COMPARISON OF VISUAL SALIENCY MODELS FOR COMPRESSED VIDEO

Sayed Hossein Khatoonabadi, Ivan V. Bajić
Simon Fraser University, Burnaby, BC, Canada
skhatoon@sfu.ca, ibajic@ensc.sfu.ca

Yufeng Shan
Cisco Systems, Boxborough, MA, USA
yshan@cisco.com

ABSTRACT
Visual saliency modeling is an increasingly important research problem. While most saliency models for dynamic scenes operate on raw video, several models have also been developed for compressed video. This paper compares the accuracy of nine such models on a common eye-tracking dataset. The results indicate that a reasonably accurate saliency estimation is possible even using only motion vectors from the compressed bitstream. Successful strategies in compressed-domain saliency modeling are highlighted, and certain challenges are identified for future improvement.

Index Terms—Visual saliency, compressed video

1. INTRODUCTION
Visual saliency models find a number of applications in image and video processing, such as quality assessment [1–4], compression [5–8], retargeting [9, 10], and so on. 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 on devices with restricted processing and memory, such as mobile devices. One way to reduce the computational cost of saliency estimation is to use compressed-domain features. This way, part of decoding can be avoided, a smaller amount of data needs to be processed, and some of the information produced during encoding (e.g., MVs and transform coefficients) can be reused. Compressed-domain algorithms for visual saliency estimation have been developed for various applications such as retargeting [9], video transcoding [11–13], quality estimation [14], video retrieval [15], video skimming [16], etc. 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.

Existing compressed-domain 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 fully reported. All of this makes a fair 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 a common ground-truth dataset, similar to what has been done in [17] for pixel-domain models. 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; the MATLAB code and data necessary to reproduce all the reported results are available online at http://mcl.ensc.sfu.ca/software/CSE-ICIP2014.rar.

2. MODELS AND DATA

2.1. Compressed-Domain Visual Saliency Models
Saliency models considered in this study operate on the information found in a compressed video bitstream, such as block-based MVs, prediction residuals or their transforms, block coding modes, etc. During 2013 we surveyed the literature on the topic and found nine 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.

Models presented in [12,15,16] utilize only MVs from the compressed bitstream, whereas others also make use of transform coefficients of pixels (in I-frames) or prediction residuals (in P- or B-frames). Models in [15,16,18,20,22] treat MV magnitude as an indicator of motion saliency, whereas other models rely on center-surround difference between MV and its neighbors as an indicator of motion saliency. Those models that make use of transform coefficients often treat low- and
mid-frequency coefficients as indicative of spatial saliency. Readers are referred to corresponding publications for further details.

We also considered a benchmark model referred to as Gaussian center-bias (GAUSS). The GAUSS saliency map is just a 2D Gaussian blob with standard deviation of 1° of visual angle, located at the center of the frame. This model assumes that the center of the 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.

### 2.2. Eye-Tracking Data

Eye-tracking data is the most widely used psychophysical ground truth for visual saliency models [1]. To evaluate a model, its saliency map is compared with the recorded gaze locations of the subjects. In this study, we use the SFU dataset [21], which contains gaze data of 15 viewers for twelve CIF (352 × 288) sequences (Foreman, Flower Garden, etc.) that have become popular in the video compression and communications community. Gaze points of the first viewing are used as the ground truth.

### 3. EVALUATION FRAMEWORK

We implemented all saliency models in MATLAB. Where possible, we verified the implementation by comparing the results with those presented in the corresponding papers and/or by contacting the authors. All models (except benchmarks) rely on compressed-domain data, but some assumed different coding standards. Since seven out of nine models could accommodate MPEG-4 ASP data by default, we decided to encode all videos in this format, and made minor modifications to MCSDM [12] and APPROX [20] by changing the minimum block size from 4 × 4 to 8 × 8. For encoding, the IPPP GOP structure was used with the GOP size of 12. Since only GAUS-CS and PNSP-CS out of all nine models considered B-frames, we did not employ B-frames in the evaluation. Unless otherwise stated, the QP value was set to 12 and MVs have 1/4-pel accuracy with no range restriction. After encoding, the DCT-P values (in I-frames) and DCT-R values (in P-frames), as well as MVs (in P-frames) were extracted for each 8 × 8 block and fed to the models. Encoding and partial decoding to extract the required data was done using the FFmpeg library (www.ffmpeg.org).

