and then removed in our HEVC features, such that our features are more effective in predicting attention. From the aspect of feature integration, compared with [31], our method is data-driven, in which a learning algorithm is developed to bridge the gap between HEVC features and video saliency. Meanwhile, our data-driven method benefits from our established eye tracking database with thorough analysis.

Specifically, the main contributions of this paper are listed in the following.

- We establish an eye tracking database on viewing 33 raw videos of the latest data sets, with the thorough analysis and observations on our database.
- We propose several saliency detection features in HEVC domain, according to the analysis and observations on our established eye tracking database.
- We develop a data-driven method for video saliency detection, with respect to the proposed HEVC features.

The rest of this paper is organized as follows. In Section II, we briefly review the related work on video saliency detection. In Section III, we present our eye tracking database as well as the analysis and observations on our database. In light of such analysis and observations, Section IV proposes several HEVC features for video saliency detection. Section V outlines our learning based method, which is based on the proposed HEVC features. Section VI shows the experimental results to validate our method. Finally, Section VII concludes this paper.

II. Related Work on Video Saliency Detection

A. Heuristic Video Saliency Detection

For modeling saliency of a video, a great number of methods [11]–[18] have been proposed. Itti *et al.* [11] started the initial work of video saliency detection, by adding two dynamic features of motion and flicker contrast into Itti's image saliency model [10]. Later, a novel term called *surprise* was defined in [14] to measure how the visual change attracts human observers. With the new term *surprise*, [14] developed a Bayesian framework to calculate the Kullback-Leibler divergence (KL) between spatio-temporal posterior and prior beliefs, for predicting video saliency. Some other Bayesian framework related methods, e.g., [15], were also proposed for video saliency detection. Most recently, some advanced video saliency detection methods [16]–[18] have been proposed.

To be more specific, Guo and Zhang [16] applied phase spectrum of quaternion Fourier transform (PQFT) on four feature channels (two color channels, one intensity channel, and one motion channel) to detect video saliency. Lin *et al.* [18] utilized earth mover's distance (EMD) to measure the center-surround difference in spatio-temporal receptive filed, for producing the dynamic saliency maps of videos. Inspired by sparse representation, Ren *et al.* [17] proposed to explore the movement of a target patch for temporal saliency detection of videos. In their method, the movement of the target patch can be estimated by finding the minimal reconstruction error of sparse representation regarding the patches of neighboring frames. In addition to temporal saliency detection, the center-surround contrast needs be modeled for spatial saliency detection. This is achieved through sparse representation with respect to neighboring patches.

In fact, top-down visual cues play an important role in determining the saliency of a scene. Thereby, the top-down visual attention models have been studied in [32] and [33] for predicting the saliency of dynamic scenes in a video. In [32], Pang *et al.* proposed to integrate the top-down information of eye movement patterns (i.e., passive and active states [13]) in video saliency detection. In [33], Wu and Xu found out that the high level features, such as face, person, car, speaker, and flash, may attract extensive human attention. Thus, these high level features are integrated with the bottom-up model [16] for saliency detection of news videos.

However, the understanding of the HVS is still in its infancy, and saliency detection thus has a long way to go yet. In fact, we may rethink saliency detection by taking advantage of the existing video coding techniques. Specifically, the video coding standards have evolved for almost three decades, with HEVC being the latest one. The evolution of video coding adopts several elegant and effective techniques to produce several sophisticated features, for continuously improving coding efficiency. For example, the state-of-the-art HEVC standard introduced fractional sample interpolation to represent MVs with quarter-sample precision, thus being able to precisely model object motions. Moreover, HEVC proposes to partition CTUs into smaller blocks using the tree structure and quadtree-like signaling [29], which can well reflect the texture complexity of video frames. On the other hand, the HEVC features, which are generated by the sophisticated process of the latest HEVC techniques, may be explored for efficient video saliency detection.

B. Data-Driven Video Saliency Detection

During the past decade, data-driven methods have emerged as a possible way to learn video saliency model from ground-truth eye tracking data, instead of the study on the HVS. The existing data-driven video saliency detection can be further divided into task-driven [13], [21], [22], [34], [35] and free-view [20], [23], [24], [36] methods.

For task-driven video saliency detection, Peter and Itti [13] proposed to incorporate the computation on signatures of each video frame. Then, a regression classifier is learned from the subjects' fixations on playing video games, which associates the different classes of signatures (seen as gist) with the gaze patterns of task-driven attention. Combined with 12 multi-scale bottom-up features, [13] has high accuracy in task-driven saliency detection. Most recently, a dynamic Bayesian network method [35] has been proposed for learning top-down visual attention model of playing video games. Besides the task of playing video games, a data-driven method [34] on video saliency detection was proposed with the dynamic consistency and alignment models, for the task of action recognition. In [34], the proposed models are learned from the task-driven human fixations on large-scale dynamic computer vision databases like Hollywood-2 [37] and UCF Sports [38]. In [21], Li *et al.* developed a probabilistic multi-task learning method to include the task-related attention models for video saliency detection. The "stimulus-saliency" functions are learned from the eye tracking database, as the top-down attention models to some typical tasks of visual search. As a result, [21] is "good at" video saliency detection in multiple tasks, more generic than other methods that focus on single visual task. However, all task-driven saliency detection methods can only deal with the specific tasks.

For free-view video saliency detection, Kienzle *et al.* [20] proposed a nonparametric bottom-up method to model video saliency, via learning the center-surround texture patches and temporal filters from the eye tracking data. Recently, Lee *et al.* [24] have proposed to extract the spatio-temporal features, i.e., rarity, compactness, center prior, and motion, for the bottom-up video saliency detection. In their bottom-up method, all extracted features are combined together by an SVM, which is learned from the training eye tracking data. In addition to the bottom-up model, Hua *et al.* [36] proposed to learn the middle level features, i.e., gists of a scene, as the top-down cue for both video and image saliency detection. Most recently, Rudoy *et al.* [23] have proposed to detect the saliency of a video, by simulating the way that humans watch the video. Specifically, a visual attention model is learned to predict the saliency map of a video frame, given the fixation maps from the previous frames. As such, the inter-frame dynamics of gaze transitions can be taken into account during video saliency detection.

As aforementioned, this paper mainly concentrates on utilizing the HEVC features for video saliency detection. However, there is a gap between HEVC features and human visual attention. From data-driven perspective, machine learning can be utilized in our method to investigate the relationship between HEVC features and visual attention, according to eye tracking data. Thus, this paper aims at learning an SVM classifier to predict saliency of videos using the features from HEVC domain.

III. Database and Analysis

A. Database of Eye Tracking on Raw Videos

In this paper, we conducted the eye tracking experiment to obtain fixations on viewing videos of the latest test sets. Here, all 33 raw videos from the test sets [9], [39], which have been commonly utilized for evaluating HEVC performance, were included in our eye tracking experiment. We further conducted the extra experiment to obtain the eye tracking data on watching all videos of our database compressed by HEVC at different quality. Through the data analysis, we found that visual attention is almost unchanged when videos are compressed at high or medium quality (more than 30 dB). This is consistent with the result of [40]. Compared with the conventional databases (e.g., SFU [41] and DIEM [42]), the utilization of these videos benefits from the state-of-the-art test sets in providing videos with diverse resolutions and content. For the resolution, the videos vary from 1080p (1920 × 1080) to 240p (416 × 240). For the content, the videos include sport events, surveillance, video conferencing, video games, videos with the subscript, etc.

In our eye tracking experiment, all videos are with YUV 4:2:0 sampling. Here, the resolutions of the videos in Class A of [39] were down-sampled to be 1280 × 800, as the screen resolution of the eye tracker can only reach to 1920 × 1080. Other videos were displayed in their original resolutions. In our experiment, the videos were displayed in a random manner at their default frame rates, to reduce the influence of video playing order on the eye tracking results. Besides, a blank period of 5 seconds was inserted between two consecutive videos, so that the subjects can have a proper rest time to avoid eye fatigue.

There were a total of 32 subjects (18 male and 14 female, aging from 19 to 60) involved in our eye tracking experiment. These subjects were selected from the campuses of Beihang University and Microsoft Research Asia. All subjects have either corrected or uncorrected normal eyesight. Note that only two subjects were experts, who are working in the research field of saliency detection. The other 30 subjects did not have any research background in video saliency detection, and they were also native to the purpose of our eye tracking experiment.

The eye fixations of all 32 subjects over each video frame were recorded by a Tobii TX300 eye tracker at a sample rate of 300 Hz. The eye tracker is integrated with a monitor of 23-inch LCD screen, and the resolution of the monitor was set to be 1920 × 1080. All subjects were seated on an adjustable chair at a distance of around 60 cm from the screen of the eye tracker, ensuring that their horizontal sight is in the center of the screen. Before the experiment, subjects were instructed to perform the 9-point calibration for the eye tracker. Then, all subjects were asked to free-view each video. After the experiment, 392,163 fixations over 13,020 frames of 33 videos were collected. Here, the eye fixations of all subjects and the corresponding Matlab code for our eye tracking database are available online: https://github.com/remega/video_database.

B. Analysis on Our Eye Tracking Database

Figure 1 has shown that the HEVC features, i.e., splitting depth, bit allocation, and MV, are effective in predicting human visual attention. It is therefore interesting to statistically analyze the correlation between these HEVC features and visual attention. From now on, we concentrate on the statistical analysis on our eye tracking database, to show the effectiveness of the HEVC features on the prediction of visual attention. This is a new finding, which reveals the correlation between HEVC features and visual attention.

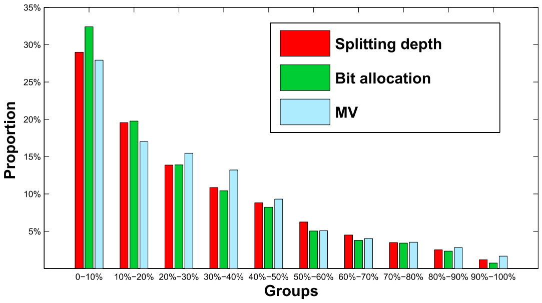


Fig. 2. The statistical results for fixations belonging to different groups of pixels, in which values of the corresponding HEVC features are sorted in the descending order. Here, all 392,163 fixations of 33 videos are used for the analysis. In this figure, the horizontal axis indicates the groups of pixels, in which the values of the corresponding HEVC features are in the descending order. For example, $0-10\%$ means that the first group of pixels, the features of which rank top 10%. The vertical axis shows the percentage of fixations that fall into each group.

For all videos of our database, the features on splitting depth, bit allocation, and MV were extracted from the corresponding HEVC bitstreams. Then, the maps of these features were generated for each video frame. Note that the configuration to generate the HEVC bitstreams can be found in Section VI. Afterwards, a 2D Gaussian filter was applied to all three feature maps of each video frame. For each feature map, after sorting pixels in the descending order of their feature values, the pixels were equally divided into 10 groups according to the values of corresponding features. For example, the group of $0-10\%$ stands for the set of pixels, the features of which rank top 10%. Finally, the number of fixations belonging to each group was counted upon all 33 videos in our database.

