Multimed Tools Appl (2017) 76:26297–26328
DOI 10.1007/s11042-016-4124-5

Compressed-domain visual saliency models: a comparative study

Sayed Hossein Khatoonabadi[1] · Ivan V. Bajić[1] · Yufeng Shan[2]

Received: 23 April 2016 / Revised: 15 September 2016 / Accepted: 3 November 2016 /
Published online: 13 December 2016

Abstract Computational modeling of visual saliency has become an important research problem in recent years, with applications in video quality estimation, video compression, object tracking, retargeting, summarization, and so on. While most visual saliency models for dynamic scenes operate on raw video, several models have been developed for use with compressed-domain information such as motion vectors and transform coefficients. This paper presents a comparative study of eleven such models as well as two high-performing pixel-domain saliency models on two eye-tracking datasets using several comparison metrics. The results indicate that highly accurate saliency estimation is possible based only on a partially decoded video bitstream. The strategies that have shown success in compressed-domain saliency modeling are highlighted, and certain challenges are identified as potential avenues for further improvement.

Keywords Visual saliency · Fixation points · Compressed-domain processing · Motion vectors

✉ Sayed Hossein Khatoonabadi
skhatoon@sfu.ca

Ivan V. Bajić
ibajic@ensc.sfu.ca

Yufeng Shan
yshan@cisco.com

[1] Simon Fraser University, Burnaby, BC, Canada

[2] Cisco Systems Boxborough, Boxborough, MA, USA

1 Introduction

Visual saliency estimation is a process of finding certain parts in an image (or video) that are likely to draw attention compared to their spatial (and temporal) surroundings. The Human Visual System (HVS) is able to automatically shift the focus of attention to salient regions in the pre-attentive, early vision phase. This ability allows the brain to restrict high-level processing of a scene to a relatively small part at any given time. Many models have been introduced based on physiological and psychophysical findings to imitate the HVS in order to predict human visual attention [37]. Visual saliency models find a large number of applications in image processing and computer vision, such as quality assessment [10, 19, 50, 56, 69, 82], compression [22, 24, 25, 32, 53, 81], retargeting [16, 59], segmentation [21, 67], object recognition [28], object tracking [63], abstraction [39], guiding visual attention [27, 64], and so on.

Many computational models have been introduced during the past 25 years to estimate visual saliency. Despite the existence of numerous models, their high computational complexity is a serious drawback when it comes to practical applications that need to run in real time, or for devices with restricted complexity and memory requirements, such as mobile devices. One way to reduce the computational cost of saliency estimation is to use compressed-domain features, such as motion vectors (MVs), motion-compensated prediction residuals or their transform coefficients, and so on. This way, part of decoding can be avoided, a smaller amount of data needs to be processed compared to pixel-domain methods, and some of the information produced during encoding (e.g., MVs and transform coefficients) can be reused [41]. Compressed-domain algorithms for visual saliency estimation have been developed for various applications such as image retargeting [16], video transcoding [57, 75, 85], quality estimation [55], video retrieval [60], video skimming [61], salient motion detection [71], and so on. Although there are relatively few compressed-domain saliency models compared to their pixel-domain counterparts, their potential for practical deployment makes them an important research topic.

The purpose of this paper is to provide a comprehensive comparison among compressed-domain visual saliency models for video, similar to what has been done for pixel-domain models in [6]. The present paper is an extension of our preliminary study in [43] and takes into consideration two well-known pixel-domain algorithms as benchmarks for comparison, as well as two more recent compressed-domain algorithms that have appeared since [43]. It also provides a more extensive comparison involving a larger number of videos from two ground truth data sets, as well as a number of different accuracy metrics. In the literature, existing models have been developed for different applications and their evaluation was based on different datasets and quantitative criteria. Furthermore, models are often tailored to a particular video coding standard, and the encoding parameter settings used in the evaluation are often not reported. All of this makes a fair and comprehensive comparison more challenging. To enable meaningful comparison, in this work we reimplemented all compared methods on the same platform, and evaluated them under the same encoding conditions on two popular eye-tracking datasets. A number of different metrics has been employed in the comparison in order to illuminate various aspects of the models' performance. The results of the comparison indicate which strategies seem promising in the context of compressed-domain saliency estimation for video, and point the way towards improving existing models and developing new ones. Last but not least, this study has been performed in a reproducible research manner [80]. The MATLAB code and data used in this study are available online [42].

The paper is organized as follows. Section 2 reviews the visual saliency models used in the study and the two ground truth gaze point datasets. Section 3 describes the evaluation framework, including the accuracy metrics employed in the evaluation and the procedures used to correct for center bias and border effects. Section 4 presents the results of the evaluation, while Sections 5 and 6 provide discussion and conclusions, respectively.

2 Models and data

Our study includes eleven compressed-domain saliency models. Their performance is compared amongst themselves, and also against two high-performing pixel-domain models in order to gain insight into the relationship between the accuracy of the current state of the art in pixel-domain and compressed-domain saliency estimation for video. Among the pixel-domain models, we chose AWS (Adaptive Whitening Saliency) [2], which takes only spatial information into account, and GBVS (Graph-Based Visual Saliency) [29] with DIOFM channels (DKL-color, Intensity, Orientation, Flicker, and Motion), which takes both spatial and temporal information into account for estimating the saliency. AWS is frequently reported as one of the top performing models on still natural images [6, 47]. GBVS is another well-known model, often used as a benchmark for comparison. Since MATLAB implementations of both these models are available, it makes the computational comparison with MATLAB implementations of compressed-domain models meaningful. This section briefly describes the eleven compressed-domain visual saliency models included in the study and the datasets used to evaluate them.

2.1 Compressed-domain visual saliency models

In this study, our goal is to evaluate visual saliency models for video that have been designed explicitly for, or have the potential to work in, the compressed domain. This means that they should operate with the kind of information found in a compressed video bitstream, such as block-based Motion Vector Field (MVF), prediction residuals or their transforms, block coding modes, etc. We surveyed the literature on the topic and found eleven prominent models listed in Table 1, sorted according to the publication year. Different models assume different coding standards, for example MPEG-1, MPEG-2, MPEG-4 SP (Simple Profile), MPEG-4 ASP (Advanced Simple Profile), and MPEG-4 part 10, better known as H.264/AVC (Advanced Video Coding). For each model, the data used from the compressed bitstream, their intended application, as well as data and evaluation method, if any, are also included in the table. As seen in the table, only a few of the most recent models have been evaluated using gaze data from eye-tracking experiments, which is thought to be the ultimate test for a visual saliency model. This fact makes the present study all the more relevant. Interested readers are referred to the supplementary material [42] for a brief description of various models used in the study.

In addition to the visual saliency models described above, two benchmark models were used in the evaluation: IO and GAUSS. These are derived from the ground truth data itself and will be described in Section 2.3, after the two eye-tracking datasets employed in the study are introduced.

Table 1 Compressed-domain visual saliency models included in the study

#	Model	First Author	Year	Codec	Data	Application	Sequences	Gaze data	Metric(s)
1	**PMES**	Ma [60]	2001	MPEG-1/2	MVF	Video Retrieval	MPEG-7 [70]	–	–
2	**MAM**	Ma [61]	2002	MPEG-1/2	MVF	Video Skimming	Specific [61]	–	Human Score
3	**PIM-ZEN**	Agarwal [3]	2003	MPEG-1/2	MVF+DCT-R	ROI Detection	QCIF Standard	–	–
4	**PIM-MCS**	Sinha [75]	2004	MPEG-4 SP	MVF+DCT-R	Video Transcoding	QCIF Various	–	–
5	**MCSDM**	Liu [57]	2009	H.264/AVC	MVF	Rate Control	QCIF Standard	–	–
6	**GAUS-CS**	Fang [17]	2012	MPEG-4 ASP	MVF+DCT-P	Saliency Detection	CRCNS [35, 36]	Yes	KLD
7	**MSM-SM**	Muthuswamy [71]	2013	MPEG-2	MVF+DCT-P/R	Saliency Detection	Mahadevan [62]	–	ROC
8	**APPROX**	Hadizadeh [23]	2013	–	MVF+DCT-P	Video Compression	SFU [26]	Yes	KLD+AUC
9	**PNSP-CS**	Fang [18]	2014	MPEG-4 ASP	MVF+DCT-P	Saliency Detection	CRCNS [35, 36]	Yes	KLD+AUC
10	**OBDL-MRF**	Khatoonabadi [46]	2015	H.264/AVC	OBDL	Saliency Detection	SFU [26]+DIEM [12]	Yes	AUC+NSS
11	**MVE+SRN**	Khatoonabadi [44, 45]	2015	H.264/AVC	MVF+DCT-R	Saliency Detection	SFU [26]+DIEM [12]	Yes	AUC+NSS+JSD

(MVF: Motion Vector Field; DCT-R: Discrete Cosine Transformation of residual blocks; DCT-P: Discrete Cosine Transformation of pixel blocks; OBDL: Operational Block Description Length; KLD: Kullback-Leibler Divergence; AUC: Area Under Curve; ROC: Receiver Operating Characteristic; NSS: Normalized Scanpath Saliency; JSD: Jensen-Shannon Divergence)

2.2 Eye-tracking video datasets

Eye-tracking data is the most typical psychophysical ground truth for visual saliency models [15]. To evaluate saliency models, each model's saliency map is compared with recorded gaze locations of the subjects. Two recent publicly available eye-tracking datasets were used in the study. The reader is referred to [84] for an overview of other existing datasets in the field.

2.2.1 The SFU dataset

The SFU eye-tracking dataset [26] consists of twelve CIF (352 × 288) sequences that have become popular in the video compression and communications community: *Bus*, *City*, *Crew*, *Foreman*, *Flower Garden*, *Hall Monitor*, *Harbour*, *Mobile Calendar*, *Mother and Daughter*, *Soccer*, *Stefan*, and *Tempete*. A total of 15 participants watched all 12 videos while wearing a Locarna Pt-mini head-mounted eye tracker. Each participant took part in the test twice, resulting in two sets of viewings per participant for each video. The first viewing is used as ground truth for evaluating the performance of saliency models, whereas the data from the second viewing is used to construct benchmark models, as described in Section 2.3. The results in [26] showed that gaze locations in the first and second viewings can differ notably, however they remain relatively close to each other when there is a single dominant salient region in the scene (for example, the face in the *Foreman* sequence.) As a result, it is reasonable to expect that good saliency models will produce high scores for those frames where the first and second viewing data agree. A sample frame from each video has been shown in Fig. 1, overlaid with the gaze locations from both viewings. The visualization is such that the less-attended regions (according to the first viewing) are indicated by darker colors. Further details about this dataset are shown in Table 2.

