Shanableh, T. (2016). Saliency detection in MPEG and HEVC video using intra-frame and inter-frame distances. Signal, Image and Video Processing, 10(4), 703-709.
The final publication is available at Springer via http://dx.doi.org/ DOI: 10.1007/s11760-015-0798-9

Saliency Detection in MPEG and HEVC Video using Intra-frame and Inter-frame Distances

Tamer Shanableh
Department of Computer Science and Engineering
College of Engineering
American University of Sharjah
Fax: +971 6 515-2979
tshanableh@aus.edu

Abstract

This paper proposes a video saliency detection model for MPEG and HEVC coded videos. The model extracts features from MPEG macro blocks and HEVC coding units. The feature variables are based on syntax elements and statistics of prediction error. The suitability of the selected features is verified through the use of stepwise regression. Three saliency maps are generated based on intra-frame distances, inter-frame distances and global distances. The proposed model is tested using the eye-1 dataset compiled by Laurent Itti lab in the University of Southern California. The accuracy of the model is quantified by comparing saliency values at human saccade locations against saliency values at random locations. The comparison is performed in terms of Kullback–Leibler distances and receiver-operator curves. The proposed solution is compared against existing work using similar experimental setup. Experimental results revealed that a Kullback–Leibler distance of 2.14 and area under the receiver-operator curve of 0.936 are achieved.

Keywords: Saliency detection, feature extraction, MPEG, HEVC

1. Introduction

Video objects or regions that stand out relative to their neighbors attract the attention of the viewer. Such objects or regions can be automatically detected through a process known as video saliency detection. Various solutions are proposed in the literature for video saliency detection which are mainly based on the concept of center-surround differences in both the spatial and temporal domains.

Video saliency detection has a number of important video processing applications including; object segmentation and extraction, object identification, Region of Interest (ROI) detection, video surveillance, compression, quality assessment and error concealment.

Spatio-temporal video information is used for saliency detection. The work in [1] proposed to model temporal saliency using motion trajectories and video reconstruction error. Spatial saliency on the other hand can be captured by detecting regions with high center-surround contrast. The temporal saliency and spatial saliency can be fused to emphasize salient regions with high confidence.

Regions with high local contrast, global rare spatial or temporal features can be used for saliency detection as well [2]. The high contrast is not restricted to the spatial domain, it can also be extended to the temporal domain [3]. Estimating the orientation contrast using spatio-temporal directional coherence is also applicable to saliency detection as reported in [4]. Additionally, spatial and temporal features are not restricted to the pixel domain, for instance, the phase spectrum of Fourier transform is used to detect spatial saliency. Similarly, phase spectrum of Fourier transform can be used to obtain the temporal saliency map of each video frame using motion vector information [5]. Likewise, local center-surround differences and global contrast can also be computed using wavelet-domain features [6].

Machine learning and prediction are also used in saliency detection where features of video regions that are of visual interest are leant by a classifier and the resulting model can be used for classification [7]. It was also shown that a saliency map can be predicted taking into account the maps of previous video frames [8].

Once saliency is detected in video frames, it can be used in a number of applications. For instance, a higher subjective quality can be achieved by spending more bits on salient regions in the application of video compression [9]. More specifically, a saliency value can be mapped to a HEVC quantization parameter to be used by a video encoder [10]. Additionally, it was found that the human visual system is more sensitive to distortions in salient video regions [11]. This led to the use of saliency detection in no-reference quality assessment of compressed video [12]. It also led to applying more error protection to salient regions in video error resiliency

applications [13]. Other applications of saliency detection include assessing blurring artifact in video frames [14], predicting eye positions [15] and obtaining superpixels in images [16].

Another rather obvious application of saliency detection is object extraction and detection. For example, foreground objects of interest can be automatically extracted using saliency information extracted from the input video [17]. Location of targets of interest can also be detected [18]. Lastly, the authors in [19] proposed a video saliency detection model based on the concept of center-surround feature differences in the compressed domain. Block-based features are extracted from motion information, luminance, color and texture. Two saliency maps are calculated based on the underlying frame type; static maps for I-frames and motion maps for predicted frames. A novel method of parameterized normalization, sum and product fusion of the static and motion saliency maps was proposed. Experimental results showed superior accuracy of video saliency detection.

In this work we propose a video saliency detection model based on spatio-temporal center-surround differences using intra and inter frame distances. We extract representative feature vectors using syntax elements and texture information of prediction error extracted from coded video. We use stepwise regression to validate the suitability of the selected features for saliency detection. Three saliency maps are generated based on intra-frame distances, inter-frame distances and global distances. The proposed solutions are applied to both MPEG1 macro blocks and HEVC Coding Units (CUs).

The main differences between the proposed work and that reported in [19] is as follows. In this work, we apply the concept of center-surround differences to compute feature vectors based on spatial, temporal and global differences. We also use stepwise regression to verify the suitability of the feature variables which are extracted from MPEG1 MBs and HEVC CUs. Consequently, the computed saliency maps are fused using a novel minimum entropy function.

The paper is organized as follows. The overview of the proposed system is presented in Section 2. The proposed feature extraction solution is presented in Section 3. The computation of saliency maps is presented in Section 4 and the experimental results are discussed in Section 5.