We employed the well-known Area Under receiver operating characteristic Curve (AUC) [25] to evaluate the accuracy of various models. The AUC is in the range [0, 1]; the larger the value, the better the correspondence between the saliency map and ground-truth gaze data. The value of 0.5 represents pure chance performance.

To generate the control set for AUC computation, the nogaze saliency values are usually sampled according to some distribution [26]. It turns out that the sampling distribution has a significant effect on the resulting score. For example, Kanan et al. [27] and Borji et al. [17] showed that a static Gaussian blob at the center of the frame (i.e., our GAUSS benchmark) may result in artificially high scores when AUC is computed using a uniform spatial distribution for the control samples. Additionally, Zhang et al. [28] showed that adding dummy zero saliency values at the border of the image changes the distribution of saliency of the random samples, leading to different scores for the same saliency map.

To decrease the influence of center bias and border effects, Tatler et al. [29] and Parkhurst and Niebur [30] suggested to distribute control samples non-uniformly, according to the measured gaze points. In this study, we use both the uniform sampling distribution, resulting in the score called AUC, and the non-uniform sampling distribution obtained by fitting a 2D Gaussian to the empirical gaze points across the whole dataset, which results in the score referred to as AUC′.

### 4. RESULTS AND DISCUSSION

A number of assessments of the models from Table 1 were carried out. Fig. 1 shows the average AUC and AUC′ scores of various models across the test sequences. One interesting point in Fig. 1 is an indication of how difficult or easy is saliency prediction in a given sequence. In the figure, the sequences are sorted along the horizontal axis in decreasing order of average score across all models. Although the order based on AUC is not exactly the same as that based on AUC′, overall, it seems that Stefan is the sequence where the saliency
is easiest to predict. Stefan contains a single salient moving object (tennis player) whose motion is sufficiently strong and different from the surroundings that almost all models are able to correctly predict viewers’ gaze locations. On the other hand, City and Tempeste seem to be the most challenging for saliency prediction. City does not contain any moving objects, only global (camera) motion. Tempeste also contains significant global motion (zoom out) and in addition shows falling yellow leaves that act like motion noise, as they do not attract viewers’ attention.

The effect of center bias is easily revealed by comparing AUC scores to the center bias-corrected AUC’ scores. This effect is most visible in the GAUSS benchmark model, which has the AUC score of around 0.8 (higher than all other models), but the AUC’ score just over 0.5, worse than other models. The center bias-corrected AUC’ score is a better reflection of the models’ performance [29, 30]. It is encouraging that all models achieve higher average AUC scores to the center bias-corrected AUC.

Another way to rank the models’ performance is via a multiple comparison test [31]. For each sequence, we compute the average score and its 95% confidence interval of a given model across all frames. Then we find the model with the highest average score, 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. Fig. 2 shows two examples. In the left panel, GAUSS has the highest average AUC’ score (solid blue circle) and its 95% confidence interval does not overlap any of the other models’ intervals. Hence, in this case, GAUSS is the sole top performer. On the other hand, in the right panel, the 95% confidence interval of the top-scoring PMES (solid blue circle) overlaps the corresponding intervals of MCSDM, GAUSS-CS, PNSP-CS and MSM-SM (black circles). In this case, all five models are considered top performers.

Model ranking in terms of the number of appearances among top performers is shown in Fig. 3. According to AUC’ in Fig. 3 (right), four methods (PMES, GAUS-CS, MSM-SM, and PIM-ZEN) achieve equal or higher ranking compared to GAUSS. Again, this illustrates that content-dependent information from the compressed bitstream can be quite useful for saliency prediction. Note that PMES, the top-ranked method in both Fig. 1 (bottom) and Fig. 3 (right), only uses MVs from the video bitstream to predict saliency, showing that reasonable saliency prediction can be made based on MVs alone.

In the assessments presented thus far, the QP value for MPEG-4 ASP encoding was set to 12. Table 2 shows the average AUC’ scores for QP ∈ {4, 12, 24}. The quality of encoded video drops as QP increases (see Table 3), so the results in Table 2 indicate the sensitivity of the models’ saliency prediction relative to the quality of encoded video. As seen in the table, the models typically score slightly poorer as the video quality drops, because of the less accurate MVs and DCT coefficients at lower qualities. Nonetheless, the saliency prediction performance is still reasonably consistent over this
range of QP values, leading to the conclusion that the models’ performance is not too sensitive to encoding parameters over a reasonable range of video qualities.