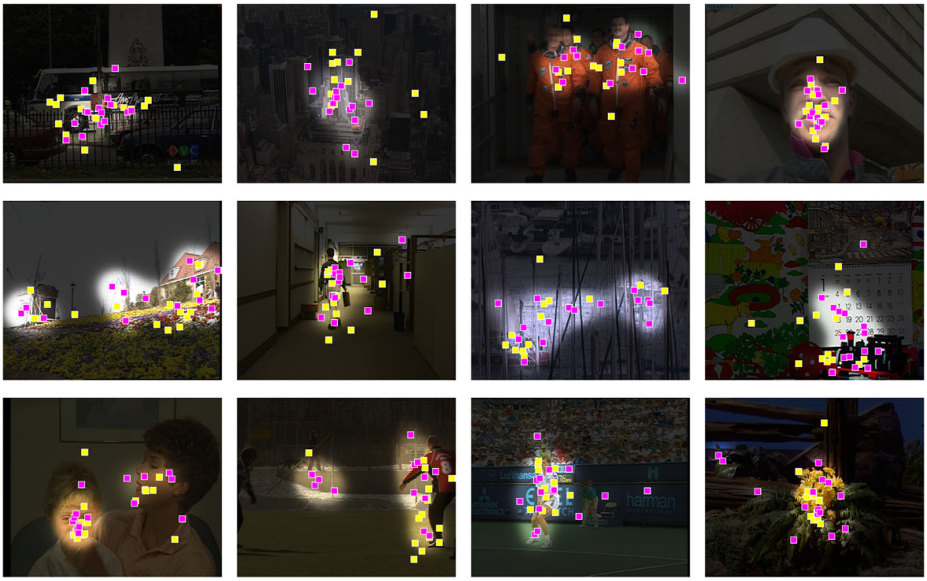

Fig. 1 Sample gaze visualization from the SFU Dataset. The gaze points from the first viewing are indicated as *pink squares*, those from the second viewing as *yellow squares*

Table 2 Datasets used to evaluate compressed-domain visual saliency models

Dataset	**SFU**	DIEM
Year	2012	2011
Sequences	12	85
Display Resolution	704 × 576*	varying
Format	RAW	MPEG-4
FPS	30	30
Frames	90-300	888-3401
Participants	15	35-53§
Viewings	2†	2‡
Screen Resolution	1280 × 1024	1600 × 1200
Screen Diagonal	19 + ”+	21.3”+
Viewing Distance	80 cm	90 cm

*The original video resolution (352 × 288) was doubled during the presentation to the participants

†Each participant watched each sequence twice, after several minutes

‡Viewings for the left/right eye are available

§A total of 250 subjects participated in the study, but not all of them viewed each video; the number of viewers per video was 35-53

2.2.2 The DIEM dataset

Dynamic Images and Eye Movements (DIEM) project [12] provides tools and data to study how people look at dynamic scenes. So far, DIEM collected gaze data for 85 sequences of 30 fps videos varying in the number of frames and resolution, using the SR Research Eyelink 1000 eye tracker. The videos were taken from various categories including movie trailers, music videos, documentary, news and advertisements. For the purpose of the study, the frames of the sequences from the DIEM dataset were re-sized to 288 pixels height, while securing the original aspect ratio, resulting in five different resolutions: 352 × 288, 384 × 288, 512 × 288, 640 × 288 and 672 × 288. Among 85 available videos, 20 sequences similar to those used in [6] were chosen for the study, and, to match the length of the SFU sequences, only the first 300 frames were used in the comparison. In the DIEM dataset, the gaze location of both eyes are available. The gaze locations of the right eye were used as ground truth in the study, while gaze locations of the left eye were used to construct benchmark models, as described in Section 2.3. Clearly, the gaze points of the two eyes are very close to each other, closer than the gaze points of the first and second viewing in the SFU dataset. A sample frame form each selected sequence, overlaid with gaze locations of both eyes, is illustrated in Fig. 2. The visualization is such that the less-attended regions (according to the right eye) are indicated by darker colors.

2.3 Benchmark models

In addition to the computational saliency models, we consider two additional models: Intra-Observer (IO) and Gaussian center-bias (GAUSS). IO saliency map is obtained by the

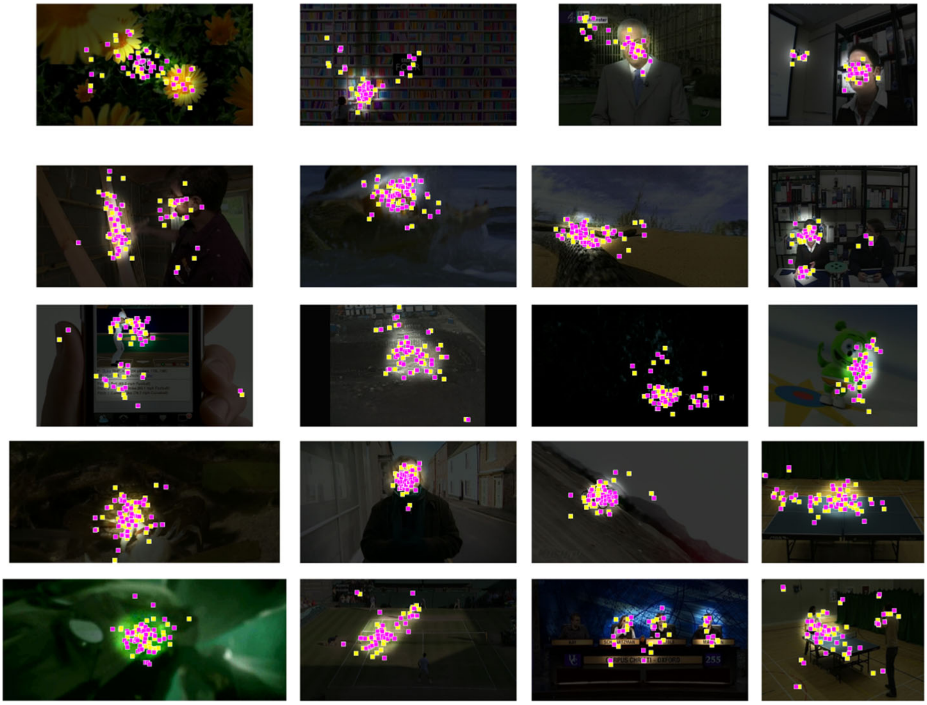

Fig. 2 Sample gaze visualization from the DIEM Dataset. The gaze points of the right eye are shown as *pink squares*, those of the left eye as *yellow squares*

convolution of a 2D Gaussian blob (with standard deviation of 1° of visual angle) with the second set of gaze points of the same observer within the dataset. Recall that both datasets have two sets of gaze points for each sequence and each observer – first/second viewing in the SFU dataset, right/left eye in the DIEM dataset. So the IO saliency maps for the sequences in the SFU dataset are obtained using the gaze points from the second viewing, while IO saliency maps for the sequences from the DIEM dataset are obtained using the gaze points of the left eye. These IO saliency maps can be considered as indicators of the best possible performance of a visual saliency model, especially in the DIEM dataset where the right and left eye gaze points are always close to each other.

On the other hand, GAUSS saliency map is just a 2D Gaussian blob with the standard deviation of 1° located at the center of the frame. This model assumes that the center of a frame is the most salient point. Center bias turns out to be surprisingly powerful and has been used occasionally to boost the performance of saliency models without taking scene content into account. The underlying assumption is that the person recording the image or video will attempt to keep the salient objects at or near the middle of the frame. On average, this assumption is not too bad. Fig. 3 shows the heatmaps indicating cumulative gaze point locations across all sequences and all participants in the SFU dataset (first viewing) and DIEM dataset (right eye). As seen in the figure, aggregate gaze point locations do indeed cluster around the center of the frame. However, since GAUSS does not take content into account, one could expect a good saliency model to outperform it.

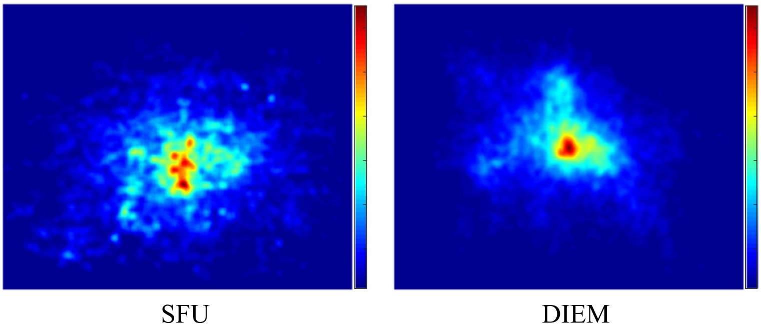

Fig. 3 The heatmap visualization of gaze points combined across all frames and all observers, for the first viewing in the SFU dataset and the right eye in the DIEM dataset. Gaze points accumulate near the center of the frame

3 Evaluation framework

3.1 Implementation settings

In order to have a unified framework for comparison, we have implemented all models in MATLAB 8.5 on the same machine, an Intel (R) Core (TM) i7 CPU at 3.40 GHz and 16 GB RAM running 64-bit Windows 8.1. Where possible, we verified the implementation by comparing the results with those presented in the corresponding papers and/or by contacting the authors. As seen in Table 1, each model assumed a certain video coding standard.[1] However, fundamentally, all models except OBDL-MRF [46] rely on the same type of information – MVs and DCT of residual blocks (DCT-R) or pixel blocks (DCT-P). The main difference is in the size of the blocks to which MVs are assigned or to which DCT is applied. OBDL-MRF, on the other hand, directly uses block description length as an indicator of saliency, without the need to decode MVs or prediction residuals. In standards up to MPEG-4 ASP, the minimum block size was 8 × 8, whereas H.264/AVC allowed block sizes down to 4 × 4 [83]. In pursuance of a fair comparison, for which all models should accept the same input data, we chose to encode all videos in two currently most widely used video formats – MPEG-4 ASP and H.264/AVC. Each choice ensured that seven out of eleven models in the study did not require modification. Minor modification was necessary in order for APPROX [23] to accept compressed input data. Specifically, for MPEG-4 ASP input data, where the spatial saliency map relies on DCT values of 16 × 16 pixel blocks, the 16 × 16 DCT was computed from the 8 × 8 DCTs using a fast algorithm from [30]. Also, minimum MV block size was set to 8 × 8. In case of H.264/AVC input data, only P-frames were considered. In summary, the first group of models that takes MPEG-4 ASP bitstream as an input comprises of models {1, 2, 3, 4, 6, 7, 8, 9} from Table 1, while the second group that takes H.264/AVC bitstreams includes models {1, 2, 3, 4, 5, 8, 10, 11} in the table.

We considered two configurations to encode the videos used in the evaluation. For the first group, the Group-of-Pictures (GOP) structure was set to IPPP with the GOP size of 12, i.e., the first frame is coded as intra (I), the next 11 frames are coded predictively (P), then the next frame is coded as I, and so on. The MV search range was set to 16 with 1/4-pel motion compensation with QP$\in$\{1, 4, 7, ..., 31\}. In the decoding stage, the DCT-P values

[1] A block diagram of a generic video encoder/decoder is provided in the supplementary material [42].

(in I-frames) and DCT-R values (in P-frames), as well as MVs (in P-frames) were extracted from the encoded MPEG-4 ASP bitstream for each 8 × 8 block. For the second group, encoding was done using H.264/AVC with QP∈{3, 6, 9, ..., 51} in the baseline profile. In our setting, for each macroblock, there exists up to four MVs having 1/4-pixel accuracy with no range restriction. Other settings were set to default. Encoding and partial decoding to extract the required data was accomplished using the FFMPEG library [20].

3.2 Accuracy evaluation

A number of methods have been used to evaluate the accuracy of visual saliency models with respect to gaze point data [5, 6, 14, 33, 34, 52]. Since each method emphasizes a particular aspect of model's performance, to make the evaluation balanced, a collection of methods and metrics is employed in this study. A model that offers high score across many metrics can be considered to be fairly accurate.

3.2.1 Area under curve (AUC)

The area under curve or, more precisely, the area under Receiver Operating Characteristic (ROC) curve, is computed from the graph of the True Positive Rate (TPR) versus the False Positive Rate (FPR) at various threshold parameters [77]. In the context of saliency maps, the saliency values are first divided into positive and negative sets corresponding to gaze and non-gaze points. Then for any given threshold, TPR and FPR are, respectively, obtained as the fraction of elements in the positive set and in the negative set that are greater than the threshold. Essentially, by varying the threshold, the ROC curve of TPR versus FPR is generated, visualizing the performance of a saliency model across all possible thresholds. The area under this curve quantifies the performance and shows how well the saliency map can predict gaze points. A larger AUC implies a greater correspondence between gaze locations and saliency predictions. A small AUC indicates weaker correspondence. The AUC is in the range [0, 1]: the value of 1 indicates the saliency algorithm performs well, the value of 0.5 represents pure chance performance, and the value of less than 0.5 represents worse than pure chance performance. This metric is also invariant to monotonic scaling of saliency maps [7].

It is worth mentioning that instead of using all non-gaze saliency values, these are usually sampled [14, 74]. The idea behind this approach is that an effective saliency model would have higher values at fixation points than at randomly sampled points. Control points for non-gaze saliency values are obtained with the help of a nonparametric bootstrap technique [13], and sampled with replacement, with sample size equal to the number of gaze points, from non-gaze parts of the frame multiple times. Finally, the average of the statistic over all bootstrap subsamples is taken as a sample mean.