We show in Figure 2 the percentages of eye fixations belonging to each group, in which the values of the corresponding HEVC features decrease alongside the groups. From this figure, we can find out that extensive attention is drawn by the regions with large-valued HEVC features, especially for the feature of bit allocation. For example, about 33% fixations fall into the regions of top 10% high-valued feature of bit allocation, whereas the percentage of those hitting the bottom 10% is much less than 2%. Hence, the HEVC features on splitting depth, bit allocation, and MV, are explored for video saliency detection in our method (Section IV).

C. Observations From Our Eye Tracking Database

Beyond the analysis of our eye tracking database, we verify some other factors on attracting human attention, with the following three observations. These observations provide insightful guide for developing our saliency detection method.

Observation 1: Human fixations lag behind the moving or new objects in a video by some microseconds.

In Figure 3, we show the frames of videos *BasketballDrive* and *Kimono* with the corresponding heat maps of human fixations. The first row of this figure reveals that the visual attention falls behind the moving object, as the fixations trail the moving basketball. In particular, the distance between the

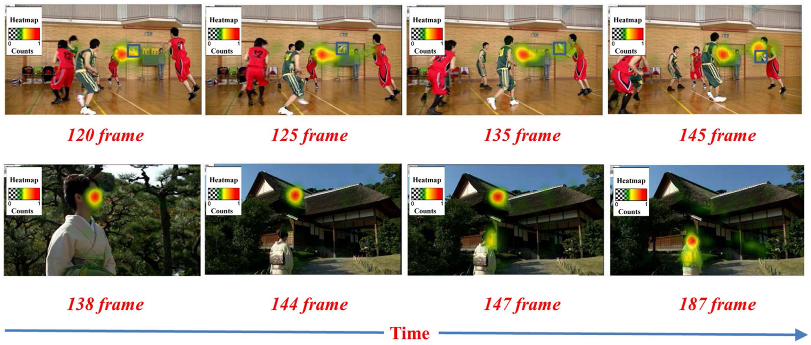


Fig. 3. Illustration of Observation 1. This figure shows the heat maps of human fixations of all 32 subjects, on several selected frames of videos *BasketballDrive* and *Kimono*. In *BasketballDrive*, the green box is drawn to locate the moving basketball.

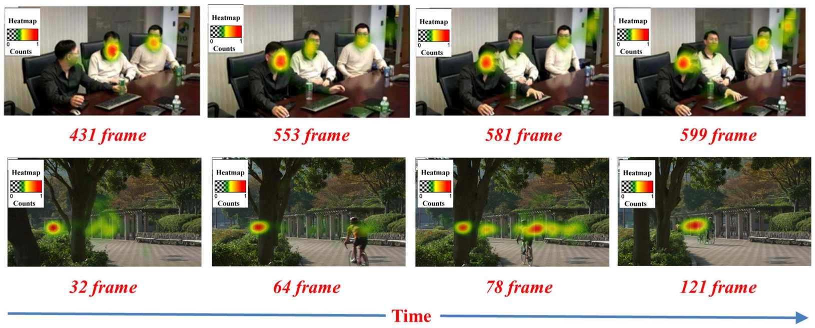


Fig. 4. Illustration of Observation 2. This figure shows the heat maps of visual attention of all 32 subjects, over several selected frames of videos *vidyo1* and *ParkScene*.

basketball and fixations becomes large, when the basketball moves at high speed. Besides, the second row of Figure 3 illustrates that the human fixations lag behind the new appearing objects by a few frames. It is because the human fixations still stay in the location of the salient region in previous frames, even when the scene has been changed. This completes the analysis of Observation 1.

Observation 2: Human fixations tend to be attracted by the new objects appearing in a video.

It is intuitive that visual attention is probably to be attracted by the objects newly emerging in a video. It is thus worth analyzing the influence of the object emergence on human visual attention. Figure 4 shows the heat maps of fixations on several frames selected from videos *vidyo1* and *ParkScene*. Note that a person appears in the door from the 553-*th* frame of the video *vidyo1*, and that a person riding bicycle arises from the 64-*th* frame of the video *ParkScene*. From Figure 4, one may observe that once a new object appears in the video, it probably attracts a huge amount of visual attention. This completes the analysis of Observation 2. Note that there exists the lag of human fixations, as the door is still fixated on when the person has left. This also satisfies Observation 1.

Observation 3: The object, which moves in the opposite direction of the surrounding objects, is possible to receive extensive fixations.

The previous work [10] has verified that the human fixations on still images are influenced by the center-surround features of color and intensity. Actually, the center-surround feature of motions also has an important effect on attracting visual attention. As seen from Figure 5, the old man with a trolley moves in the opposite direction of the surrounding crowd, and he attracts the majority of visual attention. Therefore, this suggests that the object moving in the opposite direction to its surround (i.e., it is with large center-surround motion) may receive extensive fixations. This completes the analysis of Observation 3.

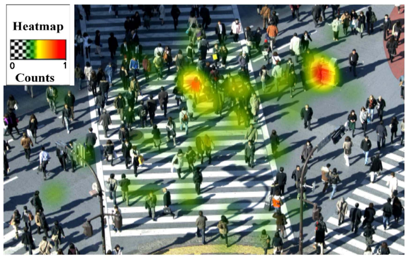


Fig. 5. Illumination of Observation 3. This figure shows the map of human fixations of all 32 subjects, over a selected frame of video *PeopleOnStreet*. Note that in the video a lot of visual attention is attended to the old man, who pushes a trolley and walks in the opposite direction of the crowd.

IV. HEVC Features for Saliency Detection

In this section, we mainly focus on exploring the features in HEVC domain, which can be used to efficiently detect video saliency. As analyzed above, three HEVC features, i.e., splitting depth, bit allocation, and MV, are effective in predicting video saliency. Therefore, they are worked out as the basic features of video saliency detection, to be presented in

Section IV-A. Note that the camera motion has to be removed for the MV feature, with an efficient algorithm developed in Section IV-A. Based on the three basic HEVC features, the features on temporal and spatial difference are discussed in Sections IV-B and IV-C, respectively.

A. Basic HEVC Features

1) Splitting Depth: The CTU partition structure [29], a new technique introduced by HEVC, can offer more flexible block sizes in video coding. In HEVC, the block sizes range from 64×64 to 8×8. In other words, the splitting depth varies from 0 ($= 64 \times 64$ block size) to 3 ($= 8 \times 8$ block size). In HEVC, rather than raw pixels, the residual of each coding block is encoded, which reflects spatial texture in intra-frame prediction and temporal variation in inter-frame prediction. Consequently, in intra-frame prediction, splitting depth of each CTU can be considered to model spatial saliency. In inter-frame prediction, splitting depth of each coding block can be used to model temporal saliency. Since Section III-B has demonstrated that most fixations fall into groups with high-valued splitting depths, the splitting depth of each CU is applied as a basic HEVC feature in video saliency detection.

Let d_{ij}^k be the normalized splitting depth of pixel (i, j) at the $k\text{-}th$ frame. First, the splitting depths of all CUs need to be extracted from HEVC bitstreams. Then, we assume that the splitting depth of each pixel is equivalent to that of its corresponding CU. Afterwards, all splitting depths should be normalized by the maximal splitting depth in each video frame. At last, all normalized d_{ij}^k can be yielded as one basic feature of our method.

2) Bit Allocation: Since the work of [30] is a state-of-the-art rate control scheme for HEVC, it has been embedded into the latest HEVC reference software (HM 16.0) for assigning bits to different CTUs. In the work of [30], the rate-distortion is optimized in each video frame, such that the CTUs with high-information are generally encoded by more bits. It has been argued in [2] that high-information regions attract extensive visual attention. Thus, the bits, allocated by [30] in HEVC, are considered a basic feature, modelling spatial saliency in intra frame prediction and temporal saliency in inter frame prediction. Specifically, Section III-B has shown that visual attention is highly correlated with the bit allocation of each CTU. Thereby, bit per pixel (bpp) is extracted from HEVC bitstreams, towards saliency detection. Let b_{ij}^k denote the normalized bpp of pixel (i, j) at the $k\text{-}th$ frame. Here, the bpp is achieved via averaging all consumed bits in the corresponding CTU. Next, the bpp is normalized to be b_{ij}^k in each video frame, and it is then included as one of basic HEVC features to detect saliency.

3) Motion Vector: In video coding, MV identifies the location of matching prediction unit (PU) in the reference frame. In HEVC, MV is sophisticatedly developed to indicate motion between neighboring frames. Intuitively, MV can be used to detect video saliency, as motion is an obvious cue [16] of salient regions. This intuition has also been verified by the statistical analysis of Section III-B. Therefore, MV is extracted as a basic HEVC feature in our method.

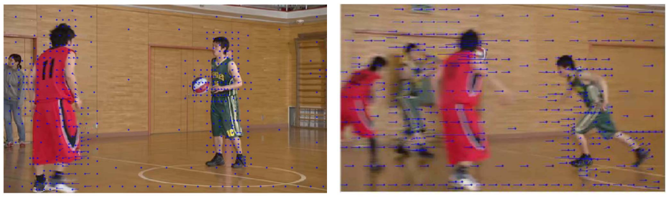


Fig. 6. An example of MV values of all PUs in (a) a frame with no camera motion, and (b) a frame with right-to-left camera motion. Note that the MVs are extracted from HEVC bitstreams. In (a) and (b), the dots stand for the origin of each MV, and the blue lines mean the intensity and angle of each MV. It can be seen that in (a) there is no camera motion, as most MV values are close to zero, whereas the camera motion in (b) is from right to left according the most MV values.

During video coding, MV is accumulated by two factors: the camera motion and object motion. It has been pointed out in [43] that in a video, moving objects may receive extensive visual attention, while static background normally draws little attention. It is thus necessary to distinguish moving objects and static background. Unfortunately, MVs of static background may be as large as moving objects, due to the camera motion. On the other hand, although temporal difference of MVs is able to make camera motion negligible for static background, it may also miss the moving objects. Therefore, the camera motion has to be removed from calculated MVs, to estimate object motion for saliency detection.

Figure 6 shows that the camera motion can be estimated to be the dominant MVs in a video frame. In this paper, we therefore develop a voting algorithm to estimate the motion of camera. Assuming that $\mathbf{m}_{ij}^k$ is the two-dimensional MV of pixel (i, j) at the $k\text{-}th$ frame, the dominant camera motion $\mathbf{m}_c^k$ in this frame can be determined in the following way.

