Predicting Video Saliency Using Crowdsourced Mouse-Tracking Data

V.A. Lyudvichenko[1], D.S. Vatolin[1]
vlyudvichenko@graphics.cs.msu.ru|dmitriy@graphics.cs.msu.ru
[1]Lomonosov Moscow State University, Moscow, Russia

This paper presents a new way of getting high-quality saliency maps for video, using a cheaper alternative to eye-tracking data. We designed a mouse-contingent video viewing system which simulates the viewers' peripheral vision based on the position of the mouse cursor. The system enables the use of mouse-tracking data recorded from an ordinary computer mouse as an alternative to real gaze fixations recorded by a more expensive eye-tracker. We developed a crowdsourcing system that enables the collection of such mouse-tracking data at large scale. Using the collected mouse-tracking data we showed that it can serve as an approximation of eye-tracking data. Moreover, trying to increase the efficiency of collected mouse-tracking data we proposed a novel deep neural network algorithm that improves the quality of mouse-tracking saliency maps.

Keywords: *saliency, deep learning, visual attention, crowdsourcing, eye tracking, mouse tracking.*

1. Introduction

When watching videos, humans distribute their attention unevenly. Some objects in the video may attract more attention than the others. This distribution can be represented by per-frame saliency maps defining the importance of each frame region for viewers. The use of saliency can improve the quality of many video processing applications such as compression [4] and retargeting etc [2].

Therefore, many research efforts have been made to develop algorithms predicting saliency of images and videos [2]. However, the quality of even the most advanced deep learning algorithms is insufficient for some video applications [1][11]. For example, deep video saliency algorithms slightly outperform eye-tracking data of a single observer [11], whereas at least 16 observers are required to get ground-truth saliency [12].

Another option to obtain high-quality saliency maps is to generate them from eye fixations of real humans using eye tracking. Arbitrarily high quality can be achieved by adding more eye-tracking data from more observers. However, collection of the data is costly and laborious because eye-trackers are expensive devices that are usually available only in special laboratories. Therefore, the scale and speed of the data collection process is limited.

Eye-tracking data is not the only way to estimate humans' visual attention. Recent works [5][9] offered alternative methodologies to eye tracking that use mouse clicks or mouse movement data to approximate eye fixations on static images. To collect such data a participant is shown an image on a screen. Initially, the image is blurred, but a participant can click on any area of the image to see the original, sharp image in a small circular region around the mouse cursor. This motivates observers to click on areas of images that are interesting to them. Therefore, the coordinates of mouse clicks can approximate real eye fixations.

Of course, such cursor-tracking data of a single observer approximates visual-attention less effectively than eye-tracking data. But in general, quality comparable with eye tracking can be achieved by adding more data recorded from more observers. The main advantage of such cursor-based approaches is that they significantly simplify the process of getting high-quality saliency maps. To collect the data only a consumer computer with a mouse is needed. Thanks to crowdsourcing web-platforms like Amazon Mechanical Turk, the data can be collected remotely and at large scale. It drastically speeds up the collection process and allows to increase the diversity of participants.

In this work, we propose a cursor-based method for approximating saliency in videos and a crowdsourcing system for collecting such data. To the best of our knowledge, it is the first attempt to construct saliency maps for video using mouse-tracking data. We show participants a video which is being played in real time in the web-browser in a special video-player simulating the peripheral vision of the human visual system. The player unevenly blurs the video in accordance with current mouse cursor position, the closer a pixel is to the cursor the less blur that is applied (Fig. 1). While watching the video a participant could freely move the cursor to see interesting objects without blurring. Using the system we collected participants' mouse-tracking data who were hired on a crowdsourcing platform. We performed an analysis of the collected data and showed that it can approximate eye-tracking saliency. In particular, saliency maps generated from mouse-tracking data of two observers have the same quality as ones generated from eye-tracking data from a single observer.

However, cursor-based approaches, as well as eye-tracking, become less efficient in terms of added quality per observer when the number of observers goes up. The contribution of each following observer to the overall quality is rapidly decreasing because the dependence between the number of observers and the quality is logarithmic in nature [7]. Thereby, each following observer is more and more expensive in terms of cost per added quality.