3.2.2 Kullback-Leibler divergence (KLD) and J-divergence (JD)

The KLD is often used to obtain the divergence between two probability distributions. It is given by the relative entropy of one distribution with respect to another [49]

$$KLD(P\|Q) = \sum_{i=1}^{r} P(i) \cdot \log_b \left(\frac{P(i)}{Q(i)} \right), \tag{1}$$

where P and Q are discrete probability distributions, b is the logarithmic base, and r indicates the number of bins in each distribution. Note that KLD is asymmetric. The symmetric version of KLD, also called J-Divergence, is [38]

$$JD(P\|Q) = KLD(P\|Q) + KLD(Q\|P). \tag{2}$$

To assess how accurately a saliency model predicts gaze locations based on the symmetric KLD, the distribution of saliency values at the gaze locations is compared against the distribution of saliency values at some random points from non-gaze locations [33–35]. If these two distributions overlap substantially, i.e., the divergence JD approaches zero, then the saliency model predicts gaze points no better than a random guess. On the other hand, as one distribution diverges from the other and the divergence JD increases, the saliency model is better able to predict gaze points.

Specifically, let there be n gaze points in a frame. Another n points different from the gaze points are randomly selected from the frame. The saliency values at the gaze points and the randomly selected points constitute the two distributions, P and Q. A good saliency model would produce a large JD. The process of choosing random samples and computing the JD is usually repeated many times and the resulting JD values are averaged to minimize the effect of random variations. While JD has certain advantages (please refer to [5, 34] for details), it also faces several problems. One of the problems with KLD and JD is the lack of an upper bound [48]. Another problem is that if $P(i)$ or $Q(i)$ is zero for some i, one of the terms in (2) is undefined. For these reasons, JD was not used in the present study.

3.2.3 Jensen-Shannon divergence (JSD)

The Jensen-Shannon divergence (JSD) is a KLD-based metric that avoids some of the problems faced by KLD and JD [54]. For two probability distributions P and Q, JSD is defined as [11]:

$$JSD(P\|Q) = \frac{KLD(P\|R) + KLD(Q\|R)}{2}, \tag{3}$$

where

$$R = \frac{P+Q}{2}. \tag{4}$$

Unlike KLD, JSD is a proper metric, is symmetric in P and Q, and is bounded in [0, 1] if the logarithmic base is set to $b = 2$ [54]. The value of the JSD for the saliency map that perfectly predicts gaze points will be equal to 1. The same sampling strategy employed in AUC computation can also be used for computing JSD.

3.2.4 Normalized scanpath saliency (NSS)

NSS measures the strength of normalized saliency values at gaze locations [73]. Normalization is affine so that the resulting normalized saliency map has zero mean and unit standard deviation. The NSS is defined as the average of normalized saliency values at gaze points. A positive normalized saliency value at a certain gaze point indicates that the gaze point matches one of the predicted salient regions, zero indicates no link between predictions and the gaze point, while a negative value indicates that the gaze point has fallen into an area predicted to be non-salient.

3.2.5 *Pearson correlation coefficient (PCC)*

PCC measures the strength of the linear relationship between a predicted saliency map S and the ground truth map G. First, the ground truth map G is obtained by convolving the gaze point map with a 2D Gaussian function having the standard deviation of 1° of visual angle [52]. Then S and G are treated as random variables whose paired samples are given by values of the two maps at each pixel position in the frame. The Pearson correlation coefficient is defined as

$$corr(G, S) = \frac{cov(G, S)}{\sigma_G \sigma_S}, \tag{5}$$

where $cov(\cdot, \cdot)$ denotes covariance and σ_G and σ_S are, respectively, the standard deviations of the ground truth map and the predicted saliency map. The value of PCC is between -1 and 1; the value of ± 1 indicates the strongest linear relationship, whereas the value of 0 indicates no correlation. If the model's saliency values tend to increase as the values in the ground truth map increase, the PCC is positive. Otherwise, if the model's saliency values tend to decrease as the ground truth values increase, the PCC is negative. In this context, a PCC value of -1 would mean that the model predicts non-salient regions as salient, and salient regions as non-salient. While this is the opposite of what is needed, such model can still be considered accurate if its saliency map is inverted. While PCC is widely used for studying relationships between random variables, in its default form it has some shortcomings in the context of saliency model evaluation, especially due to center bias, as discussed in the next section.

3.3 Data analysis considerations

Here, we discuss several considerations about the ground truth data, and the methods and metrics used in the evaluation.

3.3.1 *Gaze point uncertainty*

Eye-tracking datasets usually report a single point (x, y) as the gaze point of a given subject in a given frame. However, such data should not be treated as absolute. There are at least two sources of uncertainty in the measurement of gaze points. One is the eye-tracker's measurement error, which is usually on the order of 0.5° to 1° of the visual angle [58, 68, 76]. The other source of uncertainty is the involuntary eye movement during fixations. The human eye does not concentrate on a stationary point during a fixation, but instead constantly makes small rapid movements to make the image more clear [9]. Depending on the implementation, the eye tracker may filter those rapid movements out, either due to undersampling or to create an impression of a more stable fixation. For at least these two reasons, the gaze point measurement reported by an eye tracker contains some uncertainty. At the current state of technology, the eye tracker measurement errors seem to be larger than the uncertainty caused by involuntary drifts, and so we take them as the dominant source of noise in the ground truth data. To account for this noise, we apply a local maximum operator in a radius of 0.5° of visual angle. In other words, when computing a saliency value of a given point in a frame, the maximum value within its small neighborhood is used.

The use of the local maximum operator is meant to counter the effects of measurement noise in the gaze tracking system, which is usually rated at around 0.5° of visual angle. Hence, the true fixation point may be within 0.5° of visual angle away from what the gaze measurement system reports. An accurate saliency model produces small saliency values at locations far from fixation points and high saliency values at locations near fixation points. Therefore, for an accurate model, applying a local maximum operator does not change saliency values away from fixations while it ensures that near fixations, the maximum predicted saliency value within the measurement tolerance is considered. So we expect that the accuracy score for an accurate model will increase using this approach. For an inaccurate model (one that produces low saliency values near fixations and large ones away from fixations), we expect little or no change in the score. This is because its predicted saliency values near fixations (which are low) will not increase, while its predicted saliency values away from fixations may get a boost, but they don't matter because they are away from fixations anyway.

3.3.2 Center bias and border effects

A person recording a video will generally tend to put regions of interest near the center of the frame [72, 79]. In addition, people also have a tendency to look at the center of the image [78], presumably to maximize the coverage of the displayed image by their field of view. These phenomena are known as center bias. Figure 3 illustrates the center bias in the SFU and DIEM datasets by displaying the locations of gaze points accumulated over all sequences and all frames.

Interestingly, Kanan et al. [40] and Borji et al. [6] showed that creating a saliency map merely by placing a Gaussian blob at the center of the frame may result in fairly high scores. Such high scores are partly caused by using a uniform spatial distribution over the image when selecting control samples. Specifically, the computation of AUC, KLD and JSD for a given model involves choosing non-gaze control points randomly in an image. If these are chosen according to a uniform distribution across the image, the process results in many control points near the border, which, empirically, have little chance of being salient. As a result, the saliency values of those control points tend to be small, resulting in an artificially high score for the model under test. At the same time, since gaze points are likely located near the center of the frame, a centered Gaussian blob would tend to match many of the gaze points, which would make its NSS and PCC scores high.

Additionally, Zhang et al. [86] thoroughly investigated the effect of dummy zero borders against evaluation metrics. Adding dummy zero saliency values at the border of the image changes the distribution of saliency of the random samples as well as the normalization parameters in NSS, leading to different scores while the saliency prediction is unchanged. To decrease sensitivity to center bias and border effect, Tatler et al. [79] and Parkhurst and Niebur [72] suggested to distribute random samples according to the measured gaze points. To this end, Tatler et al. [79] distributed random samples from human saccades and choose control points for the current image randomly from fixation points in other images in their dataset. Kanan et al. [40] also picked saliency values at the gaze points in the current image, while control samples were chosen randomly from the fixations in other images in the dataset. For both techniques, control points are drawn from a non-uniform random distribution according to the measured fixations, decreasing the effect of center bias. Furthermore, this way, dummy zero borders will not affect the distribution of random samples.

In this paper, we use a similar approach for handling center bias and border effects. Instead of directly using the accumulated gaze points over all frames in the dataset (Fig. 3), we fit a 2D Gaussian distribution to the accumulated gaze points across both SFU and DIEM datasets. Then, control samples are chosen randomly from the fitted 2D Gaussian distribution. This reduces center bias in AUC and JSD.

Table 3 Summary of evaluation metrics used in the study

Metric	Symmetric	Bounded	Center-biased	Applicability	Input
AUC	Yes	Yes	Yes	General	Location
AUC′	Yes	Yes	No	Saliency	Location
KLD	No	No	Yes	General	Distribution
JD	Yes	No	Yes	General	Distribution
JSD	Yes	Yes	Yes	General	Distribution
JSD′	Yes	Yes	No	Saliency	Distribution
NSS	Yes	No	Yes	Saliency	Value
NSS′	Yes	No	No	Saliency	Value
PCC	Yes	Yes	Yes	General	Distribution

To reduce center bias and border effects in NSS, we modify the normalization as

$$S'(x, y) = \frac{S(x, y) - \widetilde{\mu}}{\widetilde{\sigma}}, \tag{6}$$

where

$$\widetilde{\mu} = \frac{1}{N} \sum_{(x,y)} F(x, y) \cdot S(x, y), \tag{7}$$

$$\widetilde{\sigma} = \sqrt{\frac{1}{N-1} \sum_{(x,y)} (F(x, y) \cdot S(x, y) - \widetilde{\mu})^2}. \tag{8}$$

In the above equations, (x, y) are the pixel coordinates, N is the total number of pixels, and $F(x, y)$ is the fitted 2D Gaussian density evaluated at (x, y) normalized such that it sums up to 1. In the normalization described by the above three equations, the pixels located near the center of the image are given more significance due to F. In other words, saliency predictions have the same bias as observers' fixations. These accuracy measures that are modified to reduce the center bias and border effects are indicated by prime (′) and referred to as NSS′, AUC′, and JSD′. We summarize all above-mentioned metrics in Table 3. Metrics can be divided by symmetry (column 2) or boundedness (column 3). Some metrics favor center-biased saliency models (column 4). Also, some metrics are specific to saliency while others have more general applicability (column 5), e.g., for comparing two distributions. The input data for various metrics comes from three sources (column 6): 1) the locations associated with estimated saliency 2) the distribution of estimated saliency and 3) the values of estimated saliency at fixation points.

4 Results

4.1 Qualitative comparison

In this section, we show a qualitative comparison of saliency maps produced by various models on two specific examples.[2] Figures 4 and 5 show the saliency maps for frame #150 of *City* and frame #150 of *one-show* produced from MPEG-4 ASP and H.264/AVC

[2] Additional examples are provided in the supplementary material [42].