First, the static background $\mathbf{S}_b^k$ is roughly extracted to be

$$\mathbf{S}_b^k = \{(i, j)|d_{ij}^k \cdot b_{ij}^k < \frac{1}{|\mathbf{I}^k|} \sum_{(i',j')\in\mathbf{I}^k} d_{i'j'}^k \cdot b_{i'j'}^k\}, \qquad (1)$$

for the $k\text{-}th$ frame $\mathbf{I}^k$ (with $|\mathbf{I}^k|$ pixels). It is because the static background is generally with less splitting depth and bit allocation than the moving foreground objects. Then, the azimuth $a(\mathbf{m}_c^k)$ for the dominant camera motion can be calculated via voting all MV angles in the background $\mathbf{S}_b^k$ as,

$$\max \text{hist}(\bigcup_{i,j\in\mathbf{S}_b^k} a(\mathbf{m}_{ij}^k)), \qquad (2)$$

where $a(\mathbf{m}_{ij}^k)$ is the azimuth for MV $\mathbf{m}_{ij}^k$, and $\text{hist}(\cdot)$ is the azimuth histogram of all MVs. In this paper, 16 bins with equal angle width ($= 360°/16 = 22.5°$) are applied for the histogram. After obtaining $a(\mathbf{m}_c^k)$, radius $r(\mathbf{m}_c^k)$ for the camera motion needs to be calculated via averaging over all MVs from the selected bin of $a(\mathbf{m}_c^k)$. Finally, the camera motion of each frame can be achieved upon $a(\mathbf{m}_c^k)$ and $r(\mathbf{m}_c^k)$. For justification, we show in Figure 7 some subjective results of the camera motion estimated by our voting algorithm (in yellow arrows), as well as the annotated ground truth of camera motion (in blue arrows). As can be seen from this figure, our algorithm is capable of accurately estimating the

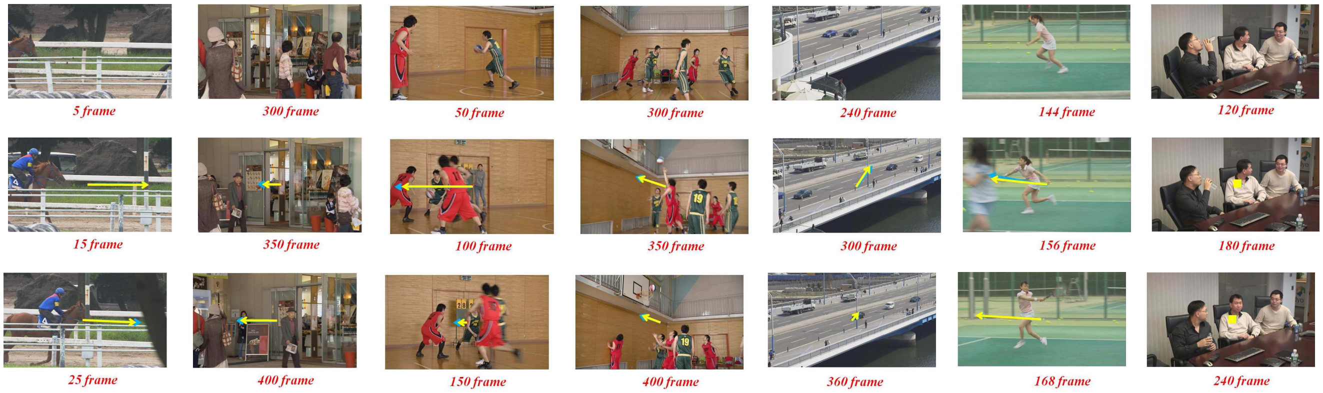


Fig. 7. The results of camera motion estimation, yielded by our voting algorithm. The first six videos are with some extended camera motion, whereas the last one is without any camera motion. In (b), the yellow and blue arrows represent the estimated and manually annotated vectors of the camera moving from frames of (a) to frames of (b), respectively. Similarly, the yellow and blue arrows of (c) show the camera motion from frames of (b) to (c). Refer to Appendix for the way of annotating ground truth camera motion.

camera motion. See Appendix for more justification on the estimation of camera motion.

Next, in order to track the motion of objects, all MVs obtained in HEVC domain need to be processed to remove the estimated camera motion. All processed MVs should be then normalized in each video frame, denoted as $\hat{\mathbf{m}}_{ij}^k$. Since it has been argued in [16] that visual attention is probably attracted by moving objects, $||\hat{\mathbf{m}}_{ij}^k||_2$ is utilized as one of the basic HEVC features to predict video saliency.

B. Temporal Difference Features in HEVC Domain

As revealed in **Observation 2**, humans tend to fixate on the new objects appearing in a video. In fact, the new appearing or moving objects in the video also lead to large temporal difference of HEVC features in co-located regions of neighboring frames. Hence, the temporal difference features, which quantify the dissimilarity of splitting depth, bit allocation and MV across neighboring frames, are developed as novel HEVC features in our method. However, the temporal difference in co-located region across video frames refers to the sum of object motion and camera motion. It has been figured out in [43] that moving objects attract extensive visual attention, whereas camera motion receives little attention. Therefore, when developing temporal difference features, camera motion needs to be removed to compensate object motion (to be discussed in the following).

Specifically, let us first look at the way on estimating temporal difference of splitting depths. For pixel (i, j) at the k-th frame, $\Delta_t d_{ij}^k$ is defined as the difference value of splitting depth across neighboring frames. It can be calculated by averaging the weighted difference values of the splitting depths over all previous frames,

$$\Delta_t d_{ij}^k = \frac{\sum_{l=1}^k \exp(-\frac{l^2}{\sigma_d^2})||d_{ij}^k - d_{ij}^{k-l}||_1}{\sum_{l=1}^k \exp(-\frac{l^2}{\sigma_d^2})}, \tag{3}$$

where parameter σ_d controls the weights on splitting depth difference between two frames. In (3), d_{ij}^{k-l} is the splitting depth of pixel (i, j) at the $(k-l)$-th frame. After considering the camera motion with our voting algorithm, we assume that $(i^{k,l}, j^{k,l})$ is the pixel at the $(k-l)$-th frame matching to pixel (i, j) at the k-th frame. To remove the influence of the camera motion, we replace d_{ij}^{k-l} in (3) by $d_{i^{k,l} j^{k,l}}^{k-l}$. Then, (3) is rewritten to be

$$\Delta_t d_{ij}^k = \frac{\sum_{l=1}^k \exp(-\frac{l^2}{\sigma_d^2})||d_{ij}^k - d_{i^{k,l} j^{k,l}}^{k-l}||_1}{\sum_{l=1}^k \exp(-\frac{l^2}{\sigma_d^2})}. \tag{4}$$

After calculating (4), $\Delta_t d_{ij}^k$ needs to be normalized in each video frame, as one of temporal difference features in HEVC domain.

Furthermore, the bpp difference across neighboring frames is also regarded as a feature for saliency detection. Let $\Delta_t b_{ij}^k$ denote the temporal difference of the bpp at pixel (i, j) between the currently processed k-th frame and its previous frames. Similar to (4), $\Delta_t b_{ij}^k$ can be obtained by

$$\Delta_t b_{ij}^k = \frac{\sum_{l=1}^k \exp(-\frac{l^2}{\sigma_b^2})||b_{ij}^k - b_{i^{k,l} j^{k,l}}^{k-l}||_1}{\sum_{l=1}^k \exp(-\frac{l^2}{\sigma_b^2})}, \tag{5}$$

where σ_b decides the weights of the bpp difference between frames. In (5), with the compensated camera motion, $b_{i^{k,l} j^{k,l}}^{k-l}$ is the bpp for pixel $(i^{k,l}, j^{k,l})$ at the $(k-l)$-th frame, which matches to pixel (i, j) at the k-th frame.

Finally, the temporal difference of MV is also taken into account, by adopting the similar way presented above. Recall that $\hat{\mathbf{m}}_{ij}^k$ is the extracted MV of each pixel, with the camera motion being removed. Since $\hat{\mathbf{m}}_{ij}^k$ is a 2D vector, ℓ_2-norm operation is applied to compute the temporal difference of MVs (denoted by $\Delta_t \hat{m}_{ij}^k$) as follows,

$$\Delta_t \hat{m}_{ij}^k = \frac{\sum_{l=1}^k \exp(-\frac{l^2}{\sigma_m^2})||\hat{\mathbf{m}}_{ij}^k - \hat{\mathbf{m}}_{i^{k,l} j^{k,l}}^{k-l}||_2}{\sum_{l=1}^k \exp(-\frac{l^2}{\sigma_m^2})}. \tag{6}$$

In (6), we can use parameter σ_m to determine the weights of MV difference between two frames. Moreover, $\hat{\mathbf{m}}_{i^{k,l} j^{k,l}}^{k-l}$ is the

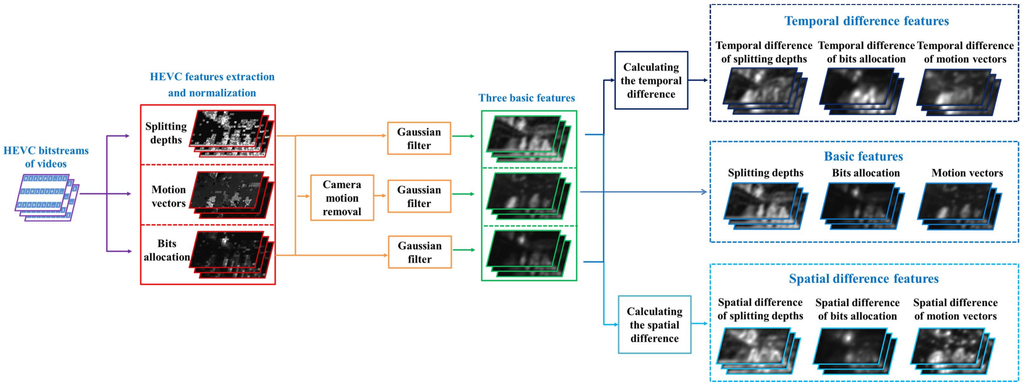

HEVC bitstreams of videos
HEVC features extraction and normalization
Splitting depths
Motion vectors
Bits allocation
Camera motion removal
Gaussian filter
Gaussian filter
Gaussian filter
Three basic features
Calculating the temporal difference
Temporal difference features
Temporal difference of splitting depths
Temporal difference of bits allocation
Temporal difference of motion vectors
Basic features
Splitting depths
Bits allocation
Motion vectors
Calculating the spatial difference
Spatial difference features
Spatial difference of splitting depths
Spatial difference of bits allocation
Spatial difference of motion vectors

Fig. 8. Framework of our HEVC feature extractor for video saliency detection.

MV value for pixel $(i^{k,l}, j^{k,l})$ at the $(k-l)$-th frame, which is the co-located pixel of (i, j) at the k-th frame, after the camera motion is removed by our voting algorithm.