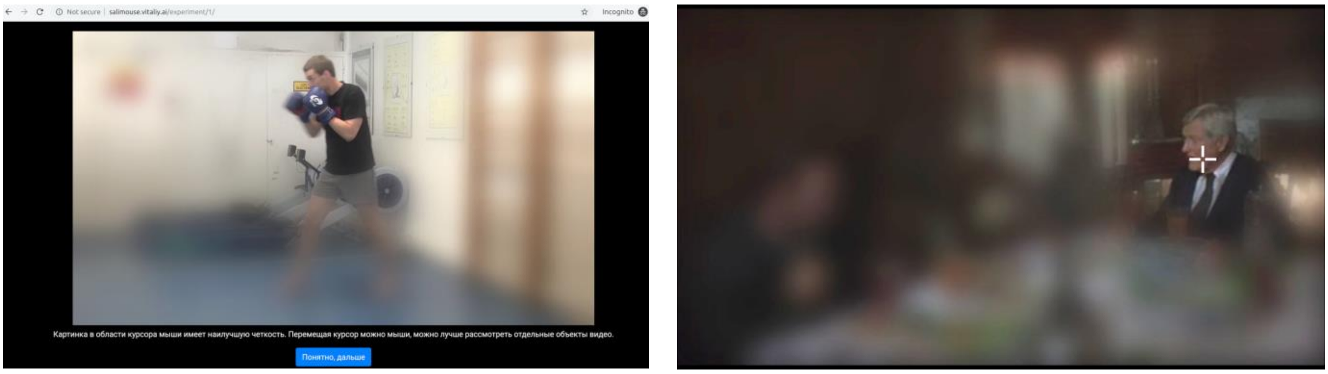
To tackle this problem the semiautomatic paradigm for predicting saliency was proposed in [4]. Unlike conventional saliency models, semiautomatic approaches take eye-tracking saliency maps as an additional input and postprocess them which enables better saliency maps using less data.

We generalized the semiautomatic paradigm to mouse-tracking data and proposed a new deep neural network algorithm working within this paradigm. The algorithm is based on SAM-ResNet [3] architecture, in which two modifications were made. Since SAM-ResNet was designed to predict saliency in images, we firstly added an LSTM layer and adapted the SAM's attention module to exploit temporal cues of videos. Then, we added a new external prior to the network which integrates mouse-tracking saliency maps into the network. We showed that both modifications applied separately and jointly improve the quality. In particular, we demonstrated that the algorithm can take mouse-tracking saliency maps that had the quality comparable with eye-tracking from three observers and improve them to the quality of eight observers.

2. Related work

The paper makes a contribution to two topics: cursor-based alternatives to eye tracking and semiautomatic saliency modeling. Hereafter we provide a brief overview of these topics.

Cursor-based alternatives to eye tracking. There were many efforts to use mouse tracking as a cheap alternative to eye tracking. However, most of these efforts were focused on webpage analysis [15]. Therefore we provide an overview of the most notable universal approaches working with natural images.



Картинка в области курсора мыши имеет наилучшую четкость. Перемещая курсор можно мыши, можно лучше рассмотреть отдельные объекты видео.

Fig. 1. An example of a tutorial page and the mouse-contingent video player used in our system. The video around the cursor is sharp.

Huang et al. [5] designed a mouse-contingent paradigm that allowed the use a mouse instead of the eye tracker to record humans' behaviors of viewing static images. They show participants the image for five seconds. The shown image is adaptively blurred to simulate peripheral vision as though a participant's gaze is focused on the mouse cursor. Participants can freely move the mouse cursor. Cursor coordinates are recorded, clustered and filtered to remove outliers. Authors showed that such cursor-based fixations have high similarity with eye-tracking fixations. Using AMT crowdsourcing platform they estimated saliency of 10000 images which were published as the SALICON dataset.

BubbleView [9] has a similar methodology, but it does not use the adaptive blurring and reveals the unblurred area of the image only when a participant clicks on it.