2. System overview

We start by presenting the overall system architecture for computing the saliency maps. The first step is to extract features for each and every block in the coded video. In MPEG video, we extract features from Macro Blocks (MB), whereas in HEVC video, features are extracted from Coding Units (CUs). The details of feature extraction are presented in the next section. We propose to extract features from syntax

elements of the video stream, prediction error and DCT coefficients for inter and intra coded blocks. Once the feature vectors are extracted, the computation of the saliency maps commences.

This work uses the concept of center-surround differences in the computation of saliency maps [19]. The basic idea is to compute the distances between a feature vector representing a MB/CU and its surrounding feature vectors. We refer to this as an intra-frame distance. We also compute the distances between a feature vector representing a MB/CU with co-located feature vectors from previous frames. We refer to this as an inter-frame distance. Lastly, to capture global features, the distance between each feature vector and the mean feature vector in a given video frame is computed, this is referred to as global distance. Each of the mentioned distances results in one saliency map as illustrated in Figure 1.

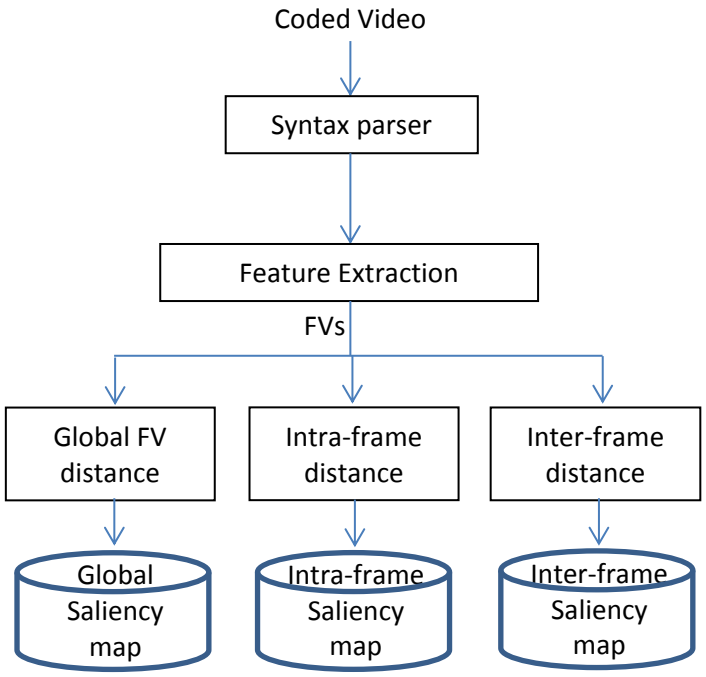

Figure 1. Overall system overview for computing saliency maps from coded videos.

The three saliency maps can then be used individually or fused into one as explained in the Section 3.

3. Feature extraction and stepwise regression

The feature variables representing MPEG MBs are based on syntax elements and statistics from DCT coefficients and prediction error. The variables are listed in Table 1.

Table 1. Description of feature variables extracted from MPEG MBs.

	ID	Feature Description
Syntax level	1	The number of bits needed to code a MB.
	2	The coding type of the MB.
	3	The magnitude of the MB's MVs.
	4	The phase of the MB's MVs.
	5	The average difference between the magnitude of the current MV and the surrounding ones.
	6	The average difference between the phase of the current MV and the surrounding ones.
	7	MB coded block pattern showing skipped and coded blocks.
Prediction error level	8	The absolute sum of high DCT frequencies of a MB.
	9	Texture's mean.
	10	Texture's standard deviation.
	11	Texture's smoothness.
	12	Texture's 3rd moment as an indication of histogram skewness.
	13	Texture's uniformity.
	14	Texture's entropy.

Such feature extraction level does not require full video decoding as motion compensation is not required. In the table, feature ID 2 indicates the MB type. If the type is intra then all MV information is represented by zeros. If the type indicates forward prediction then all backward MV information is represented by zeros and vice a versa. And if the type is bidirectional then the information is available for both forward and backward motion vectors.

The texture's smoothness for feature ID 11 is defined as:

$$s_i = 1 - 1/\left(1 + \sigma_i^2\right) \qquad (1)$$

Where s_i is the smoothness of MB index *i* and σ_i is its texture standard deviation.

The texture's 3rd moment for feature ID 12 is defined as:

$$m_i = \sum_{n=0}^{N-1}(p_n - \bar{p})^3 f(p_n) \qquad (2)$$

Where m_i is the third moment of MB index *i*, N is the total number of pixels in a MB, p_n is a pixel value at index *n*. $\bar{p}$ is the mean pixel value and *f(.)* is the relative frequency of a given pixel value.