C. Spatial Difference Features in HEVC Domain

The above features are not sufficient to model saliency in a video, since some smooth regions may stand out from complicated background for drawing attention (like a salient smooth ball appearing in grass land). Generally speaking, the basic features of splitting depth and bit allocation in a smooth region are significantly different from those in its surrounding background. Thus, we here develop spatial difference features for saliency detection. In addition, according to **Observation 3**, the object moving in the opposite direction to the nearby objects may result in extensive visual attention. Actually, the dissimilarity of object motion can be measured by the spatial difference of MVs between neighboring PUs. Hence, the spatial difference of all three basic features is incorporated into our method, as follows.

Recall that $\mathbf{I}^k$ is the k-th video frame, and that d^k_{ij}, b^k_{ij}, and $\mathbf{m}^k_{ij}$ denote the splitting depth, bit allocation, and MV for pixel (i, j) of this video frame. For the spatial difference of MV, the camera motion has to be removed in each $\mathbf{m}^k_{ij}$, defined by $\hat{\mathbf{m}}^k_{ij}$. Then, we have

$$\begin{cases} \Delta_s d^k_{ij} = \dfrac{\sum_{(i',j')\in\mathbf{I}^k} \exp(-\frac{(i'-i)^2+(j'-j)^2}{\xi_d^2})||d^k_{i'j'} - d^k_{ij}||_1}{\sum_{(i',j')\in\mathbf{I}^k} \exp(-\frac{(i'-i)^2+(j'-j)^2}{\xi_d^2})} \\ \Delta_s b^k_{ij} = \dfrac{\sum_{(i',j')\in\mathbf{I}^k} \exp(-\frac{(i'-i)^2+(j'-j)^2}{\xi_b^2})||b^k_{i'j'} - b^k_{ij}||_1}{\sum_{(i',j')\in\mathbf{I}^k} \exp(-\frac{(i'-i)^2+(j'-j)^2}{\xi_b^2})} \\ \Delta_s \hat{\mathbf{m}}^k_{ij} = \dfrac{\sum_{(i',j')\in\mathbf{I}^k} \exp(-\frac{(i'-i)^2+(j'-j)^2}{\xi_m^2})||\hat{\mathbf{m}}^k_{i'j'} - \hat{\mathbf{m}}^k_{ij}||_2}{\sum_{(i',j')\in\mathbf{I}^k} \exp(-\frac{(i'-i)^2+(j'-j)^2}{\xi_m^2})}, \end{cases} \tag{7}$$

to compute the spatial difference of splitting depth, bit allocation, and MV. As in the above equations, ξ_d, ξ_b, and ξ_m are the parameters to control the spatial weighting of each feature.

Finally, all nine features in HEVC domain can be achieved in our saliency detection method. Since all the proposed HEVC features are block-wise, the block-to-pixel refinement is required to obtain smooth feature maps. For the block-to-pixel refinement, a 2D Guassian filter is applied to three basic features. In this paper, the dimension and standard deviation of the Gaussian filter are tuned to be $\frac{2h}{15} \times \frac{2h}{15}$ and $\frac{h}{30}$, where h is the height of the video. It is worth mentioning that the above features on spatial and temporal difference are explored in compressed domain with the block-to-pixel refinement, while the existing methods compute contrast features in pixel domain (e.g., in [10] and [11]). Additionally, unlike the existing methods, the camera motion is estimated and removed when calculating the feature contrast in our method. Despite simple and straightforward, these features are effective and efficient, as evaluated in experiment section.

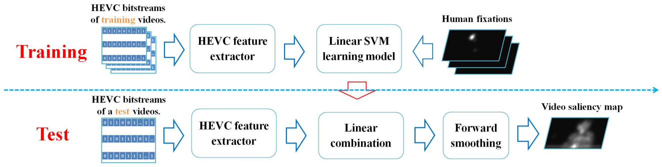


Fig. 9. Framework of our learning based method for video saliency detection with HEVC features. For the HEVC feature extractor, refer to Figure 8.

Figure 8 summarizes the procedure of HEVC feature extraction in our saliency detection method. As seen from Figure 8, the maps of nine features have been obtained, based on splitting depth, bit allocation and MV of HEVC bitstreams. We argue that one single feature is not capable enough [2] but has different impact on saliency detection. We thus integrate the maps of all nine features with the learned weights. For more details, refer to the next section.

V. Learning Based Video Saliency Detection

This section mainly concentrates on learning an SVM classifier to detect video saliency, using the above mentioned nine HEVC features. The framework of our learning based method is summarized in Figure 9. As shown in this figure, given the HEVC bitstreams, all HEVC features need to be extracted and calculated. Then, the saliency map of each single video frame is yielded by combining the HEVC features with C-support vector classification (C-SVC), which is a kind of non-linear SVM classifer. Here, the C-SVC classifer is learned from the ground-truth human fixations of training videos. At last, a simple forward smoothing filter is applied to the yielded saliency maps across video frames, outputting the final video saliency maps. More details about our learning based method are to be discussed in the following.

A. Training Algorithm

In our method, the non-linear C-SVC [44], a kind of SVM, is trained as the binary classifier to decide how possible each pixel attracts attention, according to the proposed HEVC features. First, for the binary classifier, both positive and negative samples need to be obtained from the training set, in which the positive samples mean the pixels attracting fixations and negative samples indicate the pixels without any visual attention. Next, three basic HEVC features of each training sample are extracted from the HEVC bitstreams, and then other spatial and temporal features are computed upon the corresponding basic features. Let $\{(\mathbf{f}_n, l_n)\}_{n=1}^{N}$ be those training samples, where $\mathbf{f}_n$ is the vector of the nine HEVC features for the $n\text{-}th$ training sample, and $l_n \in \{-1, 1\}$ is the class label indicating whether the sample is positive ($\text{ł}_n = 1$) or negative ($\text{ł}_n = -1$). Finally, the C-SVC for saliency detection can be worked out, via solving the following optimization problem,

$$\min_{\mathbf{w}, b, \{\beta_n\}_{n=1}^{N}} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{n=1}^{N}\beta_n$$
$$\text{s.t.} \quad \forall n, \quad l_n(\mathbf{w}^T \cdot \phi(\mathbf{f}_n) + b) \geq 1 - \beta_n, \quad \beta_n \geq 0. \quad (8)$$

In (8), $\mathbf{w}$ and b are the parameters to be learned for maximizing the margin between positive and negative samples, and β_n is a non-negative slack variable evaluating the degree of classification error of $\mathbf{f}_n$. In addition, C balances the trade-off between the error and margin. Function $\phi(\cdot)$ transforms the training vector of HEVC features $\mathbf{f}_n$ to higher dimensional space. Then, $\mathbf{w}$ can be seen as the linear combination of transformed vectors:

$$\mathbf{w} = \sum_{m=1}^{N} \lambda_m l_m \cdot \phi(\mathbf{f}_m), \quad (9)$$

where λ_m is the Lagrange multiplier to be learned. Then, the following holds,

$$\begin{aligned}\mathbf{w}^T \cdot \phi(\mathbf{f}_n) &= \left(\sum_{m=1}^{N} \lambda_m l_m \cdot \phi(\mathbf{f}_m)\right)^T \cdot \phi(\mathbf{f}_n) \\ &= \sum_{m=1}^{N} \lambda_m l_m \cdot \langle \phi(\mathbf{f}_m), \phi(\mathbf{f}_n)\rangle. \quad (10)\end{aligned}$$

Note that $\langle \phi(\mathbf{f}_m), \phi(\mathbf{f}_n)\rangle$ indicates the inner product of $\phi(\mathbf{f}_m)$ and $\phi(\mathbf{f}_n)$. To calculate (10), a kernel of radial bias function (RBF) is introduced:

$$K(\mathbf{f}_m, \mathbf{f}_n) = \langle \phi(\mathbf{f}_m), \phi(\mathbf{f}_n)\rangle = \exp(-\gamma\,||\mathbf{f}_m - \mathbf{f}_n||_2^2), \quad (11)$$

where γ (> 0) stands for the kernel parameter. Here, we utilize the above KBF kernel due to its simplicity and effectiveness. When training the C-SVC for saliency detection, the penalty parameter C in (8) is set to 2^{-3}, and γ of the KBF kernel is tuned to be 2^{-15}, such that the trained C-SVC is rather efficient in detecting saliency. Finally, $\mathbf{w}$ and b can be worked out in the trained C-SVC as the model of video saliency detection, to be discussed below.

B. Saliency Detection

To detect the saliency of test videos, all nine HEVC features are integrated together using the learned $\mathbf{w}$ and b of our C-SVC classifier. Then, the saliency map $\mathbf{S}_k$ for each single video frame can be yielded by

$$\mathbf{S}_k = \mathbf{w}^T \cdot \phi(\mathbf{F}_k) + b, \quad (12)$$

where $\mathbf{F}_k$ defines the pixel-wise matrix of nine HEVC features at the k-th video frame. Note that $\mathbf{w}$ in (12) is one set of weights for the binary classifier of C-SVC, which have been obtained using the above training algorithm.

Since **Observation 1** offers a key insight that visual attention may lag behind the moving or new appearing objects, a forward smoothing filter is developed in our method to take into account the saliency maps of previous frames. Mathematically, the final saliency map $\hat{\mathbf{S}}_k$ of the $k\text{-}th$ video frame is calculated by the forward smoothing filter as follows,

$$\hat{\mathbf{S}}_k = \frac{1}{\lceil t \cdot \text{fr}\rceil} \sum_{k'=k-\lceil t\cdot\text{fr}\rceil+1}^{k} \mathbf{S}_{k'}, \quad (13)$$

where t (> 0) is the time duration[1] of the forward smoothing, and fr is the frame rate of the video. Note that a simple forward smoothing filter of (13) is utilized here, since we mainly concentrate on extracting and integrating features for saliency detection. Some advanced tracking filters may be applied, instead of the forward smoothing filter in our method, for further improving the performance on saliency detection. To model visual attention on video frames, the final saliency maps need to be smoothed with a 2D Gaussian filter, which is in addition to the one for each single feature map (as shown in Figure 8). Note that the 2D Gaussian filter here shares the same parameters as those for single feature maps.

VI. Experimental Results

In this section, we present the experimental results on video saliency detection to validate the performance of our method. Section VI-A shows the settings of our method, and Section VI-B discusses the parameter selection in our method. Sections VI-C and VI-D compare the saliency detection results by our and other 7 methods, over our and other 2 public databases, respectively. For comparing the accuracy of saliency detection, receiver operating characteristic (ROC) curves, the equal error rate (EER), the area under ROC curve (AUC), normalized scanpath saliency (NSS), linear correlation coefficient (CC), and KL were measured on the saliency maps generated by our and other 7 methods. Section VI-E evaluates the performance of our method at different working conditions.