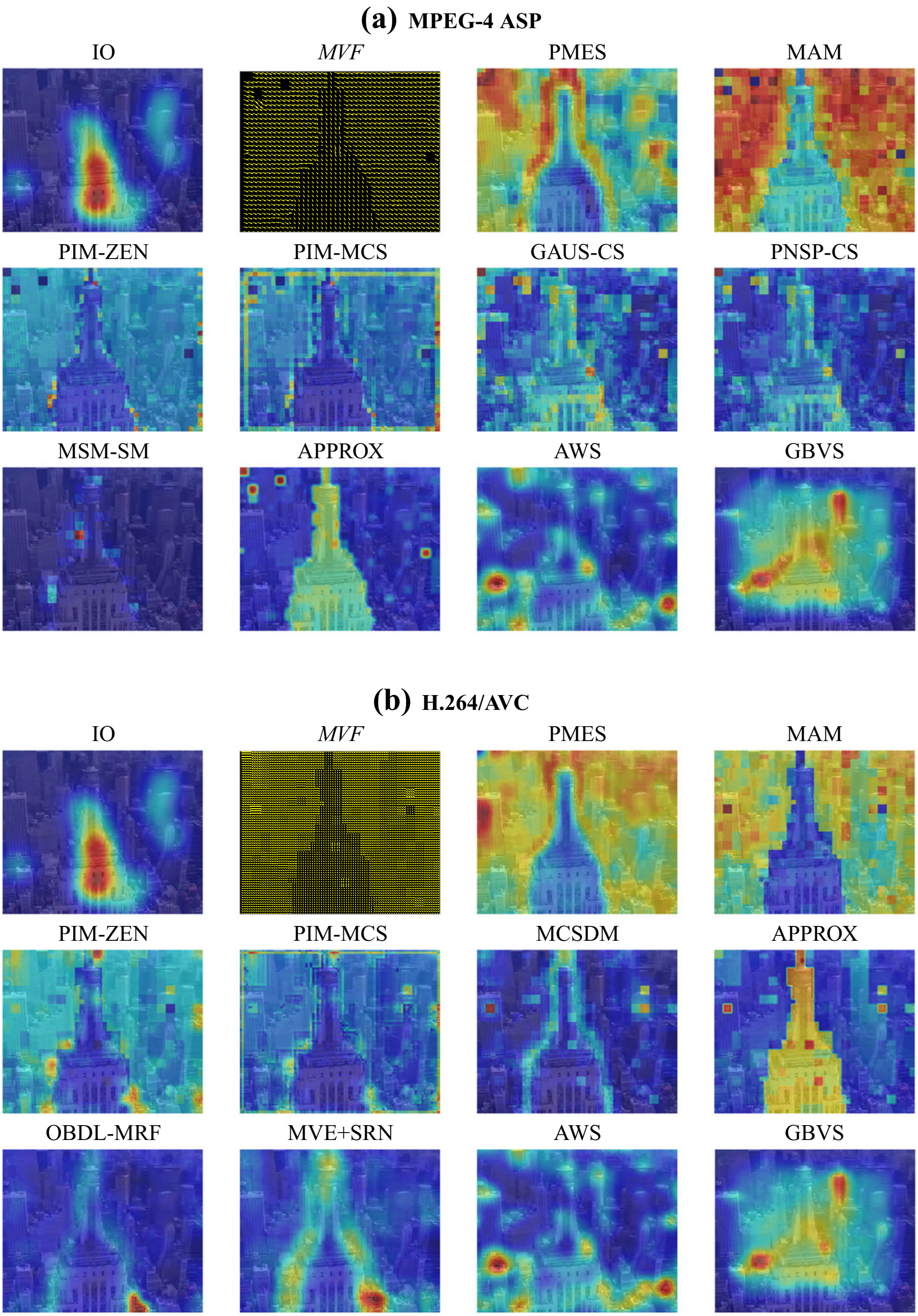

Fig. 4 Sample saliency maps obtained by various models for *City*

bitstreams. The QP values for MPEG-4 ASP and H.264/AVC were set to 16 and 36, respectively. This selection brings about the same average PSNR ($\approx$ 30.0dB) over the whole

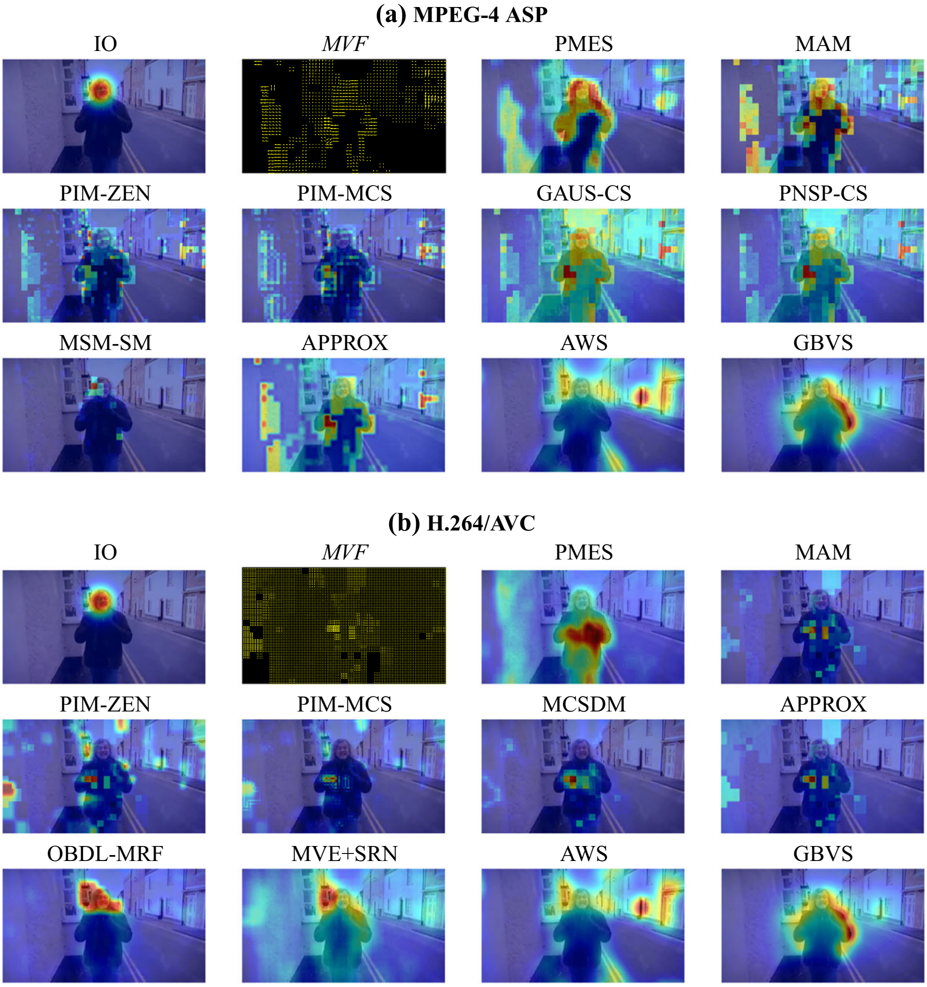

Fig. 5 Sample saliency maps obtained by various models for *one-show*

sequence. In the figure, the MVF of each frame is also shown. Note that due to the differences in MVs and residuals of MPEG-4 ASP and H.264/AVC, the resulting saliency maps for those models that are able to accept both formats can be different. In the figures, these would be PMES, MAM, PIM-ZEN, PIM-MCS and APPROX. On *City*, saliency maps produced by the same model from two different bitstreams do resemble each other, but on *one-show* they could be quite different. This is mainly due to the fact that the MVF on *City* is more consistent between MPEG-4 ASP and H.264/AVC, whereas on *one-show*, the two encoders produce fairly different MVFs.

In *City*, all the motion is due to camera movement. While observers typically look at the building in the center of the frame (see IO in Figs. 4 and 5), all models, no matter which encoding is used, declare the boundary of the building as salient, where local motion is different from the global motion. Meanwhile, APPROX is also able to detect the central building as salient. Note that APPROX is the only model in the study that employs global motion compensation (GMC) and its high scores on *City* (e.g. AUC$'$ = 0.57 over MPEG-4 ASP and AUC$'$ = 0.58 over H.264/AVC encoded bitstream) comparing to IO score (AUC$'$ = 0.65) are an indication that other models could benefit from incorporating GMC.

In *one-show*, large noisy MVs in low-texture areas cause all compressed-domain models except MSM-SM and OBDL-MRF to mistakenly declare them as salient regions. Note that MSM-SM does not directly use motion magnitude but rather uses processed MVs in the form of a motion binary map. Meanwhile, OBDL-MRF uses the number of bits per block, rather than any direct measure of motion magnitude, to predict saliency. In this sequence, observers mostly focus on the face (see IO in Figs. 4 and 5) so a model that was able to perform face detection would have done well in this example. Unfortunately, none of the models is currently able to do face detection in the compressed domain - this seems like a rather challenging problem. AWS and GBVS also declare some part of non-salient regions as salient.

4.2 Quantitative comparison

First, we present quantitative assessment of the saliency models using the MPEG-4 ASP encoded data from the SFU and DIEM datasets. We start with the assessment based on AUC$'$. Figure 6 shows the average AUC$'$ scores of various models across the test sequences. Note that all models are able to produce saliency maps for P-frames, while only some of them are able to produce a saliency map for I-frames. Hence, Fig. 6a shows the average AUC$'$ scores on I-frames for those models able to handle I-frames, while Fig. 6b shows the average AUC$'$ scores for all models on P-frames. Sequences from the SFU dataset are indicated with capital first letter.

As seen in the figure, all models achieved average AUC$'$ scores between those of IO, which represents a kind of an upper bound (especially on the DIEM dataset), and GAUSS, which represents center-biased, content-independent static saliency map. Note that GAUSS itself has a slightly better AUC$'$ score than the pure chance score of 0.5. Recall that AUC$'$ corrects for center bias by random sampling of control points based on empirical gaze distribution across all frames and all sequences. It is encouraging that all models are able to surpass GAUSS and achieve average AUC$'$ scores around 0.6.

Another interesting point in Fig. 6 is an indication of how difficult or easy is saliency prediction in a given sequence according to AUC$'$. In the figure, the sequences are sorted along the horizontal axis in decreasing order of average AUC$'$ score across all models. Although the order is not the same for I- and P-frames, overall, it seems that *one-show* is the one for which saliency prediction is easiest, whereas *City* is the one for which saliency prediction is hardest. We will return to this issue shortly. Note that IO has better performance on the sequences from the DIEM dataset. Here, IO saliency maps are formed by the left eye gaze points and represent an excellent indicator of the ground-truth right eye gaze points. In the sequences from the SFU dataset, where IO saliency map is formed from the gaze points of the second viewing, the IO scores are not as high because the second-viewing gaze points are not as good of a predictor of the ground-truth first-viewing gaze points.

A similar set of results quantifying the models' performance according to NSS$'$ is shown in Fig. 7.[3] As seen in Figs. 6 and 7, the models that are able to handle I-frames ((a) parts

[3] Results for other metrics are provided in the supplementary material [42].