Sidorov et al. [13] addressed the problem of temporal saliency of video, i.e. how a whole frame is important for viewers. To estimate the temporal importance they show participants a blurred video and allow them to turn off blurring under the cursor when the mouse button is held down. Participants have a limited amount of time when they can see unblurred frames, therefore they push the button down only on interesting frames.

To the best of our knowledge, our method is the first attempt to estimate spatial saliency of video using mouse-tracking data.

Semiautomatic saliency modeling. Lyudvichenko et al. [10] proposed a semiautomatic visual-attention algorithm for video. The algorithm takes eye-tracking saliency maps as an additional input and performs postprocessing transformations to them yielding saliency maps with better quality. The postprocessing is done in three steps: firstly they propagate fixations from neighboring frames to the current frame according to motion vectors, then they apply brightness correction and add a center prior image to the saliency maps maximizing the similarity between the result and ground-truth.

3. Cursor-based saliency for video

We propose a methodology for high-quality visual-attention estimation based on mouse-tracking data and a system collecting such data using crowdsourcing platforms. We show a participant the video in a special video player in real-time in full-screen mode. The player simulates the peripheral vision of the human visual system by blurring the video as though the participant's gaze is focused on the mouse cursor. The human eye retina consists of receptor cells, which are unevenly distributed throughout the eye, with a peak at the center of the field of view. The central, foveal area is most clearly visible, whereas other, peripheral ones are blurrier. We simulate that specificity by adaptively blurring video in accordance with the position of the mouse cursor. A participant can freely move the cursor simulating shifting of the gaze.

To enable real-time rendering of the adaptively blurred frames we use a simple Gaussian pyramid with two layers $\mathbf{L}^0$ and $\mathbf{L}^1$, where $\mathbf{L}^0$ is the original frame, $\mathbf{L}^1$ is a blurred frame with σ_1. The displayed image is constructed as follows: $\mathbf{I}_p = \mathbf{W}_p \mathbf{L}_p^0 + (1 - \mathbf{W}_p)\mathbf{L}_p^1$, where p is pixel coordinates and $\mathbf{W}_p$ is a blending coefficient dependent on the retina density at p. Thus, $W_p = \exp(-\|p - g\|^2 / 2\sigma_w^2)$, where g is the position of the mouse cursor, σ_w is a parameter. Both parameters σ_1 and σ_w represent the size of the foveal area and depend on screen size and the distance between the participant and the screen. Since we record the data in uncontrolled conditions and cannot compute these parameters exactly we chose $\sigma_1 = 0.02w$ and $\sigma_w = 0.2w$, where w is video width.

The system consists of front-end and back-end parts. The back-end part allocates videos among participants, stores the recorded data and communicates with a crowdsourcing platform. Before watching videos the system shows three educational pages explaining how the video player works, Fig. 1 shows the first page. The front-end part implements the video player using the HTML5 Canvas API. Also, it checks that the participant's screen size is at least 1024 pixels width and its browser is able to render video at least 20 FPS. We excluded data from participants who didn't pass these checks.

4. Semiautomatic deep neural network

To improve saliency maps generated using the cursor positions as eye fixations we developed a new neural network algorithm. The algorithm is based on SAM [3] architecture which was originally designed to predict saliency of static images. Though SAM is a static model, its retrained ResNet version can outperform the latest temporal-aware models like ACL [14] and OM-CNN [6][11]. Also, SAM architecture can be more easily adapted to video because its attentive module already uses LSTM layer to iteratively update the attention.

We make two modifications to the original SAM-ResNet architecture: adapt it for more effective video processing and add the external prior to integrate mouse-tracking saliency maps. The modified architecture is shown in Fig. 2.

Saliency models can significantly benefit from using temporal video cues. Therefore we extract 256 temporal features in addition to 256 spatial features yielded from 2048 final features of ResNet subnetwork by 1×1 convolution. The temporal features are produced by additional convolutional LSTM layer with 3×3 kernels which is fed with the final features of ResNet. Spatial and temporal features are concatenated all together and passed to the Attentive ConvLSTM module. Also, we make the Attentive ConvLSTM module truly temporal-aware by passing its states from the last iteration of the previous frame to the first iteration of the following frame. It allowed reducing the number of per-frame iterations from 4 to 3 without quality loss.