[1] We found out through experiments that $t = 0.3$ second makes the saliency detection accuracy highest. So, time duration t of our forward smoothing was set to be 0.3 in Section VI.

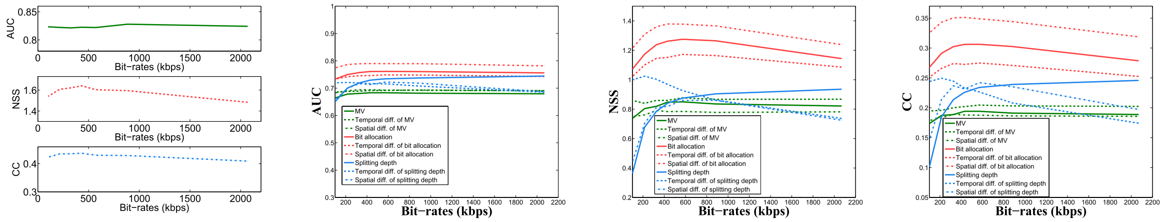


Fig. 10. Performance comparison of our method (first column) and our single features (second to fourth columns) at different bit-rates. The bit-rates of each video in our rate control are the same as those of fixed QPs, i.e., QP = 27, 32, 35, 37, 39, 42, and 47. Here, the bit-rates averaged over all 33 videos are shown in the horizontal axis.

In Section VI-F, we demonstrate the effectiveness of each single HEVC feature in saliency detection.

A. Setting on Encoding and Training

1) HEVC Configuration: Before saliency detection, the bitstreams of both training and test videos were generated by the HEVC encoder, for extracting features. In our experiments, the HEVC reference software HM 16.0 [45] was used as the HEVC encoder. Then, the HEVC bitstreams of all 33 videos in our database were produced for both training and test. In HM 16.0, the low delay (LD) P main configuration was chosen. In addition, the latest $R - \lambda$ rate control scheme [30] was enabled in HM 16.0. Since the test videos are with diverse content and resolutions, we followed the way of [30] to set the bit-rates the same as those at fixed QPs. The CTU size was set to 64×64 and maximum CTU depth was 3, to allow all possible CTU partition structures for saliency detection. Each group of picture (GOP) was composed of 4 P frames. Other encoding parameters were set by default, using the common *encoder_lowdelay_P_main.cfg* configuration file of HM.

2) Other Working Conditions: The implementation of our method in random access (RA) configuration is to be presented in Section VI-E. The rate control of RA in HM 16.0 was also enabled. In our experiments, we set all other parameters of RA via the *encoder_randomaccess_main.cfg* file. Note that the GOP of RA is 8 B frames for HM 16.0. Section VI-E further presents the saliency detection results of our method for the bitstreams of x265, which is more practical than the HM encoder from the aspects of encoding and decoding time.[2] Here, x265 v1.8 encoder, embedded in the latest ffmpeg, was applied. For x265, both LD and RA were tested. In x265, the bit-rates were chosen using the same way as we applied for HM 16.0. The GOP structure is 4 P frames for LD and 4 frames (BBBP) for RA. Other parameters were all set by default in the ffmpeg with the x265 codec. It is worth pointing out that the x265 codec was used to extract features from the bitstreams encoded by x265, while the features of HM 16.0 bitstreams were extracted by the software of HM 16.0.

3) Training Setting: In order to train the C-SVC, our database of Section III-A was divided into non-overlapping sets. For the fair evaluation, 3-fold cross validation was conducted in our experiments, and the averaged results are reported in Sections VI-B and VI-C. Specifically, our database was equally partitioned into three non-overlapping sets. Then, two sets were used as training data, and the remaining set was retained for validating saliency detection. The cross validation process is repeated by three folds, with each of the three sets being used exactly once as the validation data. In the training set, 3 pixels of each video frame were randomly selected from top 5% salient regions of ground-truth fixation maps as the positive samples. Similarly, 3 pixels of each video frame were further chosen from bottom 70% salient regions as negative samples. Then, both positive and negative samples were available in each cross validation, to train the C-SVC with (8).

B. Analysis on Parameter Selection

In HEVC, the bit allocation, splitting depth and MV of each CTU may change along with increased or decreased bit-rates. Therefore, we analyze the performance of our method with regard to the videos compressed at different bit-rates. Since the resolutions of test videos vary from 416×240 to 1920×1080, there is an issue on finding bit-rates suitable for all videos to ensure proper visual quality. To solve such an issue, we followed [30] in setting the bit-rates of each video for rate control the same as those of fixed QPs. Then, we report in Figure 10 the AUC, CC and NSS results of our method at different bit-rates. Note that the bit-rates averaged over all 33 videos are shown, varying from 2,068 kbps to 100 kbps. Figure 10 shows that our method achieves the best performance in terms of CC and NSS, when the averaged bit-rate of rate control is 430 kbps (equal to those of fixed $\mathrm{QP} = 37$). Therefore, such bit-rate setting is used for the following evaluation. Figure 10 also shows that the bit-rates have slight impact on the overall performance of our method in terms of AUC, NSS, and CC. The minimum values of AUC, NSS and CC are above 0.82, 1.52 and 0.41 at different bit-rates, which are superior to all other methods reported in Section VI-C. Besides, one may observe from Figure 10 that the saliency detection accuracy of some HEVC features is fluctuating when the bit-rate is changed. Hence, this figure suggests that our saliency detection should not rely on a single feature. On the contrary, the combination of all features is robust across various bit-rates, implying the benefit of applying the C-SVC in learning to integrate all HEVC features for saliency detection.

Next, we analyze the parameters of our saliency detection method. When computing the spatial difference features

[2] It takes around 100 seconds for HM to encode a 1080p video frame, in a PC with Intel Core i7-4770 CPU and 16 GB RAM. By contrast, x265 adopts parallel computing and fast methods to encode videos, such that real-time 4K HEVC encoding can be achieved by x265.

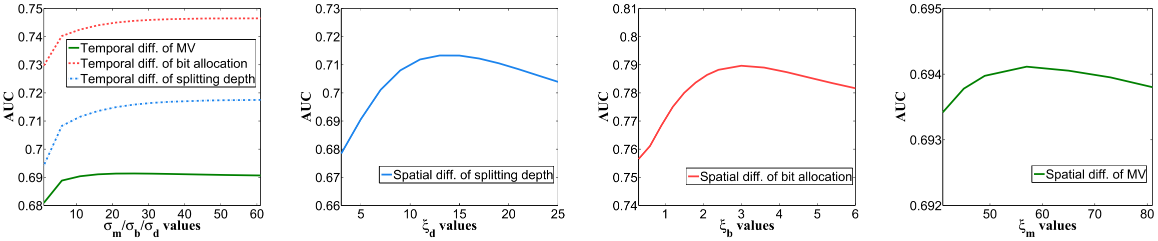


Fig. 11. Saliency detection performance of each single feature at different parameter settings. Note that only the AUC is utilized here to evaluate the saliency detection performance. For other metrics (e.g., NSS and CC), similar results can be found for choosing the optimal values of parameters.

TABLE I

THE AVERAGED ACCURACY OF SALIENCY DETECTION BY OUR AND OTHER 7 METHODS, IN MEAN (STANDARD DEVIATION) OF ALL TEST VIDEOS OF 3-FOLD CROSS VALIDATION OVER OUR DATABASE.

	Our	Itti [10]	Surprise [14]	Judd [19]	PQFT [16]	Rudoy [23]	Fang [27]	OBDL [28]
AUC	**0.823**(0.071)	0.688(0.066)	0.752(0.083)	0.816(0.065)	0.750(0.084)	0.785(0.100)	0.797(0.073)	0.785(0.086)
NSS	**1.658**(0.591)	0.445(0.464)	1.078(0.739)	1.427(0.440)	1.300(0.529)	1.401(0.708)	1.306(0.560)	1.511(0.825)
CC	**0.438**(0.133)	0.119(0.098)	0.272(0.156)	0.387(0.111)	0.311(0.121)	0.386(0.186)	0.370(0.133)	0.352(0.166)
KL	**0.300**(0.086)	0.104(0.043)	0.183(0.086)	0.285(0.076)	0.239(0.076)	0.269(0.111)	0.266(0.081)	0.236(0.111)
EER	**0.241**(0.075)	0.365(0.051)	0.305(0.075)	0.250(0.064)	0.307(0.074)	0.270(0.094)	0.269(0.071)	0.277(0.098)

through (7), parameters ξ_d, ξ_b, and ξ_m were all traversed to find the optimal values. The results are shown in Figure 11. As can be seen in this figure, parameters ξ_d, ξ_b, and ξ_m should be set to 13, 3 and 57 for optimizing saliency detection results. In addition, the saliency detection accuracy of temporal difference features almost reaches the maximum, when σ_d, σ_b and σ_m of (4), (5) and (6) are equivalent to 46, 46 and 26. Finally, we achieve the optimal parameter selection for the following evaluation (i.e., $\xi_d = 13$, $\xi_b = 3$, $\xi_m = 57$, $\sigma_d = 46$, $\sigma_b = 46$ and $\sigma_m = 26$).

The effectiveness of the center bias in saliency detection has been verified in [46], as humans tend to pay more attention on the center of the image/video than the surround. In this paper, we follow [46] to impose the same center bias map **B** to both our and other compared methods, for fair comparison. Specifically, the center bias is based on the Euclidean distance of each pixel to video frame center (i_c, j_c) as follows,

$$\mathbf{B}(i, j) = 1 - \frac{\sqrt{(i - i_c)^2 + (j - j_c)^2}}{\sqrt{i_c^2 + j_c^2}}, \quad (14)$$

where $\mathbf{B}(i, j)$ is the center bias value at pixel (i, j). Then, the detected saliency maps of all methods are weighted by the above center bias maps.

C. Evaluation on Our Database

In this section, we evaluate the saliency detection accuracy of our method, in comparison with other 7 state-of-the-art methods,[3] i.e., Itti's model [10], Bayesian surprise [14], Judd *et al.* [19], PQFT [16], Rudoy *et al.* [23], Fang *et al.* [27] and OBDL [28]. Note that 3-fold cross validation was applied in our database for evaluation, and the saliency detection accuracy was averaged over the frames of all test videos of 3-fold cross validation. Furthermore, the saliency maps of some selected video frames are provided for each cross validation, to show the subjective saliency detection results of our and other methods.

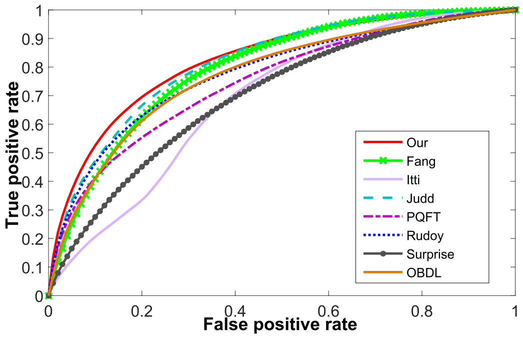


Fig. 12. ROC curves of saliency detection by our and other state-of-the-art methods. Note that the results are averaged over frames of all test videos of 3-fold cross validation.