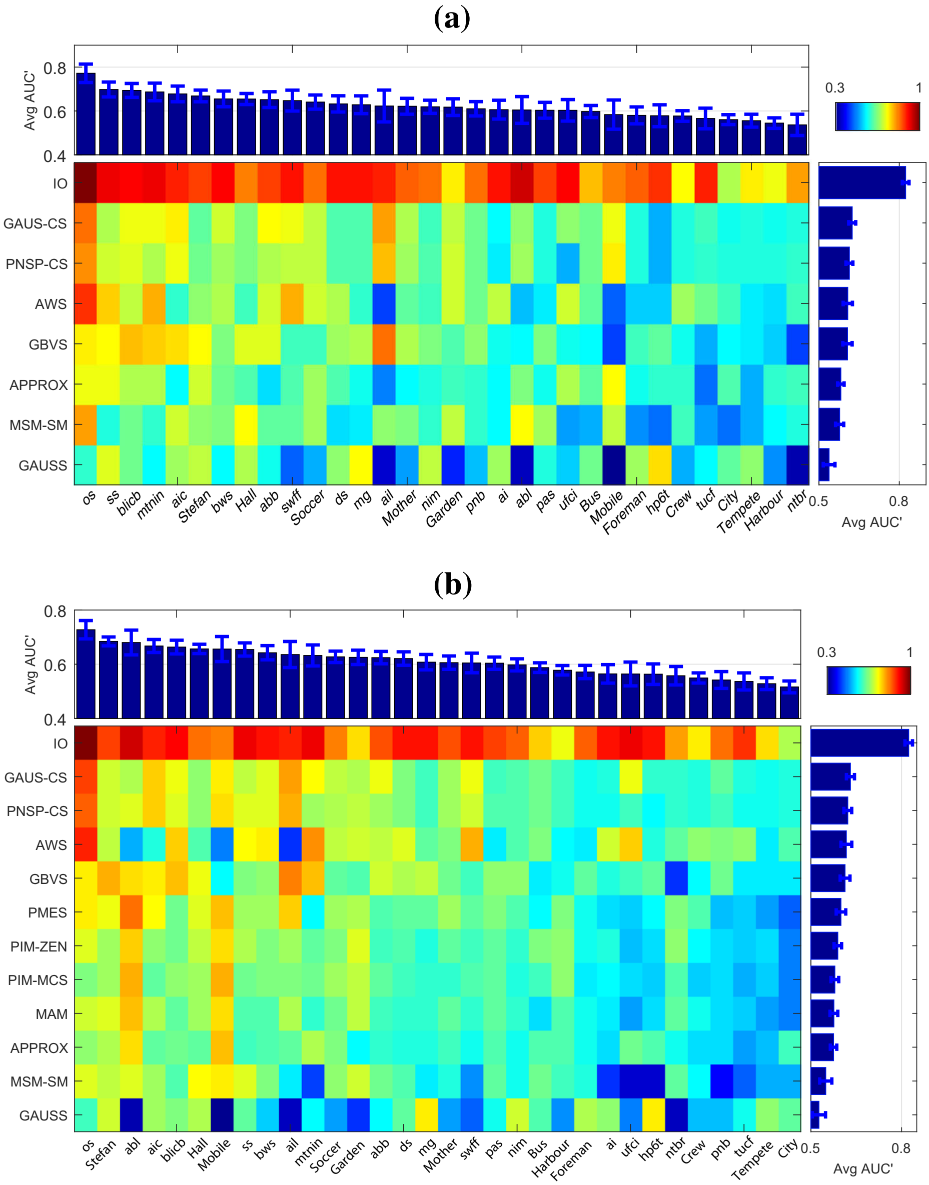

Fig. 6 Accuracy of various saliency models over MPEG-4 ASP encoded sequences according to AUC′ score for (**a**) I-frames and (**b**) P-frames. The *2D color map* shows the average AUC′ score of each model on each sequence. *Top*: Average AUC′ score for each sequence, across all models. *Right*: Average AUC′ score for each model across all sequences. *Error bars* represent standard error of the mean (SEM), $\sigma/\sqrt{n}$, where σ is the sample standard deviation of n samples

of the figures) achieve similar average scores on the I-frames as they do on the P-frames ((b) parts of the figures). For this reason, and to save space, in the remainder of the paper the results for I- and P-frames will sometimes be reported jointly. That is, in such cases,

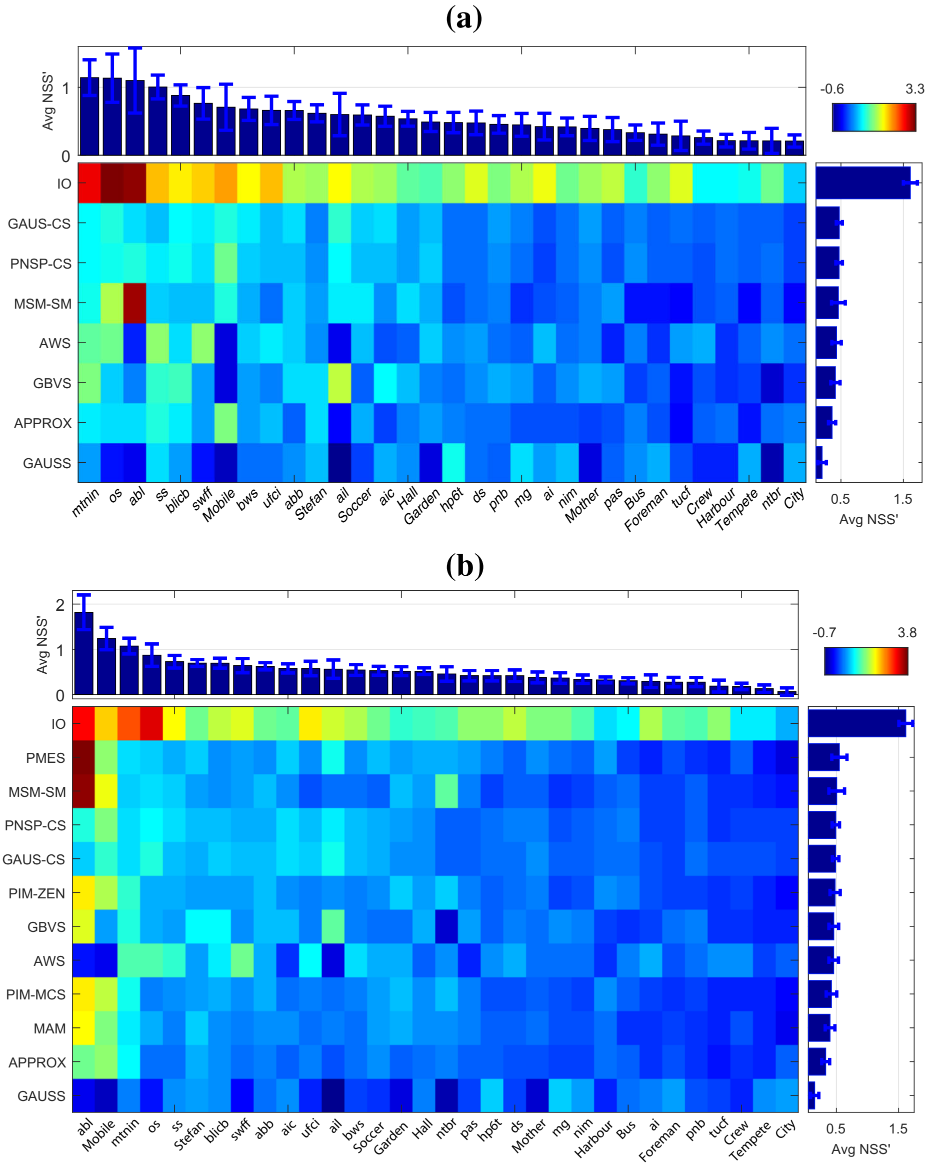

Fig. 7 Accuracy of various saliency models over MPEG-4 ASP encoded sequences according to NSS′ score for (**a**) I-frames and (**b**) P-frames. The *2D color map* shows the average NSS′ score of each model on each sequence. *Top*: Average NSS′ score for each sequence, across all models. *Right*: Average NSS′ score for each model across all sequences. *Error bars* represent standard error of the mean (SEM), $\sigma/\sqrt{n}$, where σ is the sample standard deviation of n samples

all scores will be the averages across all frames that the model is able to handle. Since the number of I-frames is much smaller than the number of P-frames, for the models that are able to handle I-frames, the effect of I-frame scores on the combined score is relatively small.

Table 4 Ranking of MPEG-4 ASP encoded test sequences according to average scores across all models excluding IO and GAUSS

Rank	AUC′	JSD′	NSS′	PCC
1	*os*	*os*	*abl*	*Stefan*
2	*abl*	*Mobile*	*Mobile*	*mtnin*
3	*Mobile*	*Stefan*	*mtnin*	*abl*
4	*Stefan*	*Hall*	*os*	*Mobile*
⋮	⋮	⋮	⋮	⋮
30	*tucf*	*ufci*	*tucf*	*City*
31	*Tempete*	*nim*	*Tempete*	*pnb*
32	*City*	*pnb*	*City*	*Tempete*

Table 4 shows the ranking of test sequences according to the average scores across all models except IO and GAUSS.[4] The sequences are ranked in decreasing order of average scores – the highest-ranked sequences are those for which the average scores are highest, and therefore seem to be the easiest for saliency prediction. Meanwhile, the lowest-ranked sequences are those for which saliency prediction seems the most difficult. Although the ranking differs somewhat for different metrics, overall, *one-show*, *advert-bbc4-library*, *Stefan* and *Mobile Calendar* seem to be among the easiest sequences for saliency prediction, while *City* and *Tempete* are among the hardest. *one-show*, *advert-bbc4-library* and *Stefan* have only one salient object, and all models are generally able to correctly identify them. *Mobile Calendar* contains several moving objects, including a ball and a train. The motion of each of these is sufficiently strong and different from the surroundings that almost all models are able to correctly predict viewers' gaze locations. It should be noted that the background of this sequence involves many static colorful regions that, in the absence of motion, would have the potential to attract attention. It is encouraging that the compressed-based models are generally able to identify the salient moving objects against such colorful and potentially attention-grabbing background. Meanwhile, the two pixel-domain models show a relatively poor performance on this sequence.

On the other hand, *City* and *Tempete* do not contain salient moving objects. In fact, *City* does not contain any moving objects; all the motion in this sequence is due to camera movement. *Tempete* also contains significant camera motion (zoom out) and in addition shows falling yellow leaves that act like motion noise, as they do not attract viewers' attention. While all models get confused by the falling leaves in *Tempete*, APPROX achieves a decent performance on *City* due to its use of global motion compensation (GMC). APPROX is the only model in the study that employs GMC and its success on *City* is an indication that other models could be improved by incorporating GMC. Note that AWS also scores well on *City* because, as a spatial saliency model, it ignores motion and therefore does not get confused by it in this sequence.

The average scores of MPEG-4 ASP based saliency models across all sequences in both datasets are shown in Fig. 8 for various accuracy metrics. Please note that the horizontal axis has been focused on the relevant range of scores. Not surprisingly, IO achieves the highest scores regardless of the metric. At the same time, the effect of center bias is easily

[4] The full ranking across the computational models and the IO model are separately provided in the supplementary material [42].

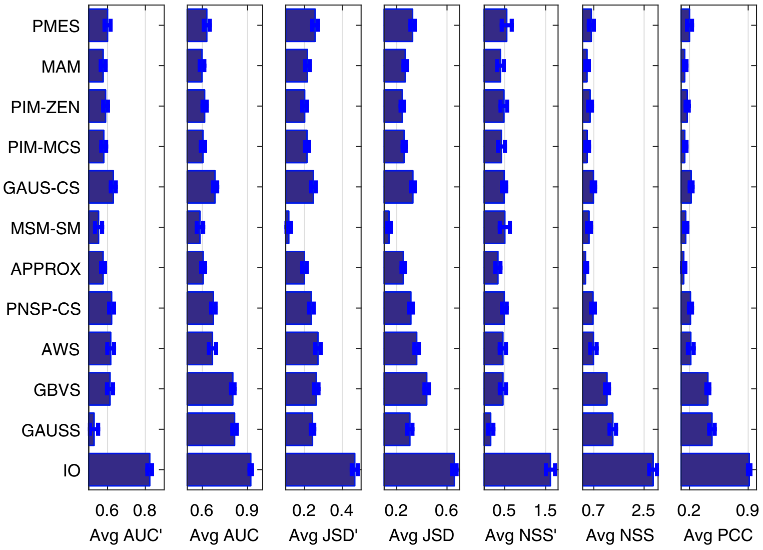

Fig. 8 Evaluation of models depending on MPEG-4 ASP video bitstream using various metrics

revealed by comparing AUC and NSS scores to their center bias-corrected versions AUC′ and NSS′. For example, the AUC measures the accuracy of saliency prediction of a particular model against a control distribution drawn uniformly across the frame. Since the uniform distribution is a relatively poor control distribution for saliency and easy to outperform, all models achieve a higher AUC score compared to their AUC′ score, which uses a control distribution fitted to the empirical gaze points shown in Fig. 3. This effect is most visible in the GAUSS benchmark model, which has the AUC score of around 0.8 (higher than all the models except IO), but the AUC′ score of only slightly above 0.5 (lower than all other models). This over-exaggeration of the accuracy of a simple scheme such as GAUSS when plain AUC used was the reason why [40, 72] suggest center bias correction via non uniform control sampling. The center bias-corrected AUC′ score is a better reflection of the models' performance. Center bias also has a significant effect on NSS, but a less pronounced effect on JSD. It can also be observed that GAUSS (and then GBVS) achieves a higher PCC score than any other method except IO, due to the accumulation of fixations near the center of the frame.

Thus far, we showed the results for saliency models that accept MPEG-4 ASP encoded bitstream. The accuracy assessment according to AUC′ and NSS′ over the saliency models that accept H.264/AVC-encoded data is shown in Fig. 9.[5] Two recent compressed-domain methods, MVE+SRN and OBDL-MRF, top all other methods, including pixel-domain ones, on both metrics. Based on these results, MVE+SRN seems like the best saliency predictor, while OBDL-MRF comes a closes second. Both of these models have been built upon compressed-domain features that are highly correlated with human gaze, and are therefore able to compete even with high-performing pixel-domain models such as GBVS and AWS.