Then we integrate the external map priors in three places of the network. Firstly we add this prior to the existing Gaussian priors at the network head.

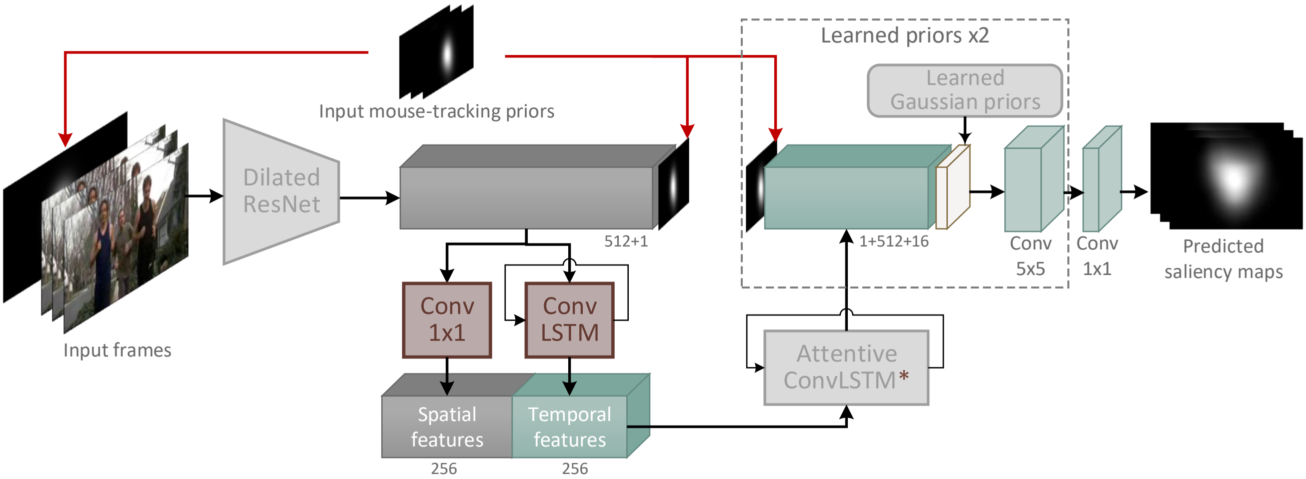


Fig. 2. Overview of proposed temporal semiautomatic model based on SAM-ResNet [3]. We introduce the external prior maps and concatenate them with the features of the input layer and three intermediate layers. To make the network temporal-aware we introduce new spatiotemporal features and adapt the attentive ConvLSTM module so that it can pass the states to the following frames. The made modifications are marked by the red color on the schema.

To learn more complex dependencies between the prior and spatiotemporal features we concatenate downsampled prior and the output of the ResNet subnetwork. Also, we concatenate it with three RGB channels of source frames. Since we use a pretrained ResNet network that expects the input with three channels, we update the weight of the first convolutional layer by adding a forth input feature initialized by zero weights.

5. Experiments

We used our cursor-based saliency system to collect mouse-movement data in 12 random videos from Hollywood-2 video saliency dataset [12] that are each 20–30 seconds long. We hired participants on Subjectify.us crowdsourcing platform, showed them 10 videos and paid them $0.15 if they watched all videos. In total, we collected data of 30 participants resulting in 22–30 views per video.

Using the collected data we estimated how good mouse- and eye-tracking fixations from the different number of observers approximate ground-truth saliency maps (generated from eye-tracking fixations). Fig. 3 shows the results and illustrates that mouse-tracking of two observers have the same quality as eye-tracking of the single observer, so the data collected with the proposed system can approximate eye-tracking.