1) ROC Curves: The ROC curves of our and other 7 methods are shown in Figure 12, to evaluate the accuracy of saliency detection in predicting human fixations. As can be seen in this figure, our method generally has higher true positive rates than others at the same false positive rates. In a word, the ROC curves illustrate the superior performance of our method in saliency detection.

2) AUC and EER: In order to quantify the ROC curves, we report in Table I the AUC and EER results of our and other 7 state-of-the-art methods. Here, both mean and standard deviation are provided for the AUC and EER results of all test video frames of 3-fold cross validation. This table shows that our method performs better than all other 7 methods. Specifically, there are 0.026 and 0.038 enhancement of AUC, over Fang *et al.* [27] and OBDL [28], respectively, which also work in compressed domain. The EER of our method has 0.028 and 0.036 decrease, compared with compressed domain methods of [27] and [28]. Smaller EER means that there is

[3] In our experiments, we directly used the codes by the authors to implement all methods except Fang *et al.* [27], which was realized by ourselves as the code is not available online.

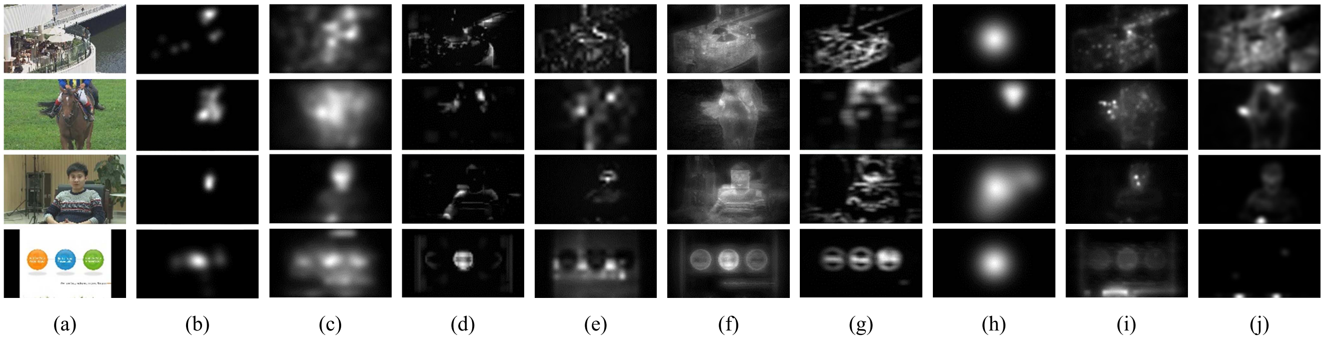


Fig. 13. Saliency maps of four videos selected from the first time of our cross validation experiments. The maps were yielded by our and other 7 methods as well the ground-truth human fixations. Note that the results of only one frame are shown for each selected video. (a) Input. (b) Human. (c) Our. (d) Itti. (e) Surprise. (f) Judd. (g) PQFT. (h) Rudoy. (i) Fang. (j) OBDL.

a lower miss-classifying probability in our method when the false positive rate equals to the false negative rate. The possible reasons for the improvement of our method are: (1) the new compressed domain features (i.e., CTU structure and bit allocation) are developed in light of the latest HEVC standard; (2) the camera motion has been removed in our method; (3) the learning mechanism is incorporated into our method to bridge the gap between HEVC features and human visual attention. Besides, our method outperforms uncompressed domain learning-based methods [19] and [23], with 0.007 and 0.038 improvement in AUC as well as 0.009 and 0.029 reduction in EER. This verifies the effectiveness of the newly proposed features in compressed domain, which benefit from the well developed HEVC standard. However, since extensive high and middle level features are applied in [19], there is little AUC improvement (around 0.007) of our method over [19]. Generally speaking, our method outperforms all other 7 methods, which are in compressed or uncompressed domain.

3) NSS, CC, and KL: Now, we concentrate on the comparison of NSS, CC, and KL metrics to evaluate the accuracy of saliency detection on all test videos. The averaged results (with their standard deviation) of NSS, CC and KL, by our and other 7 state-of-the-art methods, are also reported in Table I. Note that the method with a higher value of NSS, CC or KL, can better predict the human fixations. Again, it can be seen from Table I that our method improves the saliency detection accuracy over all other methods, in terms of NSS, CC and KL. Moreover, the improvement of NSS, CC and KL, especially CC, is much larger than that of AUC.

4) Saliency Maps: Figure 13 shows the saliency maps of 4 randomly selected test videos, detected by our and other 7 methods, as well as the ground-truth human fixation maps. Note that the results of only one frame for each video are shown in these figures. From these figures, one may observe that in comparison with all other 7 methods, our method is capable of well locating the saliency regions in a video frame, much closer to the maps of human fixations. In summary, the subjective results here, together with the objective results above, demonstrate that our method is superior to other state-of-the-art methods in our database.

5) Computational Time: For time efficiency evaluation, the computational time of our and other methods have been recorded[4] and listed in Table II. We can see from this table that our method ranks third in terms of computational speed, only slower than Itti [10] and PQFT [16]. However, as discussed above, the performance of Itti and PQFT is rather inferior compared with other methods, and their saliency detection accuracy is much lower than that of our method. In summary, our method has high time efficiency with effective saliency prediction performance. The main reason is that our method benefits from the modern HEVC encoder and the learning mechanism, thus not wasting much time on exploiting saliency detection features. We further transplanted our method into C++ program on the VS.net platform to figure out its potential in real-time implementation. After the transplantation, our method consumes averaged 140 ms per frame over all videos of our database, and achieves real-time detection for 480p videos at 30 frame per second (fps). It is worth pointing out that some speeding-up techniques, like parallel computing, may further reduce the computational time of our method for real-time saliency detection of high resolution videos.

TABLE II
COMPUTATIONAL TIME PER VIDEO FRAME AVERAGED OVER OUR DATABASE FOR OUR AND OTHER 7 METHODS

	Our	Itti	Surprise	Judd	PQFT	Rudoy	Fang	OBDL
Time(s)	3.1	1.6	40.6	23.9	0.5	98.5	15.4	5.8

D. Evaluation on Other Database

For evaluating the generalization of our method, we compared our and other 7 methods on all videos of SFU [41] and DIEM [42], which are two widely used databases. In DIEM, the first 300 frames of each video were tested for matching the length of videos in SFU and our databases. Here, all 33 videos of our database were selected for training the C-SVC classifier. Table III presents the saliency detection accuracy of our and other methods over the SFU and DIEM databases. Again, our method performs much better than others in terms of all five metrics. Although the C-SVC was trained on our database, our method still significantly outperforms all 7 conventional methods over other databases.

[4] All methods were run in the same environment: Matlab 2012b at a computer with Intel Core i7-4770 CPU@3.4 GHz and 16 GB RAM.

TABLE III

MEAN (STANDARD DEVIATION) VALUES FOR SALIENCY DETECTION ACCURACY OF OUR AND OTHER METHODS OVER SFU AND DIEM DATABASES

	SFU							
	Our	Itti [10]	Surprise [14]	Judd [19]	PQFT [16]	Rudoy [23]	Fang [27]	OBDL [28]
AUC	**0.832**(0.06)	0.705(0.07)	0.658(0.12)	0.770(0.07)	0.729(0.08)	0.799(0.08)	0.801(0.07)	0.802(0.07)
NSS	**1.415**(0.34)	0.278(0.36)	0.479(0.58)	1.058(0.33)	0.867(0.45)	1.388(0.57)	1.236(0.40)	1.361(0.57)
CC	**0.492**(0.11)	0.090(0.09)	0.166(0.17)	0.372(0.10)	0.299(0.14)	0.464(0.16)	0.427(0.12)	0.448(0.16)
KL	**0.278**(0.07)	0.097(0.03)	0.131(0.08)	0.187(0.06)	0.198(0.07)	0.258(0.10)	0.244(0.07)	0.270(0.09)
EER	**0.239**(0.06)	0.346(0.06)	0.329(0.09)	0.298(0.06)	0.286(0.06)	0.268(0.07)	0.268(0.07)	0.261(0.06)
	DIEM							
	Our	Itti [10]	Surprise [14]	Judd [19]	PQFT [16]	Rudoy [23]	Fang [27]	OBDL [28]
AUC	**0.858**(0.07)	0.775(0.07)	0.754(0.12)	0.751(0.09)	0.795(0.08)	0.804(0.11)	0.808(0.09)	0.790(0.12)
NSS	**1.823**(0.65)	0.545(0.67)	0.935(0.91)	0.990(0.40)	1.282(0.75)	1.488(0.91)	1.232(0.57)	1.621(1.01)
CC	**0.488**(0.14)	0.133(0.12)	0.236(0.19)	0.295(0.11)	0.308(0.15)	0.412(0.22)	0.351(0.14)	0.394(0.22)
KL	**0.367**(0.10)	0.105(0.06)	0.198(0.13)	0.202(0.07)	0.257(0.11)	0.296(0.14)	0.283(0.10)	0.307(0.13)
EER	**0.210**(0.07)	0.287(0.07)	0.295(0.10)	0.310(0.08)	0.267(0.07)	0.252(0.10)	0.252(0.08)	0.261(0.11)

TABLE IV

COMPARISON TO THE RESULTS REPORTED IN [23]

	Our	PQFT [16]	Rudoy [23]
Median shuffled-AUC	**0.74**	0.68	0.72

Although above results were mainly upon the codes by their authors, it is more fair to compare with the results reported in their literature. However, it is hard to find the literature reporting the results of all 7 methods on a same database. Due to this, we only compare to the reported results of the method with top performance. We can see from Tables I and III that among all methods we compared, Rudoy et al. [23] generally ranks highest in our, SFU and DIEM databases. Thus, we implemented our method on the same database as Rudoy et al. [23] (also the DIEM database), and then we compared the results of our method to those of PQFT [16] and Rudoy et al. [23], which were reported in [23]. The comparison is provided in Table IV. Note that the comparison is in terms of median shuffled-AUC, as shuffled version of AUC was measured with median values available in [23]. Note that shuffled-AUC is much smaller than AUC, due to the removed center bias prior. We can see from Table IV that our method again performs better than [16] and [23].

E. Evaluation on Other Work Conditions

For further assessing the generalization of our method, we extended the implementation of our method at different HEVC working conditions. The working conditions include HM 16.0 and x265 v1.8 encoders, at both LD and RA configurations. We have discussed the parameter settings of these working conditions in Section VI-A. The rate control at these working conditions was also enabled, with the bit-rates the same as above.