The texture's uniformity for feature ID 13 is defined as:

$$u_i = \sum_{n=0}^{N-1} f^2(p_n) \qquad (3)$$

Where u_i is the uniformity of MB index *i*. The rest of the variables and the function are defined in Equation (2). Lastly, the texture's entropy for feature index 14 is defined as:

$$e_i = -\sum_{n=0}^{N-1} f(p_n) log_2 f(p_n) \quad (4)$$

Where e_i is the entropy of MB index *i*. The rest of the variables and functions are defined in Equation (2). Once the MB features are extracted, the feature vectors are normalized to the same range. The normalization is applied to each feature separately using z-scores which is defined as:

$$z_i = (x_i - E(\mathbf{x}))/\sigma_{\mathbf{x}} \quad (5)$$

Where the scalars $z_i$and x_i are the normalized and non-normalized feature values of feature index *i* respectively. *E(x)* is the expected value of the feature variable and σ_x is its standard deviation. Both are computed based on the feature vector population.

The feature variables for HEVC coding units are similar, however since the HEVC syntax is more sophisticated than MPEG, we start with a brief review of the coding units concept. A video frame is divided into square blocks known as coding units (CUs). The maximum allowed size is 64 × 64 for the luma component and the minimum size is 8x8. The syntax of each CU indicates the type of prediction, the Transform Unit (TU) sizes and the types of the Prediction Units (PU) used. The syntax also defines if a CU is coded in split mode. The largest CU has a depth of 0 and if it is further split then the four resultant CUs have a depth of 1, and so forth. The partitioning used for motion estimation and compensation is carried out according to the size of the PUs. In this work, we are interested in extracting features from CUs and their PUs by setting the size of CUs to 16x16 pixels. Further details about HEVC can be found in [20].

The feature variables representing HEVC CUs are based on syntax elements and statistics from DCT coefficients and prediction error. The variables are listed in Table 2. Since each CU can be partitioned into many PUs, then the corresponding feature variables are averaged for all PUs in a given CU. Clearly, if the CU has a depth of zero then an averaging is not required.

Table 2. Description of feature variables extracted from HEVC CUs.

ID		Feature description
1	CU features	Total number of bits in a CU
2		X coordinate of a CU in pixels
3		Y coordinate of a CU in pixels
4		Total number of partitions in a CU
5	PU features	Coding depth
6		Partition type (2Nx2N, 2NxN,…)
7		Partition width
8		Partition height
9		Coding mode

10		Transformation index
11		Merge Flag
12		Merge Index
13		Inter prediction direction
14		Coded block flag
15		Magnitude of MV
16		Phase of MV
17		Magnitude of difference MV
18		Phase of difference MV
19		Variance of prediction error
20		Mean of prediction error
21		Skewness of prediction error

To verify the suitability of the feature variables to the task of saliency detection, we use the stepwise regression procedure. To use this procedure we treat the feature vectors as predictors. The response variable in this case is the existence or lack of a saccade at a given MB location. The feature vectors are extracted from the videos of the "eye-1" dataset by *Itti et. al*. as elaborated upon in the experimental results section. The saccade locations are available from the same dataset as well. The process of identifying the important feature variables is illustrated in Figure 2.

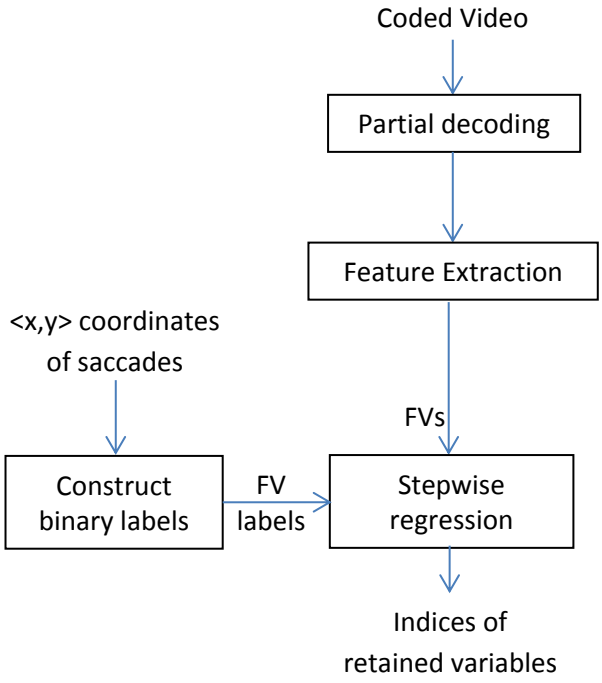

Figure 2. Use of stepwise regression for feature selection.

Stepwise regression is an objective procedure used for selecting important feature variables. Again, we treat the feature variables $x_1, x_2, \ldots, x_k$ as predictors where k is the number of features in each feature

vector. The response variable, y represents the existence or lack of a saccade at a given MB/CU location. In [21], the stepwise regression procedure is described using the following steps. In the first step, the procedure tests all possible one-predictor regression models in an attempt to find the predictor that has the highest correlation with the response variable. The model is of the form:

$$\hat{y} = \beta_0 + \beta_1 x_i \qquad (6)$$

A hypothesis test is conducted for each model where H_0: $\beta_1 = 0$ and H_1: $\beta_1 \neq 0$. The test is conducted using the well-known *T* test at a specific level of significance, say α = 0.1. The predictor that generates the largest absolute *T* value is selected. Refer to this predictor as x_1.

In the second step, the remaining *k-1* predictors are scanned for the best two-predictor regression model of the form:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_i \quad (7)$$

This is achieved by testing all two-predictor models containing x_1 which was selected from the first step. The *T* value of the *k-1* models are computed for H_0: $\beta_2 = 0$. The predictor that generates the highest absolute *T* value is retained, Refer to this predictor as x_2.

