Data-driven approach to dynamic visual attention modelling

Dubravko Ćulibrk[a], Srdjan Sladojević[a], Nicolas Riche[b], Matei Mancas[b] and Vladimir Crnojević[a]

[a]University of Novi Sad, Trg Dositeja Obradovića 6, Novi Sad, Serbia;
[b]University of Mons, Blvd. Dolez 31, Mons, Belgium

ABSTRACT

Visual attention deployment mechanisms allow the Human Visual System to cope with an overwhelming amount of visual data by dedicating most of the processing power to objects of interest. The ability to automatically detect areas of the visual scene that will be attended to by humans is of interest for a large number of applications, from video coding, video quality assessment to scene understanding. Due to this fact, visual saliency (bottom-up attention) models have generated significant scientific interest in recent years. Most recent work in this area deals with dynamic models of attention that deal with moving stimuli (videos) instead of traditionally used still images.

Visual saliency models are usually evaluated against ground-truth eye-tracking data collected from human subjects. However, there are precious few recently published approaches that try to learn saliency from eye-tracking data and, to the best of our knowledge, no approaches that try to do so when dynamic saliency is concerned. The paper attempts to fill this gap and describes an approach to data-driven dynamic saliency model learning. A framework is proposed that enables the use of eye-tracking data to train an arbitrary machine learning algorithm, using arbitrary features derived from the scene. We evaluate the methodology using features from a state-of-the art dynamic saliency model and show how simple machine learning algorithms can be trained to distinguish between visually salient and non-salient parts of the scene.

Keywords: Saliency, dynamic, video, data-driven, machine learning, eye-tracking

1. INTRODUCTION

When faced with visual stimuli the human vision system (HVS) does not process the whole scene in parallel. Part of the visual information sensed by the eyes is discarded in a systematic manner to attend to objects of interest. The most important function of selective visual attention is to direct our gaze rapidly towards objects of interest in our visual environment[1].[2] This results in a series of eye (in)voluntary movements (saccades), which can be recorded using eye-tracking devices.

The data obtained in such experiments is fundamental in verification of the various models of visual attention, as it represents the ground truth for such studies. While this data could potentially also be used in conjunction with machine learning techniques[3] to learn the models of attention, this venue has not been explored by the researchers. There are several probable reasons for this. First, the visual stimulus is an image or a video in case of dynamic attention and the goal of a attention model is to predict whether a pixel of an image or a frame is of interest or not. This means that for a standard resolution video (720 × 540), one has to deal with environ 9 million points per second. State of the art machine learning algorithms are unable to handle such large amounts of data. Second, the eye-tracking data is usually fairly imprecise due to the acquisition process and unsuitable for determining the level of interest on a per-pixel level. Finally, the eye-tracks of subjects viewing the same scene vary to a great extent. This is usually solved through aggregation, when using it to verify an attention model. However, in the case when one is trying to learn the attention model directly from the data

Further author information: (Send correspondence to D. C.)
E-mail: dculibrk@uns.ac.rs, Telephone: +381216571135
E-mail: S.S.:sladojevic@uns.ac.rs, V.C.:crnojevic@uns.ac.rs, M.M.:matei.mancas@umons.ac.be, N.R.:nicolas.riche@umons.ac.be

using machine learning, a more appropriate approach would be to let the machine learning algorithm generalize across the different subjects.

In this paper we explore the possibility of learning the dynamic attention model from eye-tracking data. We propose to solve the discussed problems using a novel data sampling methodology, which allows us to create a representative training and testing data set from a database of 24 videos, for which the eye-tracks of 13 users have been collected. The data is then used to train a machine learning algorithm to discern between interesting and points that are not of interest. The resulting visual attention model is then verified using standard methodology.

The rest of the paper is organized as follows: Section 2 provides an overview of the related published work. Section 3 describes the data we learn from. Section 4 is dedicated to the description of the proposed methodology. The results achieved can be found in Section 5. Finally, Section 6 contains the discussion and our conclusions.