Figure 14 compares the saliency detection performance of our method applied to HM and x265 encoders with LD and RA configurations. The performance is evaluated in terms of AUC, CC, NSS and KL, averaged over all videos of the three databases, i.e., our, SFU and DIEM databases. The results of Rudoy et al. [23] and Fang et al. [27] are also provided in this figure as the reference. As seen from Figure 14, although our method in RA performs a bit worse than that in LD, it is much superior to other state-of-the-art methods. We can further see from Figure 14 that the performance of our method slightly decreases, when using x265 bitstreams instead of HM bitstreams. Such a slight decrease is probably due to the simplified process of x265 over HM. More importantly, when applied to x265 bitstreams, our method still significantly outperforms other methods. In summary, our method is robust to different working conditions.

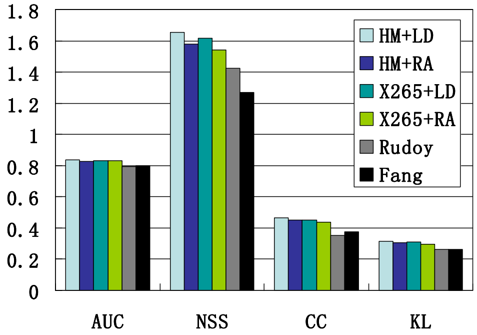


Fig. 14. Performance of our method at different working conditions, compared with Rudoy et al. [23] and Fang et al. [27]. The performance is assessed in terms of AUC, NSS, CC and KL, averaged over all videos of our, SFU, and DIEM databases.

F. Effectiveness of Single Features and Learning Algorithm

It is interesting to investigate the effectiveness of each HEVC feature in our method. We utilized each single feature of our method to detect saliency of all 33 videos from our database. Since the learning process is not required when evaluating each feature of our method, all 33 videos of our database were tested here without any cross validation. In Table V, we tabulate the saliency detection accuracy of each single feature, measured by AUC, NSS, CC, KL, and EER. This table shows that the AUC results of all 9 HEVC features in our method are significantly better than that of random hit, the AUC of which is 0.5. This confirms that the HEVC encoder can be utilized as an effective feature extractor for saliency detection. Besides, it can be clearly observed from this table that the accuracy of bit allocation related features ranks the highest among all features. Therefore, we can conclude that the bit allocation of HEVC is rather effective in saliency detection, compared to other HEVC features.

Furthermore, Figure 15 evaluates the robustness of each single feature across various working conditions (HM+LD,

TABLE V

MEAN (STANDARD DEVIATION) VALUES FOR SALIENCY DETECTION ACCURACY BY EACH SINGLE FEATURE OF OUR METHOD, AVERAGED OVER THE FRAMES OF ALL 33 TEST VIDEOS

	Basic features			Temporal difference features			Spatial difference features		
	Splitting depth	Bit allocation	MV	Splitting depth	Bit allocation	MV	Splitting depth	Bit allocation	MV
AUC	0.729(0.101)	0.761(0.093)	0.683(0.107)	0.717(0.093)	0.746(0.090)	0.691(0.102)	0.713(0.102)	0.790(0.081)	0.694(0.102)
NSS	0.840(0.487)	1.262(0.717)	0.848(0.665)	0.968(0.552)	1.152(0.630)	0.860(0.605)	0.818(0.495)	1.379(0.701)	0.783(0.621)
CC	0.226(0.122)	0.306(0.154)	0.194(0.154)	0.233(0.123)	0.273(0.139)	0.202(0.147)	0.231(0.127)	0.351(0.148)	0.188(0.147)
KL	0.185(0.086)	0.243(0.103)	0.192(0.086)	0.202(0.077)	0.236(0.089)	0.191(0.078)	0.188(0.076)	0.267(0.091)	0.195(0.085)
EER	0.268(0.079)	0.290(0.085)	0.350(0.094)	0.325(0.083)	0.303(0.082)	0.342(0.090)	0.331(0.092)	0.269(0.091)	0.347(0.092)

TABLE VI

THE AVERAGED ACCURACY OF SALIENCY DETECTION BY OUR METHOD WITH C-SVC AND EQUAL WEIGHT

	AUC	NSS	CC	KL	EER
C-SVC	0.823(0.071)	1.658(0.591)	0.438(0.133)	0.300(0.086)	0.241(0.075)
Equal weight	0.775(0.087)	1.268(0.546)	0.330(0.129)	0.247(0.083)	0.279(0.084)

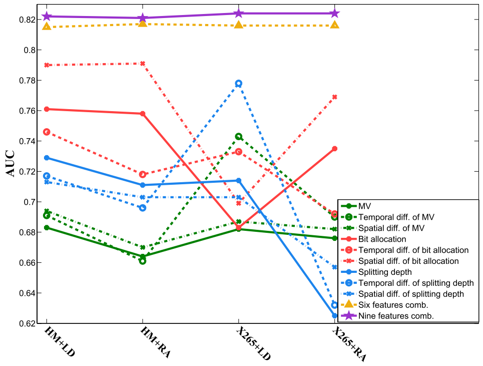


Fig. 15. AUC curves of saliency detection by each single feature and feature combination. Six comb. and nine comb. mean the results of saliency detection by 6 features (excluding features of splitting depth) and by all 9 features, respectively. Similar results can be found for other metrics, e.g. CC.

HM+RA, x265+LD and x265+RA). Here, the evaluation is performed on AUC averaged all 33 videos of our database. We can see that the AUC of each single feature, especially the features of splitting depth, varies at different working conditions. This implies that each single feature relies on the working conditions. Benefitting from the machine learning power of the C-SVC (presented in Section V), the performance of combining all features is significantly more robust than a single feature as shown in Figure 15. Since the splitting depth is least robust across various working conditions, we plot in Figure 15 the AUC values of integrating 6 features (excluding spitting depth related features). It shows that the integration of 6 features underperforms the integration of all 9 features for all working conditions. Thus, we can validate that the features of spitting depth are able to improve the overall performance of our method at various working conditions.

Finally, it is necessary to verify the effectiveness of the C-SVC learning algorithm in our method, since it bridges the gap between the proposed HEVC features and saliency. Provided that the learning algorithm is not incorporated, equal weighting is a common way for feature integration (e.g., in [10]). Table VI compares saliency detection results of our method with the C-SVC learning algorithm and with equal weighting. As can be seen in this table, the C-SVC produces significantly better results in all metrics, compared with the equal weight integration. This indicates the effectiveness of the learning algorithm applied in our method for saliency detection.

VII. CONCLUSION AND FUTURE WORK

In this paper, we found out that the state-of-the-art HEVC encoder is not only efficient in video coding, but also effective in providing the useful features in saliency detection. Therefore, this paper has proposed a novel method for learning to detect video saliency with several HEVC features. Specifically, to facilitate the study on video saliency detection, we first established an eye tracking database on viewing 33 uncompressed videos from test sets commonly used for HEVC evaluation. The statistical analysis on our database revealed that human fixations tend to fall into the regions with the high-valued HEVC features of splitting depth, bit allocation, and MV. Besides, three observations were also found from our eye tracking database. According to the analysis and observations, we proposed to extract and then compute several HEVC features, on the basis of splitting depth, bit allocation, and MV. Next, we developed the C-SVC, as a non-linear SVM classifier, to learn the model of video saliency with regard to the proposed HEVC features. Finally, the experimental results verified that our method outperforms other state-of-the-art saliency detection methods, in terms of ROC, EER, AUC, CC, NSS, and KL metrics.

In reality, almost all videos exist in the form of bitstreams, generated by video coding techniques. Since HEVC is the latest video coding standard, there is no doubt that the HEVC bitstreams will be prevalent in the near future. Accordingly, our method, performed in HEVC domain, is more practicable over other state-of-the-art uncompressed domain methods, as both time and storage complexity on decompressing videos can be saved.

There exist three directions for the future work. (1) Our work in its present form merely concentrates on the bottom-up model to predict video saliency. In fact, videos usually contain some top-down cues indicating salient regions, such as human faces. Indeed, an ideal vision system, like the one of humans, requires the information flow in both directions of

TABLE VII

THE EVALUATION RESULTS OF CAMERA MOTION ESTIMATION FOR VIDEOS OF FIGURE 7 (FROM LEFT TO RIGHT)

Videos	First	Second	Third	Fourth	Fifth	Sixth	Overall
Precision	100%	100%	100%	97.4%	96.3%	100%	98.8%
Recall	100%	100%	89.5%	97.4%	100%	78.9%	96.0%
Distance error E_d	4.20%	13.87%	15.05%	10.41%	3.41%	2.02%	8.61%
Angle error E_a (°)	0.02	0.00	0.25	2.34	1.15	2.49	1.04

bottom-up and top-down. Hence, the protocol, integrating the top-down model into our bottom-up saliency detection method, shows a promising trend in future. (2) Many advanced tracking filters (e.g., Kalman filter and particle filter) have emerged during the past few decades. It is quite an interesting future work to incorporate our method with those filters, rather than the forward smoothing filter of this paper. In that case, the performance of our method may be further improved. (3) A simple SVM learning algorithm, the C-SVC, was developed in our work for video saliency detection. Other state-of-the-art machine learning techniques may be applied to improve the accuracy of saliency detection, and it can be seen as another promising future work.

APPENDIX
EVALUATION ON CAMERA MOTION ESTIMATION

Annotation: In this appendix, we evaluate the accuracy of camera motion estimated by our voting algorithm. Before the evaluation, the ground truth of camera motion needs to be obtained. Since it is intractable to obtain the camera motion data recorded during video shooting, we manually annotated camera motion to approximate the ground truth. Due to space limitation, we only show annotation results of seven videos in Figure 7. Among them, six videos were randomly selected from those containing frames with camera motions in our database. Besides, one video was randomly selected from other videos in our database, which is without any camera motion. Next, every 10 frames[5] of those six videos with camera motion were labeled by hand with 0 for static camera and 1 for moving camera. Among those frames labeled as moving camera, we further annotated the vectors of camera motion to be the ground truth. The annotation of camera motion vectors is conducted by manually finding 5 matched pairs of key points at background (the motion of which is only caused by camera), across two frames. At last, the motions between matched points were measured for each pair, and then averaged over 5 pairs as the annotated camera motion vector between those two frames. The annotated ground truth of camera motion is also available in our website, along with our database.