Now that $\beta_2 x_2$ is added to the model, the procedure goes back and reexamines the suitability of including β_1 in the model. If the corresponding *T* value becomes insignificant (i.e. the alternative hypothesis H_1 is rejected.), x_1 is removed and the predictors are searched for a variable that generates the highest *T* value in the presence of $\beta_2 x_2$. In the third step, remaining *k-2* predictors are scanned for the best three-predictor regression model of the form:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_{2+} \beta_3 x_i \qquad (8)$$

The procedure repeats until no further predictors are added/removed from the model. Stepwise regression was successfully used for selecting feature variables for detecting the existence of delated frames in video forensics as reported in [22].

Having applied the stepwise regression procedure to the MPEG features of Table 1, it was found that feature ID 6 does not correlate with the response variable; “The average difference between the phase of the current MV and the surrounding ones”. Note that the phase of the MV is already present in feature ID 4. The exclusion of variable 6 can be attributed to the fact that the magnitude of the motion might attract the human attention more than its phase. Likewise, feature ID 6 does not correlate with the response variable as well. Lastly, when applying the procedure to the HEVC features of Table 2, it was found that the partition type and height are not required (Feature IDs 6 and 8). This can be justified by the use of other feature variables that might give similar information such as feature ID 5,7 and 9.

Interestingly, the phase of the MVs was not selected just like the case for MPEG MB features. The rest of the variables are selected which indicates the suitability of the selected variables for saliency detection.

4. Computation of saliency maps

The basic idea is to compute the distances between a feature vector of a MB and its surrounding feature vectors. We refer to this as an intra-frame distance approach. In this work we extend this approach to compute the difference between a feature vector of a MB with its co-located feature vectors from previous frames. We refer to this as an inter-frame distances approach. Global features are captured by computing the distance between each feature vector and the mean feature vector in a given video frame, this is referred to as global distance. Each distance results in a separate saliency map which can be used individually or fused to generate one saliency map.

The intra-frame distances are illustrated in Figure 3 part 'a'. The idea is to compute the Euclidean distance between the feature vector of the center MB and its surroundings at level 1 (L=1). The summation of distances is then multiplied by a given weight. The process is repeated for level 2 (L=2) up to level *l*. The assigned weight is inversely proportional to the distance from the center MB.

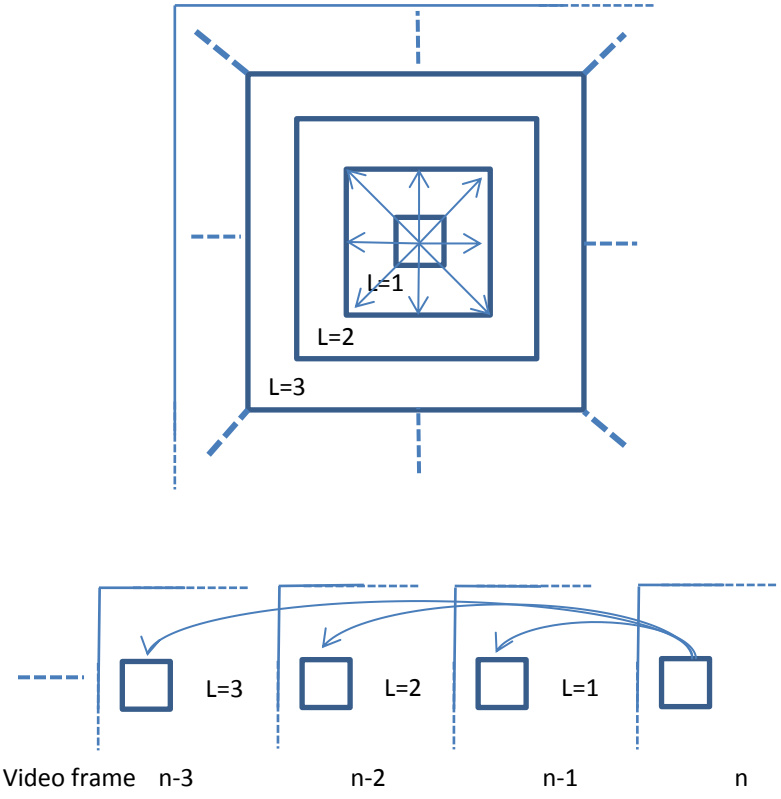

Figure 3. Illustration of spatio-temporal center-surround differences. (a) Top figure, illustrates intra-frame difference and (b) Bottom figure, illustrates inter-frame differences.

Hence the saliency map of intra-frame distances for a given MB can be computed as follows:

$$S_{Intra}^{i} = \sum_{L} \boldsymbol{w}_{L} * \sum_{j} \left\| FV_{i} - FV_{L,j} \right\|_{2} \qquad (9)$$

Where S_{Intra}^{i} denotes the saliency value for MB *i* using intra-frame distance. The summation variable *L* denotes the level of surrounding MBs as illustrated in Figure 3-a. In this work, 4 levels are used for intra-frame distances and 15 for inter-frame distances. The latter corresponds to only half a second of video as the temporal resolution is around 30 frames per second. FV_{L} is the set of FVs belonging to the surrounding MBs at Level *L*. Each surrounding level has a weight associated with it which decays as the level increases. In this work, the following set of weights is used $\boldsymbol{w}$ = {1, 1/2, 1/3,…,1/maxLevel}.