Note, when we estimated the eye-tracking performance of N observers we compared them with the remaining $M - N$ observers of total M observers. Therefore the eye-tracking curve has stopped increasing since $N = 8$ because Hollywood-2 dataset has data of 16 observers only. All our experiments convert fixation points to saliency maps using the formula $\mathbf{SM}_p = \sum_{i=1..N} \mathcal{N}(p, f_i, \sigma)$, where $\mathbf{SM}_p$ is the resulting saliency map value at pixel p, f_i is the position of the i-th fixation point of N and $\mathcal{N}$ is a Gaussian with $\sigma = 0.0625w$, w is video width.

We also tested how the previous semiautomatic algorithm [11] works with mouse-tracking data from a different number of observers. Fig. 3 illustrates that the algorithm visibly improves mouse-tracking saliency maps making them comparable with eye-tracking. In particular, it improves mouse-tracking saliency maps of a single observer making them better than eye-tracking of a single observer.

Then we tested four configurations of proposed neural network architecture: two versions of the static variant and two versions of the temporal variant. The static variant processes frames independently, whereas the temporal one uses temporal cues. Each variant has the semiautomatic version using the external prior maps and the automatic version not using any external priors. All architectures were trained on DHF1K [14] and SAVAM [4] datasets, the training set consisted of 297 videos with 86440 frames, the validation set contained 65 videos. The NSS term was excluded from the original SAM's loss function since optimizing the NSS metric worsens all other saliency metrics. All other optimization parameters are the same as those used in the original SAM-ResNet.

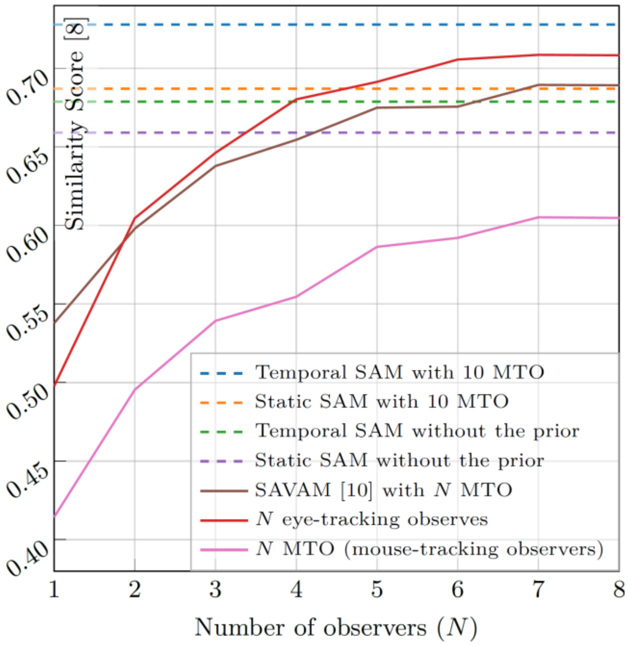


Fig. 3. Objective evaluation of four configurations of our neural network: two semiautomatic versions using the prior maps generated from mouse-tracking data of 10 observers and two automatic versions without the prior maps. The networks are compared with the mean result of $\boldsymbol{N}$ mouse- and eye-tracking observers as well as the SAVAM algorithm [10] using $\boldsymbol{N}$ mouse-tracking observers (MTO). Note, the number of observers is limited to half of the eye-tracking observers presented in the Hollywood-2 dataset [12].

The static architecture variants were trained on every 25-th frame of the videos. When training the temporal versions we composed minibatches from 3 consecutive frames of 5 different videos to use as large of a batch size as possible. Also, we disabled training of batch normalization layers to avoid problems related to small batch size.

Since the collected mouse-tracking data wasn't enough for training the semiautomatic architectures we employed transfer learning technique and used eye-tracking saliency maps for the network's external prior. The prior maps were eye-tracking saliency maps of 3 observers which have the same quality as mouse-tracking maps of 10 observers (according to Fig. 3).

Fig. 3 shows the performance of all four trained networks where the external prior maps for the semiautomatic networks were generated from mouse-tracking data of 10 observers. The figure demonstrates that the temporal configurations significantly outperform the static ones. Thus, the added temporal cues improved the Similarity Score measure [8] of the original SAM [3] static version from 0.659 to 0.678, and the semiautomatic version from 0.687 to 0.728.