2. RELATED WORK

It is not possible for the HVS to process an image entirely in parallel. Instead, our brain has the ability to prioritize the order the potentially most important points are attended to when presented with in a new scene. The result is that much of the visual information our eyes sense is discarded. Despite, we are able to quickly gain remarkable insight into a scene.

This type of attention is referred to as attention for perception: the selection of a subset of sensory information for further processing by another part of the information processing system[4].[5]

The most important function of selective visual attention is to direct our gaze rapidly towards objects of interest in our visual environment.[1] This ability to orientate rapidly towards salient objects in a cluttered visual scene has evolutionary significance because it allows the organism to detect quickly possible prey, mates or predators in the visual world.

Current research considers attentional deployment as a two-component mechanism.[1] Subjects selectively direct attention to objects in a scene using both bottom-up, image-based saliency cues and top-down, task-dependent cues; an idea that dates back to 19th century work of William James.[6]

Bottom-up processing is driven by the stimulus presented.[5] Some stimuli are intrinsically conspicuous or salient in a given context. For example, a red dinner jacket among black tuxedos at a sombre state affair, a flickering light in an otherwise static scene or a street sign against gray pavement, automatically and involuntarily attract attention. Saliency is independent of the nature of the particular task, operates very rapidly and is primarily determined in a bottom-up manner. If a stimulus is sufficiently salient, it will pop out of a visual scene. This suggests that saliency is computed in a pre-attentive manner across the entire visual field, most probably in terms of hierarchical *centre-surround mechanisms*. Similar goes for the moving stimuli; they are perceived to be moving only if they are undergoing motion different from their wider surround.[2] The speed of saliency-based (bottom-up) form of attention is on the order of 25 to 50 ms per item.[1] The second form of attention is a more deliberate affair and depends on the task at hand, memories and even past experience.[5] Such intentional deployment of attention has a price, because the amount of time that it takes (200 ms or more), rivals that needed to move the eyes. Thus, certain features in the visual world automatically attract attention and are experienced as "visually salient". Directing attention to other locations or objects requires voluntary "effort". Both mechanisms can operate in parallel.

While bottom-up factors that influence attention are well understood,[7] the integration of top-down knowledge into these models remains an open problem. Because of this, the fact that bottom-up components of a scene influence our attention before top-down knowledge does[8] and that they can hardly be overridden by top-down goals, applications of visual attention commonly rely on bottom-up models[491011].[12]

Significant progress has been made in terms of computational models of bottom-up visual attention (saliency) when working with still images[131415].[16] Motion cues, despite their importance, have until recently been usually considered as an extension of the static saliency. Dynamic saliency models, operating on videos, are gaining interest in the scientific community[1712].[18]

The general approach to detecting the salient points is to devise biologically inspired features that are sensitive to texture, center-surround differences in intensity and color or motion and changes in case of dynamic models.

These features are usually designed to achieve scale invariance of the model. Once the features are extracted for each point in an image or a video frames, various ways are used to distinguish between salient and non-salient points.

Zhang *et al.*[16] proposed a Bayesian framework for saliency detection (*SUN model*). Their approach is based on the assumption that visual saliency is actually the probability of a target (e.g. food or predators) at every location given the visual features observed. Bottom-up saliency is considered equal to what is known in information theory as self-information of the random variable that represents the visual features of point. Saliency, therefore is dependent solely on the probability distribution of feature values. Zhang *et al.* estimate this probability by computing feature response maps on a set of 138 images of natural scenes. The distribution for each feature is parametrized by fitting a zero-mean generalized Gaussian. The authors did not propose their own features, but chose to test their model using biologically-plausible Difference-of-Gaussian filters, akin to those used in the seminal paper of Itti *et al.*[19] In addition, features proposed by Bruce and Tsotsos[14] were considered, which are linear features derived form a set of natural images using Independent Component Analysis (ICA). In their study Zhang *et al.* concluded that the performance of their approach is much better when independent ICA features are used instead of dependent DoG features. Therefore, these form the foundation of the algorithm tested in this study. It should be noted that the SUN model does not handle motion explicitly and actually performs static-saliency estimation for each frame.