Likewise, the inter-frame distances are illustrated in Figure 3-b. The idea is to compute the Euclidean distance between the feature vector of the current MB and its co-located feature vector at level 1 in the previous video frame. The distance is then multiplied by a given weight. The process is repeated using the previous frame at level 2 up to level *l*. Again, the assigned weight is inversely proportional to the distance from the current MB. The saliency map of inter-frame distances for a given MB can be computed as follows:

$$S_{Inter}^{i} = \sum_{L} \boldsymbol{w}_{L} * \| FV_{i} - FV_{L} \|_{2} \qquad (10)$$

Where S_{Inter}^{i} denotes the saliency value for MB *i* using inter-frame distance. Note here that in each temporal level there is only one feature vector belonging to level *L* as illustrated in Figure 3-b. This is denoted as FV_{L} in Equation 10.

Once the saliency maps are computed, they can be individually used for saliency detection or fused into one map prior to detection. This work uses a number of techniques for fusion, such techniques are similar to the fusion techniques summarized in [23]. We start by computing a binary mask by thresholding each saliency map using its mean value, this results in three masks, $B_{j}, j \in \{inter, intra, global\}$. The saliency maps are thresholded using the corresponding masks prior to using them for saliency detection. If used without fusion then the saliency map is represented by $N(S_{j} \cap B_{j})$, where N is a normalization operator and S_{j} is a saliency map. On the other hand, if fusion is used then a number of methods are employed including normalize and sum (NSum), represented by, $\sum_{j} N(S_{j} \cap B_{j})$, normalize and multiply (NProd), represented by, $\prod_{j} N(S_{j} \cap B_{j})$ and normalize and maximum (NMax), represented by $\max_{j} N(S_{j} \cap B_{j})$. Additionally, we experiment with a minimum entropy function which is defined as:

$$S = \sum_{j} \alpha_{j} N(S_{j} \cap B_{j}) + (\alpha_{1} + \alpha_{2} + \alpha_{3})/3 * \prod_{j} N(S_{j} \cap B_{j}) \qquad (11)$$

Where $\alpha_{j}$is the reciprocal of the entropy of a thresholded saliency map.

5. Experimental results

In the following experimental results we used of the "eye-1" dataset which is an eye-tracking dataset collected from eight distinct subjects watching complex video stimuli. The dataset is contributed by the University of Southern California, Laurent Itti lab. The dataset is freely available through the Collaborative Research in Computational Neuroscience (CRCNS), data sharing website [24]. In addition to the videos and eye-tracking dataset, the download contains useful Perl and Matlab scripts that can be used for parsing, computing statistics and displaying results.

The dataset contains eye movement recordings from eight subjects watching 50 video clips. The 50 videos are consisted of around 46,000 video frames each with a spatial resolution of 480x640 pixels. The video are compressed using MPEG1 with variable bitrate coding using a quantization step size triplet of {8, 10, 25} for I,P and B frames respectively. The GoP structure is N=15, M=3, that is; IBBPBBPBBPBBPBB. The frame rate is 30.13 Hz. The saccade data is collected from eight subjects. The capture rate of the infrared video-based eye tracker used in the experiment is 240 samples/s.

In the experimental process, we coded the video using both MPEG1 and HEVC with a GoP size of 15 using one I-frame and 14 P-frames. In the latter codec, we sat the CU size to 16x16 as mentioned previously.

Our experimental setup is based on the work proposed by [25] and [26] and later adopted by [19]. We compared the saliency values at saccade locations to 100 random locations in the saliency maps. The saliency value at a saccade location is calculated as the maximum within a 64 bit radius. The higher the corresponding saliency value, the more accurate is the saliency detection model. For the 100 random locations, a search is performed around the randomly selected coordinates using the immediate surrounding MBs/CUs to find the maximum saliency value.

A saliency detection model is considered effective if it generates high saliency values at saccade locations. At the same time, the model is expected to generate low saliency values at random locations. In [8],[19] and [27], the saliency distributions at saccade locations and random locations are summarized using a histogram of ten bins where the saliency values on the x-axis are normalized to the range [0-1]. The distance between these two distributions is computed using Kullback–Leibler (KL) distance [26], [30]. The KL is a distance function from a true probability distribution to a target probability distribution. We presented the average KL distance between the histogram of saliency values corresponding to saccade locations(s) and the histogram of saliency values corresponding to random locations. Since we use 100 random locations in each saccade frame, we refer to each histogram as (R_n) where $\in \{1,2,..100\}$. The average KL distance is represented as follows:

$$\overline{KL} = \sum_{n=1}^{100} KL_n(\boldsymbol{s}, R_n) \qquad (12)$$

Where the KL distance between 2 distributions is calculated as follows:

$$KL(\boldsymbol{s}, R_n) = 0.5 * (\sum_{i=1}^{10} s_i \ln \frac{s_i}{R_{n,i}} + \sum_{i=1}^{10} R_{n,i} ln \frac{R_{n,i}}{s_i}) \qquad (13)$$

Where s_i and $R_{n,i}$ are the relative frequency values corresponding to histogram bin *i*. In this work 10 histogram bins are used. The higher the KL distance, the higher is the accuracy of the saliency detection model.