In addition to the average scores, another type of assessment of a model's performance is counting its number of appearances among top performing models for each sequence [65].

[5] The same results according to JSD′ and PCC are shown in supplementary material [42].

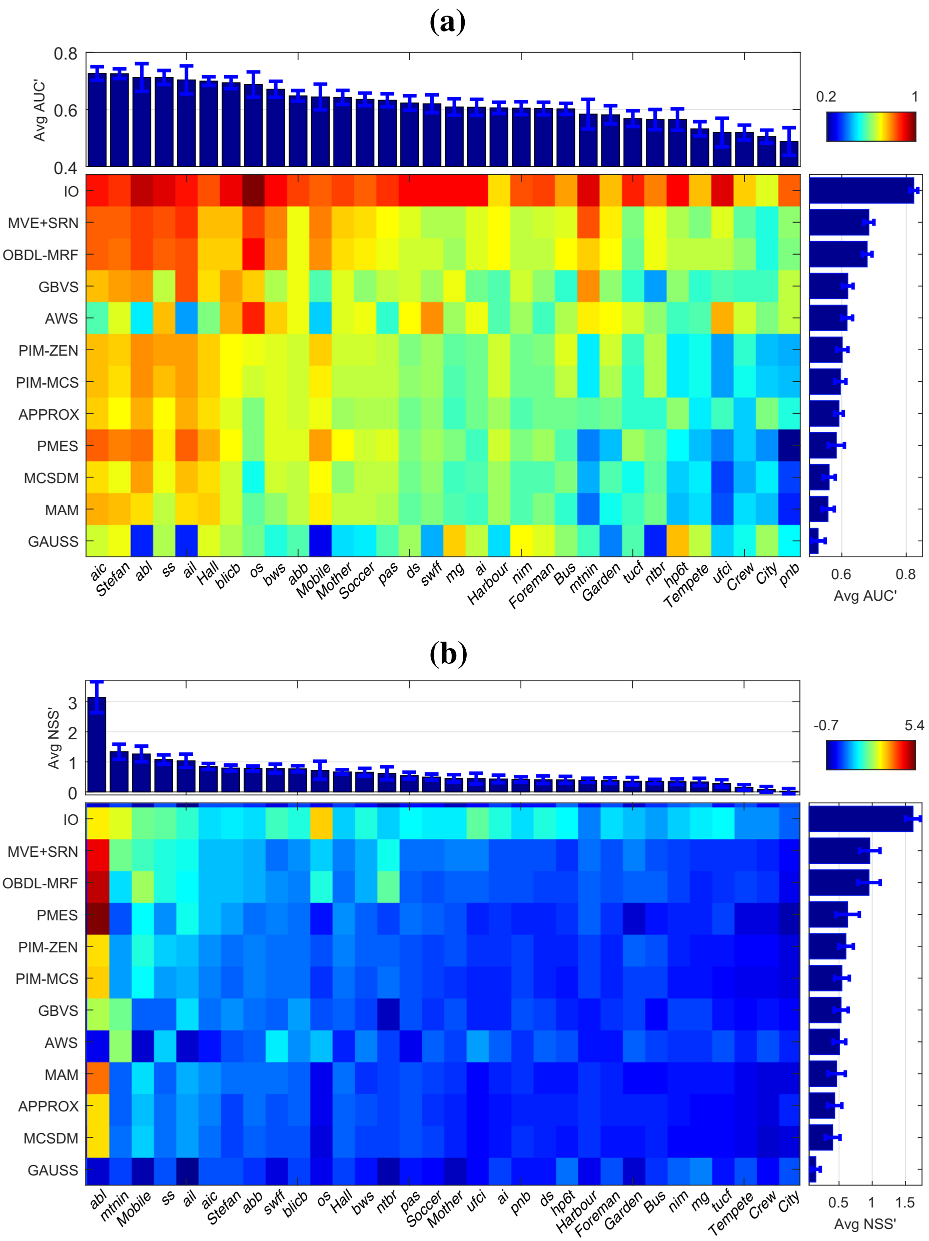

Fig. 9 Accuracy of various saliency models over H.264/AVC encoded bitstream of SFU and DIEM dataset according to (**a**) AUC′ and (**b**) NSS′. The *2D color map* shows the average score of each model on each sequence. *Top*: Average score for each sequence, across all models. *Right*: Average scores each model across all sequences. *Error bars* represent standard error of the mean (SEM), $\sigma/\sqrt{n}$, where σ is the sample standard deviation of n samples

To this end, a multiple comparison test is performed using Tukey's honestly significant difference as the criterion [31]. Specifically, for each sequence, we compute the average

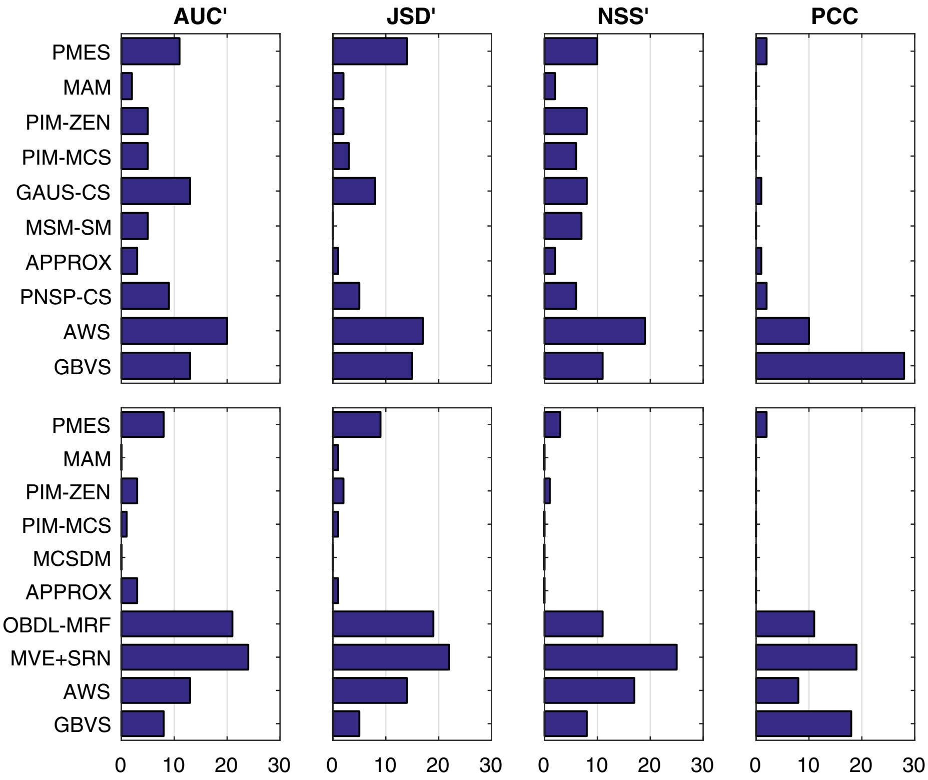

Fig. 10 The number of appearances among top performers, using various evaluation metrics. Results based on MPEG-4 ASP are shown at the *top*, those based on H.264/AVC at the *bottom*

score of a model across all frames, as well as the 95 % confidence interval for the average score. Then we find the model with the highest average score (excluding IO), and find all the models whose 95 % confidence interval overlaps that of the highest-scoring model. All such models are considered top performers for the given sequence. The number of appearances among top performers for each model is shown in Fig. 10. These results show similar trends as average scores, with MVE+SRN, OBDL-MRF, AWS, GBVS, PMES, GAUS-CS and PNSP-CS often being among top performers, while MCSDM, MAM and APPROX rarely offering top scores.

4.3 Sensitivity to compression

In the assessments presented thus far, the QP value was set to a constant value (in MPEG-4 ASP, QP = 16 and in H.264/AVC, QP = 36). The quality of encoded video drops as the QP increases due to the larger amount of compression. Figure 11 shows how the average AUC′ score changes as a function of the average PSNR by (a) varying QP∈{1, 4, 7, ..., 31} for MPEG-4 ASP and (b) varying QP∈{3, 6, 9, ..., 51} for H.264/AVC.[6] The results in Fig. 11

[6] The relationship between the average NSS′ score and the average PSNR is provided in the supplementary material [42].

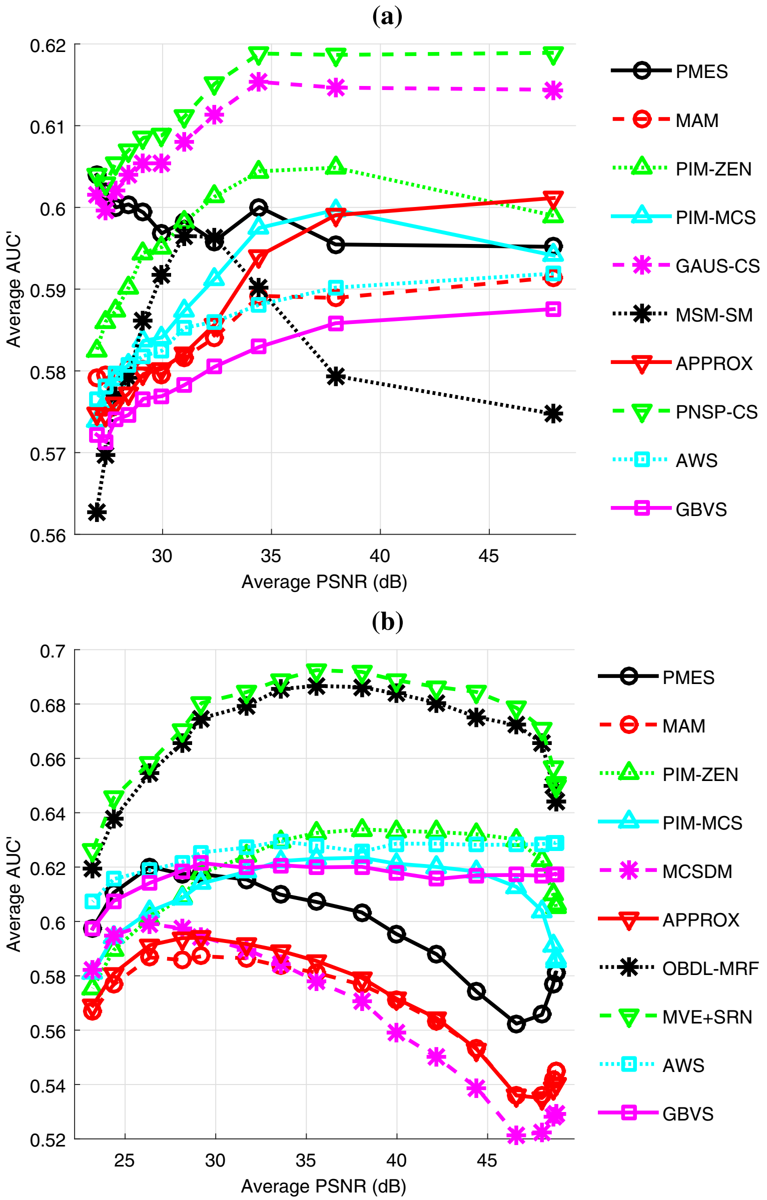

Fig. 11 The relationship between the average PSNR and the models' accuracy over SFU and DIEM dataset. (**a**) the sensitivity over MPEG-4 ASP encoded bitstream, (**b**) the sensitivity over H.264/AVC encoded bitstream

indicate the sensitivity of the models' saliency prediction relative to encoding parameters. In this experiment, AWS and GBVS were applied to the decoded video, hence they effectively used the same data as compressed-domain models, but in the pixel domain after full video reconstruction.

Table 5 Average processing time in milliseconds per frame

Model	AWS	GBVS	MAM	PMES	GAUS-CS	PNSP-CS	PIM-ZEN	OBDL-MRF	MVE+SRN	APPROX	MCSDM	PIM-MCS	MSM-SM
MPEG-4	1650	864	155	94	64	64	20			12		9	7
H.264	1468	806	600	477			42	35	18	16	8	5	

The figure shows that pixel-domain models GBVS and AWS score lower at low video qualities, while their accuracy improves as the video quality increases. Their accuracy is fairly consistent beyond a certain level of video quality, around 35 dB, suggesting that so long as video quality is sufficiently high, compression does not affect the models' ability to estimate saliency. This observation is consistent with studies undertaken by Le Meur [51], and Milanfar and Kim [47].