Seo and Milanfar[15] proposed a framework for static and space-time saliency detection (*SEO model*). Their saliency model relies solely on center-surround differences between a local (3×3 pixel) window centered on a pixel location (l) and its neighboring windows. To represent image structure in the windows, they proposed the use of Local Steering Kernels (LSK). These features are canonical kernels fitted to pixel value differences based on estimated gradients. To take motion into account and achieve space-time (dynamic) saliency detection, Seo and Milanfar simply extend the pixel location with the time coordinate, thus making their feature sensitive to changes in pixel value between two consecutive frames, but treating time simply as an extension of space. Similar to the approach of Zhang *et al.*, they estimate the conditional probability of a point being salient, depending on the features extracted for both the central window and the surrounding region. However, rather than using a parametric estimate, Seo and Milanfar estimated this conditional probability using Kernel Density Estimation (KDE). The saliency, then, is computed as a center value of a normalized adaptive kernel computed for the center + surround region.

Mancas model[17] uses only dynamic features such as speed and direction. No static cues about colors, gray levels or orientations are included in the model. The algorithm is based on three main steps: motion feature extraction, spatio-temporal filtering, and rare motion extraction. The model is inteded to quantify rare and abnormal motion. As a first step in the Mancas model, features are extracted making use of Farneback's algorithm for optical flow.[20] The features are then discretized into 4 directions (north, south, west, east) and 5 speeds (very slow, slow, mean, fast, very fast). Next, a spatio-temporal low-pass filter is used to cope with the discretized feature channels and its purpose is to keep just the consistent spatio-temporal events. Frames are first spatially low-pass filtered; then, a weighted sum is carried out in time, using a loop and a multiplication factor. This process tends to provide lower weight to those frames entering the loop several times (the older frames). After the filtering of each of the 9 feature channels (4 directions, 5 speeds), a histogram with 5 bins is computed for each resulting image and the self-information of the pixels for a given bin is computed. Self-information is a pixel saliency index and the different feature channels and then speed and direction maps are fused using the maximum operator. The final saliency map specifies the rarity of the statistics of a given video volume at two different scales.

Culibrk *et al.*[12] proposed the use of a multi-scale background modeling and foreground segmentation approach, as an efficient saliency model driven by both motion and simple static cues, which adheres to the motion-saliency principles.[2] The model employs the principles of multi-scale processing, cross-scale motion consistency, outlier detection and temporal coherence. The algorithm employs a multi-scale model of the background in the form of a Gaussian pyramid. This allows the approach to account for the spatial coherence and cross-scale consistency of changes due to motion of both camera and objects. Even with a small number of scales (3-5), the approach is able the achieve good segmentation of interesting moving objects in the scene. Moreover, it is able to do so consistently over a wide range of the amount of coding artifacts present.