To further assess the accuracy of the proposed saliency detection solutions, the Area Under the ROC Curve (AUC) is reported. We use an interval of 0.1 to compute the ROC curves. At each threshold, the true positive rate is calculated as the number of saccade locations with saliency values larger than the threshold divided by the total number of saccade locations. The false positive rate is calculated as the number of random locations with saliency values larger than the threshold divided by the total number of random locations. The AUC value is calculated 100 times pertaining to 100 random saliency locations. We report the average AUC value.

In Table 3, the KL and AUC are reported for all of the proposed solutions. It is shown that both video coders results in similar saliency detection accuracy in terms of KL and AUC. This gives an indication that the video coder type did not influence the accuracy of the model.

Table 3. KL and AUC of the proposed saliency detection solutions using a search radius of 64 bits

	HEVC		MPEG	
	KL	AUC	KL	AUC
Intra-frame.	2.121	0.938	2.146	0.936
Inter-frame.	1.771	0.912	1.656	0.905
Global	1.819	0.926	1.91	0.915
NProd	1.827	0.918	1.923	0.924
NSum	2.025	0.932	2.028	0.926
Entropy	2.013	0.931	2.003	0.926
NMax	1.812	0.926	1.817	0.919

The results in the table show that the intra-frame distances results in the best accuracy followed by the normalize and sum fusion approach. The inter-frame and global distances are both inferior to the intra-frame solution. These conclusions are valid for both types of video coders in use.

The same set of results are repeated but using a search radius of 32 bits around saccade locations. The results are reported in Table 4.

Table 4. KL and AUC of the proposed saliency detection solutions using a search radius of 32 bits.

	HEVC		MPEG	
	KL	AUC	KL	AUC
Intra diff.	1.12	0.848	1.046	0.844
Inter diff.	0.805	0.81	0.699	0.79
Global	0.886	0.827	0.827	0.815
NProd	0.771	0.802	0.783	0.813
NSum	0.99	0.837	0.951	0.829
Entropy	0.994	0.836	0.951	0.829
NMax	0.85	0.823	0.754	0.809

It is shown that the accuracy of the detection is clearly affected by this reduction is search radius. This is expected as data collected from different subjects participating in eye gaze experiment contain different saccade coordinates for the same saccade video frames. Hence, a larger search around a saccade location is needed and justified.

In Figure 5, we show the distribution of saliency values resulting from both saliency at saccade locations and saliency at 100 random locations. The x-axis represents the normalized saliency values and the y-axis represents the tally of these values. In a good prediction model, the histogram of saliency values corresponding to saccade locations should be more populated in the high value bins. And the histogram of saliency values corresponding to random locations should be more populated in the low value bins. Apart from the intra-frame distance solution, in Table 3 it was shown that the best fusion approach was the normalize and sum. We show the histograms of the best fusion solution alongside the histograms of intra/inter-frame and global distances without fusion. Hence, we generate 3 sets of histograms for each of the proposed feature extraction solutions.