Compressed-domain models exhibit a somewhat different behavior. Their accuracy is also generally low at low video qualities, because MVs are less accurate and there is a large amount of quantization noise present in prediction residuals. But unlike pixel-domain models, compressed-domain models also seem to suffer at high video qualities. As the quality increases, compressed domain features become less informative. Small quantization step size makes most transform coefficients non-zero, which makes some of the models predict high spatial saliency throughout the frame. At the same time, MVs may become too noisy, since rate-distortion optimization does not impose sufficient constraints on motion estimation. The results suggest that the PSNR range in which most compressed-domain saliency models tend to be most accurate is 30-40 dB, which also happens to be a range in which a good trade-off is thought to be achieved between video quality and the required bitrate.

4.4 Complexity

The average processing time per frame on the SFU dataset (CIF resolution videos at 30 fps) using two different input formats is listed in Table 5. The time taken for extracting MVs and DCT values from the bitstream is excluded. Please note that these results correspond to MATLAB implementations of the models and the processing time can be significantly decreased by implementation in a low-level programming language such as C/C++. Despite this, some of the models are fast enough for real time performance (under 33 ms per frame) even when implemented in MATLAB. Discussion of accuracy and complexity of the models is presented in the next section.

5 Discussion

Considering the results in Figs. 8 and 10, MVE+SRN, OBDL-MRF, AWS, GBVS, PMES and GAUS-CS consistently achieve high scores across different metrics. It is encouraging that the performance of some compressed-domain models is superior to that of high-performing pixel-domain models. Note that, in general, achieving a high score with one metric does not guarantee a high score with other metrics. As an example, MSM-SM achieves a relatively high average scores across several metrics, but the lowest JSD and JSD$'$ score. Hence, the fact that MVE+SRN, OBDL-MRF, AWS, GBVS, PMES and GAUS-CS perform consistently well across all metrics considered in this study lends additional confidence in their accuracy.

PMES was the first compressed-domain saliency model, proposed in 2001, and it only uses MVs to estimate saliency. It is well known that motion is a strong indicator of saliency in dynamic visual scenes [34, 62, 66], so it is not surprising that MVs would be a powerful cue for saliency estimation. PMES estimates saliency by considering two properties: large motion magnitude in a spatio-temporal region, and the lack of coherence among MV angles in that region. These two properties seem to describe salient objects reasonably well in most cases, as demonstrated by the results. Taken together, they resemble a center-surround mechanism where a region is considered salient if it sufficiently "stands out" from its surroundings.

GAUS-CS and PNSP-CS show high performance in both I- and P-frames. Both models are based on the center-surround difference mechanism, and both employ MVs for saliency estimation in P-frames and DCT of pixel values in I-frames. The capability of center-surround difference mechanism to predict where people look has been discussed extensively [37], so their success is also not surprising.

Although PIM-MCS and MSM-SM also attempt to employ the center-surround difference mechanism, their scores are not as consistently high as those of GAUS-CS and PNSP-CS. The reason may be that in GAUS-CS and PNSP-CS models, the contrast is inversely proportional to the distance between the current DCT block and all other DCT blocks in the frame, which means that they consider not only the contrast between blocks, but also the distance between them. This seems to be a good strategy for compressed-domain saliency estimation.

OBDL-MRF and MVE-SRN are two of the most recent compressed-domain saliency models. Taking advantage of the availability of gaze point data for video, which was not the case when earliest models such as PMES were developed, both OBDL-MRF and MVE-SRN were built upon compressed-domain features that have been shown to be highly correlated with gaze points in video. Their advantage over other compressed-domain models is therefore not surprising. What is perhaps surprising is their ability to go toe-to-toe with the best pixel-domain models, and be more accurate in many cases. Their success lends further support to the hypotheses that relate saliency to compressibility [8, 34], although their operational realization is quite different from these earlier works.

According to the results in the previous section, the lowest-scoring models on most metrics were APPROX and MCSDM. Incidentally, APPROX was originally developed for a different type of input data and had to be modified for this comparison, which may have had a negative impact on its performance.

The influence of global (camera) motion on visual saliency is still a fairly open research problem, with limited work in the literature addressing this issue. Reference [1] studied separately the effect of pan/tilt and zoom-in/-out. It was found that in the case of pan/tilt, the gaze points tend to shift towards the direction of pan/tilt, in the case of zoom-in, they tend to concentrate near the frame center, and in the case of zoom-out, they tend to scatter further out. On the other hand, according to [4], the presence of camera motion tends to concentrate gaze points around the center of the frame "according to the direction orthogonal to the tracking speed vector."

Among the models tested in the present study, only APPROX took global motion into account by removing it prior to the analysis of MVs. This paid off in the case of *City*, which was overall the most difficult sequence for other spatio-temporal saliency models in Figs. 6 and 7. However, global motion compensation (GMC) did not help much in the case of *Tempete* or *Flower Garden*. In fact, *Tempete* contains strong zoom-out, which, according to [1], would tend to scatter the gaze points around the frame. However, Figs. 6 and 7 show that GAUSS, with its simple center-biased saliency map, scores well here (even with center-bias-corrected metrics), suggesting that the gaze points are still located near the center of the frame. This is due to the presence of a yellow bunch of flowers in the center of the frame, which turns out to be highly attention-grabbing. Apparently, the key to accurate saliency estimation in *Tempete* is not in the motion, but rather in the color present in the scene. *Flower Garden* is another example where GMC did not pay off. The viewers' gaze in this sequence is attracted to the objects in the background, specifically the windmill and the pedestrians, whose motion tends to be zeroed out after GMC on 8 × 8 MVs. Overall, the results suggest that global motion is not sufficiently well handled by current compressed-domain methods, and that further research is needed to make progress on this front.

Considering models' complexity and processing time in Table 5, MSM-SM, PIM-MCS and MCSDM are the fastest while AWS is the most demanding. Note that the smallest block size is 4 × 4 in H.264/AVC encoded bitstream and 8 × 8 in MPEG-4 ASP encoded bitstream, and therefore more data typically needs to be processed in the H.264/AVC case, which is why compressed-domain models that are able to accept both input formats tend to take more time when applied on H.264/AVC bitstreams. MSM-SM, PIM-MCS and MCSDM are the least complex models, but unfortunately not the most accurate.

While MVE+SRN and OBDL-MRF scored the highest in terms of accuracy, this did not come at a cost of high complexity. In fact, according to complexity, they are in the middle of the pack, with processing times below those of other high-performing saliency models. MVE+SRN appears twice as fast as OBDL-MRF because entropy decoding time of MVs and DCT residuals was not taken into account (as with other compressed-domain models). But OBDL-MRF does not require any such decoding and would therefore likely end up being faster in a real-world scenario.

6 Conclusions

In this study we attempted to provide a comprehensive comparison of eleven compressed-domain visual saliency models for video. All methods were reimplemented in MATLAB and tested on two eye-tracking datasets using several accuracy metrics. Care was taken to correct for center bias and border effects in the employed metrics, which were issues found in earlier studies on visual saliency model evaluation. Pixel-domain saliency models have the potential for higher accuracy compared to compressed-domain saliency models, but also have higher computational complexity. In contrast to the pixel-domain methods, compressed-domain approaches make use of the data from the compressed video bitstream, such as motion vectors, block coding modes, motion-compensated prediction residuals or their transform coefficients, etc. The lack of full pixel information often leads to lower accuracy, but the main advantage of compressed-domain methods in practical applications is their generally lower computational cost. This is due to the fact that part of decoding can be avoided, a smaller amount of data needs to be processed compared to pixel-domain methods, and some of the information produced during encoding (e.g., motion vectors and transform coefficients) can be reused. Therefore, compressed-domain methods are more suitable for real-time applications.

The results of the study indicate that reasonably accurate visual saliency estimation is possible using only a limited set of data from the compressed bitstream, such as motion vectors, prediction residuals, or even just the number of bits per block, without further decoding. Several compressed-domain saliency models showed competitive accuracy with some of the best currently known pixel-domain models. On top of that, some of the compressed-domain methods are fast enough for real-time saliency estimation on CIF video even with a relatively inefficient MATLAB implementation, which suggests that their optimized implementation could be used for online saliency estimation in a variety of applications, even for higher-resolution video.

Many sequences that have turned out to be difficult for models to handle contain global (camera) motion. The influence of global motion on visual saliency is not very well understood, and most models in the study did not account for it. A number of compressed-domain global motion estimation methods, based on motion vectors alone, have been developed recently, so it is reasonable to expect that compressed-domain saliency models should be able to benefit from these developments.

Acknowledgments This work was supported in part by the Cisco Research Award CG# 573690 and NSERC Grant RGPIN 327249.

References

1. Abdollahian G, Pizlo Z, Delp E (2008) A study on the effect of camera motion on human visual attention. In: Proc. IEEE ICIP'08, pp 693–696
2. Adaptive whitening saliency model (AWS). http://persoal.citius.usc.es/xose.vidal/research/aws/AWSmodel.html. [Online]
3. Agarwal G, Anbu A, Sinha A (2003) A fast algorithm to find the region-of-interest in the compressed MPEG domain. In: Proc. IEEE ICME'03, vol 2, pp 133–136
4. Baudrier E, Rosselli V, Larabi M (2009) Camera motion influence on dynamic saliency central bias. In: Proc. IEEE ICASSP'09, pp 817–820
5. Borji A, Itti L (2013) State-of-the-art in visual attention modeling. IEEE Trans Pattern Anal Mach Intell 35(1):185–207
6. Borji A, Sihite DN, Itti L (2013) Quantitative analysis of human-model agreement in visual saliency modeling: a comparative study. IEEE Trans Image Process 22(1):55–69
7. Brown CD, Davis HT (2006) Receiver operating characteristics curves and related decision measures: A tutorial. Chemometrics Intell Lab Syst 80(1):24–38
8. Bruce N, Tsotsos J (2006) Saliency based on information maximization. Adv Neural Inf Process Syst 18:155
9. Carlson NR (2010) Psychology: the science of behaviour, 4th edn. Pearson Education Canada
10. Culibrk D, Mirkovic M, Zlokolica V, Pokric M, Crnojevic V, Kukolj D (2011) Salient motion features for video quality assessment. IEEE Trans Image Process 20(4):948–958
11. Dagan I, Lee L, Pereira F (1997) Similarity-based methods for word sense disambiguation. In: Proceedings of the eighth conference on European chapter of the association for computational linguistics. Association for Computational Linguistics, pp 56–63
12. DIEM dataset. http://thediemproject.wordpress.com. [Online]
13. Efron B, Tibshirani R (1993) An introduction to the bootstrap, vol 57. CRC Press
14. Einhäuser W, Spain M, Perona P (2008) Objects predict fixations better than early saliency. J Vis 8(14)
15. Engelke U, Kaprykowsky H, Zepernick HJ, Ndjiki-Nya P (2011) Visual attention in quality assessment. IEEE Signal Process Mag 28(6):50–59
16. Fang Y, Chen Z, Lin W, Lin CW (2012) Saliency detection in the compressed domain for adaptive image retargeting. IEEE Trans Image Process 21(9):3888–3901
17. Fang Y, Lin W, Chen Z, Tsai CM, Lin CW (2012) Video saliency detection in the compressed domain. In: Proc. 20th ACM intl. conf. multimedia. ACM, pp 697–700
18. Fang Y, Lin W, Chen Z, Tsai CM, Lin CW (2014) A video saliency detection model in compressed domain. IEEE Trans Circ Syst Video Technol 24(1):27–38
19. Feng X, Liu T, Yang D, Wang Y (2011) Saliency inspired full-reference quality metrics for packet-loss-impaired video. IEEE Trans Broadcast 57(1):81–88
20. FFMPEG project. http://www.ffmpeg.org. [Online]
21. Fukuchi K, Miyazato K, Kimura A, Takagi S, Yamato J (2009) Saliency-based video segmentation with graph cuts and sequentially updated priors. In: Proc. IEEE ICME'09, pp 638–641
22. Guo C, Zhang L (2010) A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. IEEE Trans Image Process 19(1):185–198
23. Hadizadeh H (2013) Visual saliency in video compression and transmission. Ph.D. thesis, Simon Fraser University
24. Hadizadeh H, Bajić IV (2011) Saliency-preserving video compression. In: Proc. IEEE ICME'11, pp 1–6