The average processing time per frame (CIF resolution videos at 30 fps) on a Dell Optiplex system with 2.99 GHz Intel Core 2 Quad CPU and 8 GB RAM is listed in Table 4. The time taken for extracting MVs and DCT values from the bitstream is excluded. Recall that these results correspond to MATLAB implementations of the models and the processing time can be significantly decreased through a more efficient implementation. Despite this, some of the models are fast enough for real time performance (under 33 ms per frame) even when implemented in MATLAB.

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 [32, 33]. Among the models tested in the present study, only APPROX took global motion into account by removing it prior to the analysis of results tested in the present study, only APPROX took global motion into account. Recall that these results correspond to MATLAB implementations of the models and the processing time can be significantly decreased through a more efficient implementation. Despite this, some of the models are fast enough for real time performance (under 33 ms per frame) even when implemented in MATLAB.

The effect of different QP values on the average AUC is shown in Table 2. The models in this study, only PMES, MSM-SM and PIM-ZEN perform MV preprocessing prior to saliency computation. In fact, MV preprocessing is one of the main differences between PMES and MAM, as well as between PIM-ZEN and PIM-MCS. The results show that PMES and PIM-ZEN always scored higher than MAM and PIM-MCS, respectively. All these three models (PMES, MSM-SM and PIM-ZEN) show strong performance in Fig. 1 (bottom) and Fig. 3 (right), indicating that proper MV preprocessing is useful in achieving good saliency prediction.

It is known that MVs in the compressed bitstream do not always represent true motion, since they are usually selected based on rate-distortion criteria. The resulting MV noise may have a negative impact on a model’s performance. Among the models in this study, only PMES, MSM-SM and PIM-ZEN perform MV preprocessing prior to saliency computation. In fact, MV preprocessing is one of the main differences between PMES and MAM, as well as between PIM-ZEN and PIM-MCS. The results show that PMES and PIM-ZEN always scored higher than MAM and PIM-MCS, respectively. All these three models (PMES, MSM-SM and PIM-ZEN) show strong performance in Fig. 1 (bottom) and Fig. 3 (right), indicating that proper MV preprocessing is useful in achieving good saliency prediction.

Many sequences that turned out to be challenging for saliency prediction contain global (camera) motion. The field of global motion estimation has also progressed in recent years and there are a number of methods for estimating global motion in the compressed domain (e.g. [34]), so it is reasonable to expect that compressed-domain saliency models should be able to benefit from taking global motion into account.

In addition to MVs and transform coefficients, other potentially useful information exists in the compressed bitstream that hasn’t been used by any of the models, such as block sizes and coding modes. This additional information might lead to improved accuracy of saliency prediction.

### Table 2. The effect of different QP values on the average AUC score

<table>
<thead>
<tr>
<th>Model</th>
<th>PMES</th>
<th>MAM</th>
<th>PNSP-CS</th>
<th>GAUS-CS</th>
<th>PIM-ZEN</th>
<th>PIM-MCS</th>
<th>APPROX</th>
<th>PMES</th>
<th>MAM</th>
<th>PNSP-CS</th>
<th>GAUS-CS</th>
<th>PIM-ZEN</th>
<th>PIM-MCS</th>
<th>APPROX</th>
</tr>
</thead>
<tbody>
<tr>
<td>QP = 4</td>
<td>0.60</td>
<td>0.57</td>
<td>0.58</td>
<td>0.58</td>
<td>0.59</td>
<td>0.60</td>
<td>0.57</td>
<td>0.58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QP = 12</td>
<td>0.59</td>
<td>0.57</td>
<td>0.56</td>
<td>0.56</td>
<td>0.56</td>
<td>0.59</td>
<td>0.58</td>
<td>0.58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QP = 24</td>
<td>0.59</td>
<td>0.56</td>
<td>0.54</td>
<td>0.54</td>
<td>0.55</td>
<td>0.58</td>
<td>0.56</td>
<td>0.56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 3. The average PSNR (in dB) of test sequences for various QP values