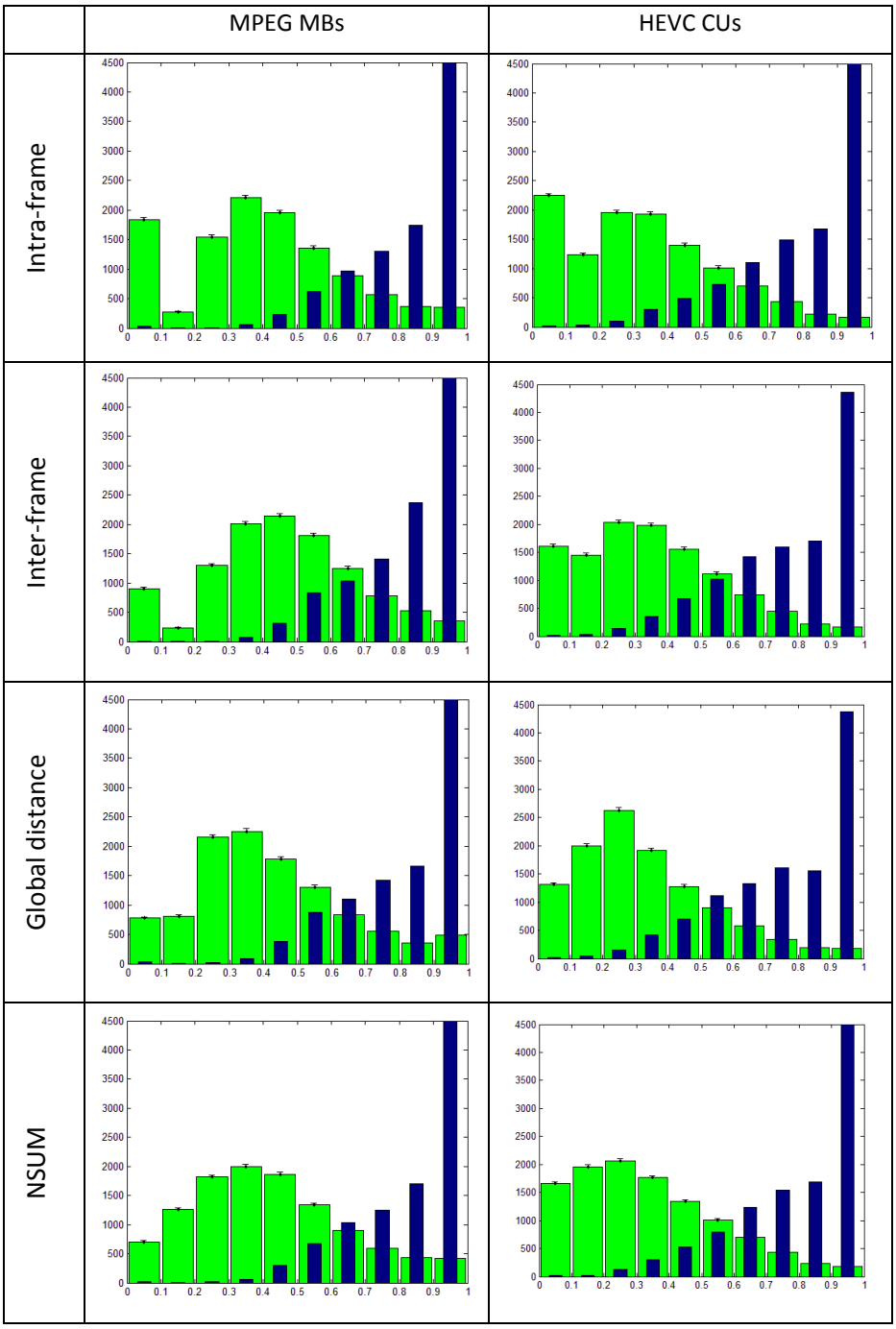

Figure 5. Normalized saliency distributions at saccade locations (represented using narrow bars) and random locations (represented using wide bars).

The histograms are as expected for a saliency detection solution. The counts of low saliency values are high for random solutions. And the counts of high saliency values are high for saccade locations. It is also shown that this statement is also evident when applying the normalize and sum to fuse the three saliency maps.

The proposed solutions are compared against existing work in Table 5. The work in [19] uses spatial center-surround block distances and takes into account both static and motion saliency maps. In [28], the saliency detection model is mainly based on global frame contrast, this model is referred to as MRS. In [25], the saliency detection model is based on the fact that human gaze is attracted towards surprise, this model is referred to as surprise. A new data observation is considered a surprise if the posterior distribution of the data is significantly different from the prior distribution. Lastly, [29] employs static saliency detection salient taking into account that local and global surroundings of saliency regions are distinctive, this model is referred to as CA. The results of existing work are based on what is reported in [19].

It is shown that the proposed solutions result in higher saliency detection accuracy compared to existing work. This indicates that the proposed feature extraction and saliency map computation result in rather accurate saliency detection.

Table 5. Comparison of saliency detection accuracy against existing work.

	MRS [28]	Surprise [25]	CA [29]	Compressed domain [19]	MPEG MBs	HEVC CUs
KL	0.529	0.593	0.76	1.828	2.146	2.121
AUC	0.771	0.782	0.802	0.93	0.936	0.938

Despite its innovative solution, one drawback of the work proposed in [19] is that it computes the static saliency map based on the previous I-frame. Therefore, the same saliency values are used for the whole GoP. Although, the authors showed that such an approach works for a GoP size of up to 24 frames, nonetheless, it will fail if a scene change occurred in a GoP. In the proposed solution however, spatio-temporal center-surround differences are used for each MB to calculate the saliency maps regardless of the GoP's I-frame.

6. Conclusion

The paper proposed a video saliency detection model for MPEG and HEVC coded videos. The model extracted block-based features from MPEG MBs and HEVC CUs. The feature variables are based on syntax elements and statistics of prediction error. The suitability of the selected features was verified through the use of stepwise regression. It was shown that most of the selected features correlate with the existence of a human saccade. Three saliency maps were generated based on intra-frame distances, inter-frame distances and global distances. The proposed model was tested using the eye-1 dataset. The

accuracy of the model was quantified by comparing saliency values at human saccade locations against saliency values at random locations. Experimental results revealed that the intra-frame distance results in the highest saliency detection accuracy. It was also shown that the detection accuracy is very similar for the two video codecs used.