25. Hadizadeh H, Bajić IV (2014) Saliency-aware video compression. IEEE Trans Image Process 23(1):19–33
26. Hadizadeh H, Enriquez MJ, Bajić IV (2012) Eye-tracking database for a set of standard video sequences. IEEE Trans Image Process 21(2):898–903
27. Hagiwara A, Sugimoto A, Kawamoto K (2011) Saliency-based image editing for guiding visual attention. In: Proc. PETMEI'11, pp 43–48
28. Han S, Vasconcelos N (2010) Biologically plausible saliency mechanisms improve feedforward object recognition. Vis Res 50(22):2295–2307
29. Harel J, Koch C, Perona P (2007) Graph-based visual saliency. Adv Neural Inf Process Syst 19:545–552
30. He Z, Bystrom M, Nawab SH (2005) Bidirectional conversion between DCT coefficients of blocks and their subblocks. IEEE Trans Signal Process 53(8):2835–2841
31. Hochberg Y, Tamhane AC (1987) Multiple comparison procedures. Wiley
32. Itti L (2004) Automatic foveation for video compression using a neurobiological model of visual attention. IEEE Trans Image Process 13(10):1304–1318
33. Itti L, Baldi P (2005) A principled approach to detecting surprising events in video. In: IEEE Computer society conference on computer vision and pattern recognition (CVPR'05), vol 1, pp 631–637
34. Itti L, Baldi P (2009) Bayesian surprise attracts human attention. Vis Res 49(10):1295–1306
35. Itti L, Baldi PF (2006) Bayesian surprise attracts human attention. Adv Neural Inf Process Syst 19:547–554
36. Itti L, Carmi R (2009) Eye-tracking data from human volunteers watching complex video stimuli. doi:10.6080/K0TD9V7F. CRCNS.org
37. Itti L, Koch C, Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. IEEE Trans Pattern Anal Mach Intell 20(11):1254–1259
38. Jeffreys H (1946) An invariant form for the prior probability in estimation problems. Proc R Soc London Series A Math. Phys. Sci. 186(1007):453–461
39. Ji QG, Fang ZD, Xie ZH, Lu ZM (2013) Video abstraction based on the visual attention model and online clustering. Signal Process Image Commun 28(3):241–253
40. Kanan C, Tong MH, Zhang L, Cottrell GW (2009) SUN: top-down saliency using natural statistics. Vis Cogn 17(6–7):979–1003
41. Khatoonabadi SH, Bajić IV (2013) Video object tracking in the compressed domain using spatio-temporal Markov random fields. IEEE Trans Image Process 22(1):300–313
42. Khatoonabadi SH, Bajić IV, Shan Y Supplementary material for compressed-domain saliency evaluation. http://geomm.ensc.sfu.ca/software/CSE-supplementary
43. Khatoonabadi SH, Bajić IV, Shan Y (2014) Comparison of visual saliency models for compressed video. In: Proc. IEEE ICIP'14, pp 1081–1085
44. Khatoonabadi SH, Bajić IV, Shan Y (2014) Compressed-domain correlates of fixations in video. In: ACM multimedia PIVP, pp 3–8
45. Khatoonabadi SH, Bajić IV, Shan Y (2015) Compressed-domain correlates of human fixations in dynamic scenes. Multimed Tools Appl. doi:10.1007/s11042-015-2802-3. To appear
46. Khatoonabadi SH, Vasconcelos N, Bajić IV, Shan Y (2015) How many bits does it take for a stimulus to be salient? In: Proc. IEEE CVPR'15, pp 5501–5510
47. Kim C, Milanfar P (2013) Visual saliency in noisy images. J Vis 13(4):1–14
48. Kullback S (1997) Information theory and statistics. Courier Dover Publications
49. Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22(1):79–86
50. Larson EC, Vu C, Chandler DM (2008) Can visual fixation patterns improve image fidelity assessment? In: Proc. IEEE ICIP'08, pp 2572–2575
51. Le Meur O (2011) Robustness and repeatability of saliency models subjected to visual degradations. In: Proc. IEEE ICIP'11, pp 3285–3288
52. Le Meur O, Baccino T (2013) Methods for comparing scanpaths and saliency maps: Strengths and weaknesses. Behav Res Methods 45(1):251–266
53. Li Z, Qin S, Itti L (2011) Visual attention guided bit allocation in video compression. Image Vis Comput 21(1):1–19
54. Lin J (1991) Divergence measures based on the Shannon entropy. IEEE Trans Inf Theory 37(1):145–151
55. Lin X, Ma H, Luo L, Chen Y (2012) No-reference video quality assessment in the compressed domain. IEEE Trans Consum Electron 58(2):505–512
56. Liu H, Heynderickx I (2011) Visual attention in objective image quality assessment: based on eye-tracking data. IEEE Trans Circuits Syst Video Technol 21(7):971–982

57. Liu Z, Yan H, Shen L, Wang Y, Zhang Z (2009) A motion attention model based rate control algorithm for H. 264/AVC. In: The 8th IEEE/ACIS International conference on computer and information science (ICIS'09), pp 568–573
58. Locarna systems. http://www.locarna.com. [Online]
59. Lu T, Yuan Z, Huang Y, Wu D, Yu H (2010) Video retargeting with nonlinear spatial-temporal saliency fusion. In: Proc. IEEE ICIP'10, pp 1801–1804
60. Ma YF, Zhang HJ (2001) A new perceived motion based shot content representation. In: Proc. IEEE ICIP'01, vol 3, pp 426–429
61. Ma YF, Zhang HJ (2002) A model of motion attention for video skimming. In: Proc. IEEE ICIP'02, vol 1, pp 129–132
62. Mahadevan V, Vasconcelos N (2010) Spatiotemporal saliency in dynamic scenes. IEEE Trans Pattern Anal Mach Intell 32(1):171–177
63. Mahadevan V, Vasconcelos N (2013) Biologically inspired object tracking using center-surround saliency mechanisms. IEEE Trans Pattern Anal Mach Intell 35(3):541–554
64. Mateescu VA, Bajić IV (2013) Guiding visual attention by manipulating orientation in images. In: Proc. IEEE ICME'11, pp 1–6
65. Mateescu VA, Hadizadeh H, Bajić IV (2012) Evaluation of several visual saliency models in terms of gaze prediction accuracy on video. In: Proc. IEEE Globecom'12 Workshop: QoEMC, pp 1304–1308
66. Milanese R, Gil S, Pun T (1995) Attentive mechanisms for dynamic and static scene analysis. Opt Eng 34(8):2428–2434
67. Mishra AK, Aloimonos Y, Cheong LF, Kassim A (2012) Active visual segmentation. IEEE Trans Pattern Anal Mach Intell 34(4):639–653
68. Mital PK, Smith TJ, Hill RL, Henderson JM (2011) Clustering of gaze during dynamic scene viewing is predicted by motion. Cogn Comput 3(1):5–24
69. Moorthy AK, Bovik AC (2009) Visual importance pooling for image quality assessment. IEEE J Sel Topics Signal Process 3(2):193–201
70. MPEG-7 Output Document ISO/MPEG: MPEG-7 visual part of experimentation model (XM) version 2.0 (1999)
71. Muthuswamy K, Rajan D (2013) Salient motion detection in compressed domain. IEEE Signal Process Lett 20(10):996–999
72. Parkhurst DJ, Niebur E (2003) Scene content selected by active vision. Spatial Vis 16(2):125–154
73. Peters RJ, Iyer A, Itti L, Koch C (2005) Components of bottom-up gaze allocation in natural images. Vis Res 45(18):2397–2416
74. Schisterman EF, Faraggi D, Reiser B, Trevisan M (2001) Statistical inference for the area under the receiver operating characteristic curve in the presence of random measurement error. Amer J Epidemiol 154(2):174–179
75. Sinha A, Agarwal G, Anbu A (2004) Region-of-interest based compressed domain video transcoding scheme. In: Proc. IEEE ICASSP'04, vol 3, pp 161–164
76. SR Research. http://www.sr-research.com. [Online]
77. Swets JA (1996) Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers. Lawrence Erlbaum Associates Inc
78. Tatler BW (2007) The central fixation bias in scene viewing: selecting an optimal viewing position independently of motor biases and image feature distributions. J Vis 7(14)
79. Tatler BW, Baddeley RJ, Gilchrist ID (2005) Visual correlates of fixation selection: effects of scale and time. Vis Res 45(5):643–659
80. Vandewalle P, Kovacevic J, Vetterli M (2009) Reproducible research in signal processing. IEEE Signal Process Mag 26(3):37–47
81. Wang Z, Bovik AC (2005) Foveated image and video coding. In: Wu HR, Rao BD (eds) Digital video image quality and perceptual coding, Marcel Dekker, pp 431–458
82. Wang Z, Li Q (2011) Information content weighting for perceptual image quality assessment. IEEE Trans Image Process 20(5):1185–1198
83. Wiegand T, Sullivan GJ, Bjontegaard G, Luthra A (2003) Overview of the H. 264/AVC video coding standard. IEEE Trans Circuits Syst Video Technol 13(7):560–576
84. Winkler S, Subramanian R (2013) Overview of eye tracking datasets. In: 5th Intl. Workshop on quality of multimedia experience (QoMEX). IEEE, pp 212–217
85. Xie R, Yu S (2008) Region-of-interest-based video transcoding from MPEG-2 to H. 264 in the compressed domain. Opt Eng 47(9):097,001–097,001
86. Zhang L, Tong MH, Marks TK, Shan H, Cottrell GW (2008) SUN: a bayesian framework for saliency using natural statistics. J Vis 8(7)

Sayed Hossein Khatoonabadi received the B.Sc. degree in computer science from Kerman University, Kerman, Iran, and the M.S. degree in computer engineering from the Amirkabir University of Technology, Tehran, Iran, in 2002 and 2005, respectively. He was awarded a PhD degree by Multimedia Communications Lab at Simon Fraser University, Burnaby, BC, Canada, in 2015. His research spans selected areas in image and video processing, multimedia communication and machine learning.

Ivan V. Bajić received the B.Sc.Eng. degree (summa cum laude) in electronic engineering from the University of Natal, Durban, South Africa, in 1998, and the M.S. degree in electrical engineering, the M.S. degree in mathematics, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2000, 2002, and 2003, respectively. He is currently Associate Professor of Engineering Science at Simon Fraser University, Burnaby, BC, Canada. His research interests include signal, image, and video processing and compression, multimedia ergonomics, and communications. He has authored about a dozen and co-authored another eight dozen publications in these fields. He has served on the organizing and program committees of various conferences in the field, including GLOBECOM, ICC, ICME, and ICIP, and was the Chair of the Media Streaming Interest Group of the IEEE Multimedia Communications Technical Committee from 2010 to 2012. He is currently serving as an Associate Editor of the IEEE Signal Processing Magazine and the Chair of the Vancouver Chapter of the IEEE Signal Processing Society.

Yufeng Shan received the B.E. and M.E. degrees in mechanical and computer engineering from Shandong University, Shandong, China, in 1991 and 1994, respectively, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2007. From 1995 to 2000, he was an Assistant Professor at the Department of Computer Science, Shandong University. In 2000, he held the position of Exterior Collaborate Professor with the PLANTE group, INRIA Sophia Antipolis, France. From 2001 to 2002, he was a Research Scientist at the Department of Electrical Engineering and Computer Science, University of California, Berkeley. From 2003 to 2007, he was with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, first as a Research Scientist and then as a Ph.D. student. He is currently with Cisco Systems, Inc., Boxborough, MA. His research interests include multimedia communications, networking, and high-performance distributed systems.