Evaluation: Given the above annotation, the accuracy of camera motion detected by our voting algorithm is justified from both subjective and objective aspects. The subjective results of Figure 7 illustrate that our algorithm is able to estimate the ground truth of camera motion with high accuracy. We further quantify the accuracy of camera motion estimation. Table VII reports the measured precision and recall on detecting whether camera is moving or not for the frames of the videos in Figure 7. Note that only some of frames in those videos are with nonzero camera motion. We can see from this table that our algorithm is capable of classifying the frames to be with or without camera motion (98.8% for precision and 96.0% for recall). Additionally, we evaluate camera motion vectors estimated by our algorithm, in terms of both distance and angle errors. Here, distance error E_d and angle error E_a are defined by

$$E_d = \frac{|\sqrt{(m_x)^2 + (m_y)^2} - \sqrt{(\hat{m}_x)^2 + (\hat{m}_y)^2}|}{\sqrt{(m_x)^2 + (m_y)^2}}, \qquad (15)$$

and

$$E_a = |\arctan \frac{m_y}{m_x} - \arctan \frac{\hat{m}_y}{\hat{m}_x}|, \qquad (16)$$

where (m_x, m_y) and $(\hat{m}_x, \hat{m}_y)$ stand for the annotated and estimated camera motion vectors, respectively. We can observe from Table VII that the errors between estimated and annotated camera motion vectors are rather small. To be more specific, the averaged distance error is 8.61%, and the averaged angle error is around 1 degree. In summary, we can draw a conclusion that our voting algorithm is effective in estimating camera motion of videos.

[5] It is because the camera motion between two consecutive frames is too small to be annotated.

REFERENCES

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

[2] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 185–207, Jan. 2013.

[3] N. J. Butko and J. R. Movellan, "Optimal scanning for faster object detection," in *Proc. CVPR*, Jun. 2009, pp. 2751–2758.

[4] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proc. CVPR*, Jun. 2015, pp. 3395–3492.

[5] 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.

[6] 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.

[7] U. Engelke, H. Kaprykowsky, H. Zepernick, and P. Ndjiki-Nya, "Visual attention in quality assessment," *IEEE Signal Process. Mag.*, vol. 28, no. 6, pp. 50–59, Nov. 2011.

[8] H. Hadizadeh and I. V. Bajic, "Saliency-aware video compression," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 19–33, Jan. 2014.

[9] M. Xu, X. Deng, S. Li, and Z. Wang, "Region-of-interest based conversational HEVC coding with hierarchical perception model of face," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 3, pp. 475–489, Jun. 2014.

[10] 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.

[11] L. Itti, N. Dhavale, and F. Pighin, "Realistic avatar eye and head animation using a neurobiological model of visual attention," *Opt. Sci. Technol.*, vol. 64, pp. 64–78, Jan. 2004.

[12] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. NIPS*, 2006, pp. 545–552.

[13] R. J. Peters and L. Itti, “Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention,” in *Proc. CVPR*, Jun. 2007, pp. 1–8.

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

[15] L. Zhang, M. H. Tong, and G. W. Cottrell, “Sunday: Saliency using natural statistics for dynamic analysis of scenes,” in *Proc. Annu. Cognit. Sci. Conf.*, 2009, pp. 2944–2949.

[16] 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.

[17] Z. Ren, S. Gao, L.-T. Chia, and D. Rajan, “Regularized feature reconstruction for spatio-temporal saliency detection,” *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3120–3132, Aug. 2013.

[18] Y. Lin, Y. Y. Tang, B. Fang, Z. Shang, Y. Huang, and S. Wang, “A visual-attention model using earth mover’s distance-based saliency measurement and nonlinear feature combination,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 314–328, Feb. 2013.

[19] T. Judd, K. Ehinger, F. Durand, and A. Torralba, “Learning to predict where humans look,” in *Proc. ICCV*, Sep./Oct. 2009, pp. 2106–2113.

[20] W. Kienzle, B. Schölkopf, F. A. Wichmann, and M. O. Franz, “How to find interesting locations in video: A spatiotemporal interest point detector learned from human eye movements,” in *Proc. Joint Pattern Recognit. Symp.*, vol. 4713. 2007, pp. 405–414.

[21] J. Li, Y. Tian, T. Huang, and W. Gao, “Probabilistic multi-task learning for visual saliency estimation in video,” *Int. J. Comput. Vis.*, vol. 90, no. 2, pp. 150–165, 2010.

[22] S. Mathe and C. Sminchisescu, “Dynamic eye movement datasets and learnt saliency models for visual action recognition,” in *Proc. ECCV*, 2012, pp. 842–856.

[23] D. Rudoy, D. B. Goldman, E. Shechtman, and L. Zelnik-Manor, “Learning video saliency from human gaze using candidate selection,” in *Proc. CVPR*, Jun. 2013, pp. 1147–1154.

[24] S.-H. Lee, J.-H. Kim, K. P. Choi, J.-Y. Sim, and C.-S. Kim, “Video saliency detection based on spatiotemporal feature learning,” in *Proc. ICIP*, Oct. 2014, pp. 1120–1124.

[25] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[26] K. Muthuswamy and D. Rajan, “Salient motion detection in compressed domain,” *IEEE Signal Process. Lett.*, vol. 20, no. 10, pp. 996–999, Oct. 2013.

[27] 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.

[28] S. Hossein Khatoonabadi, N. Vasconcelos, I. V. Bajic, and Y. Shan, “How many bits does it take for a stimulus to be salient?” in *Proc. CVPR*, Jun. 2015, pp. 5501–5510.

[29] G. J. Sullivan and R. Baker, “Efficient quadtree coding of images and video,” *IEEE Trans. Image Process.*, vol. 3, no. 3, pp. 327–331, May 1994.

[30] B. Li, H. Li, L. Li, and J. Zhang, “Domain rate control algorithm for high efficiency video coding,” *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 3841–3854, Sep. 2014.

[31] T. Shanableh, “Saliency detection in MPEG and HEVC video using intra-frame and inter-frame distances,” *Signal, Image Video Process.*, vol. 10, no. 4, pp. 703–709, 2016.

[32] D. Pang, A. Kimura, T. Takeuchi, J. Yamato, and K. Kashino, “A stochastic model of selective visual attention with a dynamic bayesian network,” in *Proc. ICME*, Apr. 2008, pp. 1073–1076.

[33] B. Wu and L. Xu, “Integrating bottom-up and top-down visual stimulus for saliency detection in news video,” *Multimedia Tools Appl.*, vol. 73, no. 3, pp. 1053–1075, Dec. 2014.

[34] A. Borji, M. N. Ahmadabadi, and B. N. Araabi, “Cost-sensitive learning of top-down modulation for attentional control,” *Mach. Vis. Appl.*, vol. 22, no. 1, pp. 61–76, 2011.

[35] A. Borji, D. N. Sihite, and L. Itti, “What/where to look next? Modeling top-down visual attention in complex interactive environments,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 5, pp. 523–538, May 2014.

[36] Y. Hua, Z. Zhao, H. Tian, X. Guo, and A. Cai, “A probabilistic saliency model with memory-guided top-down cues for free-viewing,” in *Proc. ICME*, Jul. 2013, pp. 1–6.

[37] M. Marszalek, I. Laptev, and C. Schmid, “Actions in context,” in *Proc. CVPR*, Jun. 2009, pp. 2929–2936.

[38] M. D. Rodriguez, J. Ahmed, and M. Shah, “Action mach a spatio-temporal maximum average correlation height filter for action recognition,” in *Proc. CVPR*, Jun. 2008, pp. 1–8.

[39] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, “Comparison of the coding efficiency of video coding standards—Including high efficiency video coding (HEVC),” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.

[40] O. L. Meur, A. Ninassi, P. L. Callet, and D. Barba, “Do video coding impairments disturb the visual attention deployment?” *Signal Process., Image Commun.*, vol. 25, no. 8, pp. 597–609, 2010.

[41] 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.

[42] P. K. Mital, T. J. Smith, R. L. Hill, and J. M. Henderson, “Clustering of gaze during dynamic scene viewing is predicted by motion,” *Cognit. Comput.*, vol. 3, no. 1, pp. 5–24, Mar. 2011.

[43] L. Itti, “Automatic foveation for video compression using a neurobiological model of visual attention,” *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1304–1318, Oct. 2004.

[44] C. C. Chang and C. J. Lin, “LIBSVM: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

[45] JCT-VC. *HM 16.0*, accessed on Jan. 2015. [Online]. Available: http://hevc.hhi.fraunhofer.de

[46] L. Duan, C. Wu, J. Miao, L. Qing, and Y. Fu, “Visual saliency detection by spatially weighted dissimilarity,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 473–480.

Mai Xu (M’10) received the B.S. degree from Beihang University in 2003, the M.S. degree from Tsinghua University in 2006, and the Ph.D. degree from the Imperial College London in 2010. From 2010 to 2012, he was a Research Fellow with the Electrical Engineering Department, Tsinghua University. Since 2013, he has been with Beihang University as an Associate Professor. From 2014 to 2015, he was a Visiting Researcher of MSRA. He has authored over 50 technical papers in international journals and conference proceedings. His research interests mainly include visual communication and image processing. He was a recipient of best paper awards of two IEEE conferences.

Lai Jiang received the bachelor’s degree in Beihang University, Beijing, China, in 2015, where he is currently pursuing the Ph.D. degree. His research interests include saliency prediction and video analysis. He was a recipient of the Outstanding Graduate Student of Beijing and the Beihang New Ph.D. Scholarship during his undergraduate and Ph.D. life, respectively.

Xiaoyan Sun (M’04–SM’10) received the B.S., M.S., and Ph.D. degrees in computer science from Harbin Institute of Technology, Harbin, China, in 1997, 1999, and 2003, respectively. She has been an intern in Microsoft Research Asia since 2000 and joined Microsoft Research Asia later in 2003, where she is currently a Lead Researcher with Internet Media Group. She has authored or co-authored more than 90 journal and conference papers and ten proposals to standards, including one accepted by H.264. Her current research interests include image and video compression, image processing, computer vision, and cloud computing. Dr. Sun was a recipient of the Best Paper Award of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 2009. Dr. Sun served as the Session Chair, the Area Chair, and the TC Co-Chair of several international conferences. She also serves as a TC Member of the IEEE *Multimedia Systems & Applications* and the AE of SIGNAL PROCESSING: IMAGE COMMUNICATION.

Zhaoting Ye received the B.S. degree in electrical and computer engineering from Beihang University in 2012. He is currently pursuing the master's degree in CMU, with a focus on computer vision under Robotics Institute. From 2011 to 2012, he was as a Research Assistant with Prof. Xu in Beihang University. His research area mainly includes computer vision and deep learning.

Zulin Wang (M'14) received the B.S. and M.S. degrees in electronic engineering and the Ph.D. degree from Beihang University, Beijing, China, in 1986, 1989, and 2000, respectively. He is currently the Dean of the School of Electronic and Information Engineering, Beihang University. His research interests include image processing, electromagnetic countermeasure, and satellite communication technology. He has authored or co-authored over 100 papers and published two books in these fields. He holds six patents. He has undertaken approximately 30 projects related to image/video coding and image processing.