References

[1] Z. Ren, S. Gao, L.-T. Chia and D. Rajan,"Regularized Feature Reconstruction for Spatio-Temporal Saliency Detection," IEEE Transactions on Image Processing, 22(8), August, 2013

[2] M. Decombas, N. Riche, F. Dufaux, B. Pesquet-Popescu, M. Mancas, B. Gosselin and T. Dutoit, "Spatio-temporal saliency based on rare model," IEEE International Conference on Image Processing (ICIP), 15-18 September, 2013

[3] K. Wonjun and J.-J. Han, "Video Saliency Detection Using Contrast of Spatiotemporal Directional Coherence," IEEE Signal Processing Letters, 20(10), October, 2014

[4] W. Kim and C. Kim, "Spatiotemporal Saliency Detection Using Textural Contrast and Its Applications," IEEE Transactions on Circuits and Systems for Video Technology, 24(4), April, 2014

[5] K.-T. Hu, J.-J. Leou and H.-H. Hsiao, "Visual attention region determination for H.264 videos," International Conference on Pattern Recognition (ICPR), 11-15 November, 2012

[6] N. Imamoglu, W. Lin and Y. Fang, "A Saliency Detection Model Using Low-Level Features Based on Wavelet Transform," IEEE Transactions on Multimedia, 15(1), January, 2013

[7] S. Nataraju, V. Balasubramanian and S. Panchanathan, "Learning attention based saliency in videos from human eye movements," Workshop on Motion and Video Computing, December, 2009

[8] D. Rudoy, D.B. Goldman, E. Shechtman and L. Zelnik-Manor, "Learning Video Saliency from Human Gaze Using Candidate Selection," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 23-28 June, 2013

[9] H. Hadizadeh and I.V. Bajic, "Saliency-Aware Video Compression," IEEE Transactions on Image Processing, 23(1), January, 2014

[10] S. Milani, R. Bernardini and R. Rinaldo, "A saliency-based rate control for people detection in video," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013, 26-31 May, 2013

[11] U. Engelke, M. Barkowsky, P. Le Callet and H. Zepernick, "Modelling saliency awareness for objective video quality assessment," Second International Workshop on Quality of Multimedia Experience (QoMEX), 21-23 June, 2010

[12] A. Redl, C. Keimel and K. Diepold, "Saliency based video quality prediction using multi-way data analysis," Fifth International Workshop on Quality of Multimedia Experience (QoMEX), 3-5 July, 2013
[13] H. Hadizadeh, I.V. Bajic and G. Cheung, "Video Error Concealment Using a Computation-Efficient Low Saliency Prior," IEEE Transactions on Multimedia, 15(8), December, 2013
[14] F. Dardi, L. Abate and G. Ramponi, "No-reference measurement of perceptually significant blurriness in video frames," Signal, Image and Video Processing, 5(3), pp. 271-282, 2011
[15] S. Hamel, N. Guyader, D. Pellerin and D. Houzet, "Contribution of color in saliency model for videos," Signal, Image and Video Processing, March, 2015.
[16] L. Xu, L. Zeng and Z. Wang, "Saliency-based superpixels," Signal, Image and Video Processing, 8(1), pp. 181-190, 2013
[17] W.-T. Li, H.-S. Chang, K.-C. Lien, H.-T. Chang and Y. Wang, "Exploring Visual and Motion Saliency for Automatic Video Object Extraction," IEEE Transactions on Image Processing, 22(7), July, 2013
[18] V. Mahadevan and N. Vasconcelos, "Biologically Inspired Object Tracking Using Center-Surround Saliency Mechanisms," IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(3), March, 2013
[19] Y. Fang, W. Lin, Z. Chen, C.-M. Tsai and C.-W. Lin, "A Video Saliency Detection Model in Compressed Domain," IEEE Transactions on Circuits and Systems for Video Technology, 21(1), January, 2014
[20] ISO/IEC 23008-2:2013, "Information technology -- High efficiency coding and media delivery in heterogeneous environments -- Part 2: High efficiency video coding," 2013.
[21] W. Mendenhall and T. Sincich, Statistics for Engineering and Sciences, 5th edition, Pearson, 2007.
[22] T. Shanableh, "Detection of frame deletion for digital video forensics," Digital Investigation 10 (4), October, 2013.
[23] S.M. Muddamsetty, D. Sidibe, A. Tremeau and F. Meriaudeau, "A performance evaluation of fusion techniques for spatio-temporal saliency detection in dynamic scenes," IEEE International Conference on Image Processing (ICIP), September, 2013
[24] L. Itti, Laurent and R. Carmi, "Eye-tracking data from human volunteers watching complex video stimuli," CRCNS.org. http://dx.doi.org/10.6080/K0TD9V7F, 2009
[25] L. Itti and P. Baldi, "Bayesian surprise attracts human attention," Advances in Neural Information Processing Systems, 46(8-9), April, 2006.
[26] R. J. Peters and L. Itti, "Applying computational tools to predict gaze direction in interactive visual environments," ACM Transactions on Applied Perception, 5(2), 2008
[27] X. Hou and L. Zhang, "Dynamic visual attention: Searching for coding length increments," Advances in Neural Information Processing Systems, vol. 21, pp.681–688, MIT Press, 2008.
[28] C. Guo and L. Zhang, "A novel multi-resolution spatiotemporal saliency detection model and its applications in image and video compression," IEEE Transactions on Image Process., 19(1), January, 2010

[29] S. Goferman, L. Zelnik-Manor, and A. Tal, “Context-aware saliency detection,” IEEE International Conference on Computer Vision and Pattern Recognition, January, 2010

[30] N. Singh and R. Agrawal, “Combination of Kullback–Leibler divergence and Manhattan distance measures to detect salient objects,” 9(2), pp. 427-435, 2013