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Bus</th>
<th>City</th>
<th>Crew</th>
<th>Foreman</th>
<th>Garden</th>
<th>Hall</th>
<th>Harbour</th>
<th>Mobile</th>
<th>Mother</th>
<th>Soccer</th>
<th>Stefan</th>
<th>Temple</th>
</tr>
</thead>
<tbody>
<tr>
<td>QP = 4</td>
<td>37.2</td>
<td>37.6</td>
<td>38.7</td>
<td>38.0</td>
<td>37.2</td>
<td>39.3</td>
<td>39.6</td>
<td>36.4</td>
<td>41.1</td>
<td>38.2</td>
<td>37.8</td>
<td>36.9</td>
</tr>
<tr>
<td>QP = 12</td>
<td>30.2</td>
<td>31.4</td>
<td>33.0</td>
<td>32.5</td>
<td>29.2</td>
<td>34.4</td>
<td>29.6</td>
<td>28.7</td>
<td>36.2</td>
<td>32.2</td>
<td>30.7</td>
<td>30.0</td>
</tr>
<tr>
<td>QP = 24</td>
<td>26.7</td>
<td>28.7</td>
<td>30.7</td>
<td>29.8</td>
<td>25.0</td>
<td>31.0</td>
<td>26.3</td>
<td>24.7</td>
<td>33.8</td>
<td>30.2</td>
<td>27.0</td>
<td>26.7</td>
</tr>
</tbody>
</table>

### Table 4. Average processing time in milliseconds per frame

<table>
<thead>
<tr>
<th>Model</th>
<th>MAM</th>
<th>PMES</th>
<th>PNSP-CS</th>
<th>GAUS-CS</th>
<th>PIM-ZEN</th>
<th>PIM-MCS</th>
<th>APPROX</th>
<th>PMES</th>
<th>MAM</th>
<th>PNSP-CS</th>
<th>GAUS-CS</th>
<th>PIM-ZEN</th>
<th>PIM-MCS</th>
<th>APPROX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (ms)</td>
<td>359</td>
<td>210</td>
<td>103</td>
<td>102</td>
<td>43</td>
<td>22</td>
<td>14</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 5. The average PSNR (in dB) of test sequences for various QP values

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Bus</th>
<th>City</th>
<th>Crew</th>
<th>Foreman</th>
<th>Garden</th>
<th>Hall</th>
<th>Harbour</th>
<th>Mobile</th>
<th>Mother</th>
<th>Soccer</th>
<th>Stefan</th>
<th>Temple</th>
</tr>
</thead>
<tbody>
<tr>
<td>QP = 4</td>
<td>37.2</td>
<td>37.6</td>
<td>38.7</td>
<td>38.0</td>
<td>37.2</td>
<td>39.3</td>
<td>39.6</td>
<td>36.4</td>
<td>41.1</td>
<td>38.2</td>
<td>37.8</td>
<td>36.9</td>
</tr>
<tr>
<td>QP = 12</td>
<td>30.2</td>
<td>31.4</td>
<td>33.0</td>
<td>32.5</td>
<td>29.2</td>
<td>34.4</td>
<td>29.6</td>
<td>28.7</td>
<td>36.2</td>
<td>32.2</td>
<td>30.7</td>
<td>30.0</td>
</tr>
<tr>
<td>QP = 24</td>
<td>26.7</td>
<td>28.7</td>
<td>30.7</td>
<td>29.8</td>
<td>25.0</td>
<td>31.0</td>
<td>26.3</td>
<td>24.7</td>
<td>33.8</td>
<td>30.2</td>
<td>27.0</td>
<td>26.7</td>
</tr>
</tbody>
</table>

5. CONCLUSIONS AND FUTURE WORK

In this study we tested nine compressed-domain visual saliency models for video against gaze locations from an eye-tracking dataset. All methods were reimplemented in MATLAB and tested on a common ground truth dataset, using the same set of MPEG-4 ASP bitstreams. Performance evaluation was presented in terms of the popular AUC metric, although other metrics such as Normalized Scanpath Saliency (NSS) would also be worth investigating in the future, to gain additional insight. Care was taken to correct for center bias and border effects in the employed metric, which were issues found in earlier studies on visual saliency model evaluation. The results indicate that in many cases, reasonably accurate visual saliency estimation is possible using only motion vectors from the compressed video bitstream. This is encouraging considering that motion vectors occupy a relatively small portion of the bitstream (usually up to 20%) and no further decoding is required. On top of that, the fastest 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.
6. REFERENCES