The semiautomatic versions improve their prior maps and have better quality than the automatic versions. Also, they significantly outperform the semiautomatic algorithm proposed in [10]. It's worth noting that the best temporal semiautomatic configuration, which uses the prior maps generated from mouse-tracking data of 10 observers, outperforms eye-tracking of 8 observers. Since the prior maps have the same quality as 3 eye-tracking observers, the proposed semiautomatic algorithm actually improves saliency maps as though 5 more eye-tracking observers were added.

6. Conclusion

In this paper, we proposed a cheap way of getting high-quality saliency maps for video through the use of additional data. We developed a novel system that shows viewers videos in a mouse-contingent video player and collects mouse-tracking data approximating real eye fixations. We showed that mouse-tracking data can be used as an alternative to more expensive eye-tracking data. Also, we proposed a new deep semiautomatic algorithm which significantly improves mouse-tracking saliency maps and outperforms traditional automatic algorithms.

7. Acknowledgments

This work was partially supported by the Russian Foundation for Basic Research under Grant 19-01-00785 a.

8. References

[1] Borji, A. Saliency prediction in the deep learning era: An empirical investigation. *CoRR abs/1810.03716* (2018).

[2] Borji, A., and Itti, L. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 1 (2013), 185–207.

[3] Cornia, M., Baraldi, L., Serra, G., and Cucchiara, R. Predicting Human Eye Fixations via an LSTM-based Saliency Attentive Model. *IEEE Transactions on Image Processing* 27, 10 (2018), 5142–5154.

[4] Gitman, Y., Erofeev, M., Vatolin, D., Andrey, B., and Alexey, F. Semiautomatic visual-attention modeling and its application to video compression. In *International Conference on Image Processing (ICIP)* (2014), pp. 1105–1109.

[5] Huang, X., Shen, C., Boix, X., and Zhao, Q. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *International Conference on Computer Vision* (2015), pp. 262–270.

[6] Jiang, L., Xu, M., and Wang, Z. Predicting video saliency with object-to-motion cnn and two-layer convolutional lstm. *CoRR abs/1709.06316* (2017).

[7] Judd, T., Durand, F., and Torralba, A. A benchmark of computational models of saliency to predict human fixations. *Tech. rep., Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology*, 2012.

[8] Judd, T., Ehinger, K., Durand, F., and Torralba, A. Learning to predict where humans look. *In International Conference on Computer Vision (ICCV)* (2009), pp. 2106–2113.

[9] Kim, N. W., Bylinskii, Z., Borkin, M. A., Gajos, K. Z., Oliva, A., Durand, F., and Pfister, H. Bubbleview: An interface for crowdsourcing image importance maps and tracking visual attention. *ACM Trans. Comput.-Hum. Interact.* 24, 5 (2017), 1–40.

[10] Lyudvichenko, V., Erofeev, M., Gitman, Y., and Vatolin, D. A semiautomatic saliency model and its application to video compression. In *13th IEEE International Conference on Intelligent Computer Communication and Processing* (2017), pp. 403–410.

[11] Lyudvichenko, V., Erofeev, M., Ploshkin, A., and Vatolin, D. Improving video compression with deep visual-attention models. In *International Conference on Intelligent Medicine and Image Processing* (2019).

[12] Mathe, S., and Sminchisescu, C. Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2015), 1408–1424.

[13] Sidorov, O., Pedersen, M., Kim, N. W., and Shekhar, S. Are all the frames equally important? *CoRR abs/1905.07984* (2019).

[14] Wang, W., Shen, J., Guo, F., Cheng, M.- M., and Borji, A. Revisiting video saliency: A large-scale benchmark and a new model. *IEEE Conference on Computer Vision and Pattern Recognition* (2018).

[15] Xu, P., Sugano, Y., and Bulling, A. Spatiotemporal modeling and prediction of visual attention in graphical user interfaces. In *CHI Conference on Human Factors in Computing Systems* (2016), pp. 3299–3310.