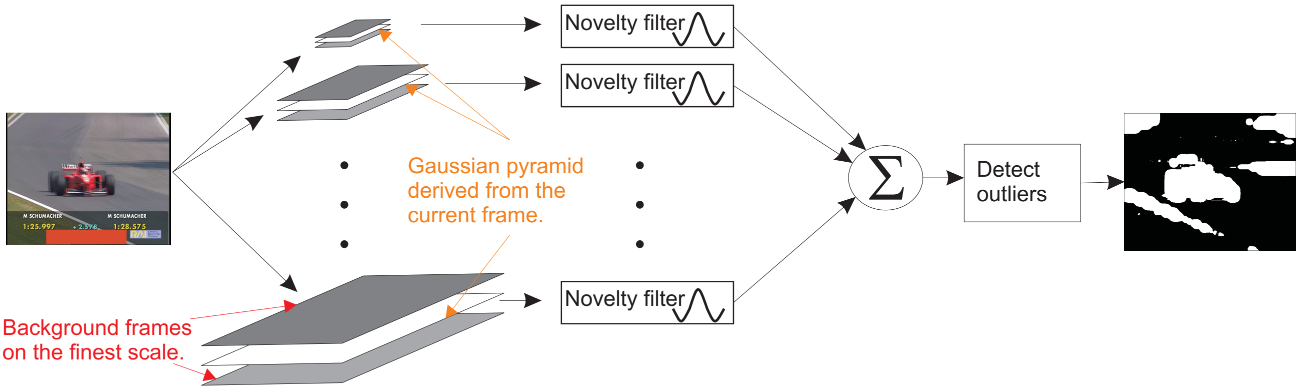


Figure 1. Salient motion detection approach of Culibrk *et al.*

The background frames at each level are obtained by infinite impulse response (running average) filtering commonly used in background subtraction[21].[22] This allows the approach to take into account temporal consistency in the frames and to reduce the saliency of static objects as time passes. Rather than attemptig to learn the distribution of the features, outlier detection[23] is used to detect salient changes in the frame. The assumption is that the salient changes are those that differ significantly from the changes undergone by most of the pixels in the frame.

A block diagram of the Culibrk *et al.* approach is shown in Fig. 1. Each frame of the sequence is iteratively passed to a 2D Gaussian filter and decimated to obtain a pyramid of frame representations at different scales. The background model is maintained in the form of two (background) frames for each scale of the Gaussian pyramid, updated in accordance with Eq. 1.

$$b_l(i) = (1 - \alpha_l) b_l(i) + \alpha_l p(i), l \in \{1, 2\} \tag{1}$$

where: α_l is the learning rate used to filter the l-th background frame, $p(i)$ is the value of pixel at location i in the current frame, $b_l(i)$ is the value of pixel at location i in the l-th background frame.

In the study presented here, we used the dynamic-saliency approach of Culibrk *et al.* to extract the features for each pixel. However, instead of assuming that the salient points are the outliers of the distribution of feature values for a single frame, we used the extracted features to train a machine learning algorithm to classify between salient and non-salient points.

To the best of our knowledge such an approach has not been attempted before, when dynamic saliency is concerned. The closest approach is that of Rajashekar *et al.*,[24] who used eye-tracking data to create a model that would predict users' eye-fixations. However, the eye-tracking data was used just to verify that the distribution of the features they proposed differs in the fixated regions from the rest of the scene and to ponder the contribution of each feature in the linear expression used to calculate saliency. No machine learning was used in that study.

When static (still image) saliency is concerned Judd and Toralba[25] attempted to learn a saliency model based on a complex set of features, including the low-level features first proposed by Itti,[19] horizon, face and person detection derived features and distance form the center of the image. They used a support vector machine classifier to learn the saliency. Recently, Carbone and Pirri proposed an approach to static saliency scan path (sequence of fixations) learning using independent component analysis to create the low-level features from a database of 'natural' images and mixtures of Bernoulli distributions to learn the saliency. They validate they approach by showing that their saliency model outperforms random selections when compared with the eye-tracking map. Both approaches make no attempt to deal with dynamic stimuli.

3. EYE TRACKING DATA

As the data to learn from, we used the ASCMN database of videos and eye-tracks. The database was created by recording the position of peoples' gaze at the videos' frame rate (30 frames per second), for 24 videos, using a commercial FaceLab eye tracking system.[26]

Table 1. The 5 classes of videos contained into the ASCMN database

Video classes	Description	Videos Nb.
1) Containing abnormal motion (**ABNORMAL**)	Some moving blobs have different speed or direction compared to the main stream: Figure 2 line 1.	2, 4, 16, 18, 20
2) Video surveillance style (**SURVEILLANCE**)	Classical surveillance camera with no special motion event: Figure 2 line 2.	1, 3, 5, 9
3) Crowd motion (**CROWD**)	Motion of more or less dense crowds: Figure 2 line 3.	8, 10, 12, 14, 21
4) Videos with moving camera (**MOVING**)	Videos taken with a moving camera: Figure 2 line 4.	6, 19, 22, 24
5) Motion noise with sudden salient motion (**NOISE**)	No motion during several seconds followed by sudden important motion: Figure 2 line 5.	7, 11, 13, 15, 17, 23

The database contains videos obtained from other databases including: the Itti's CRCNS database,[27] Vasconcelo's database[28] and a standard complex-background video surveillance database.[21] These have been extended with Internet crowd movies and proprietary videos from a crowd database acquired in Mons. The database is divided into several classes of movies as described in Table 1. Some sample frames can be seen in Figure 2.

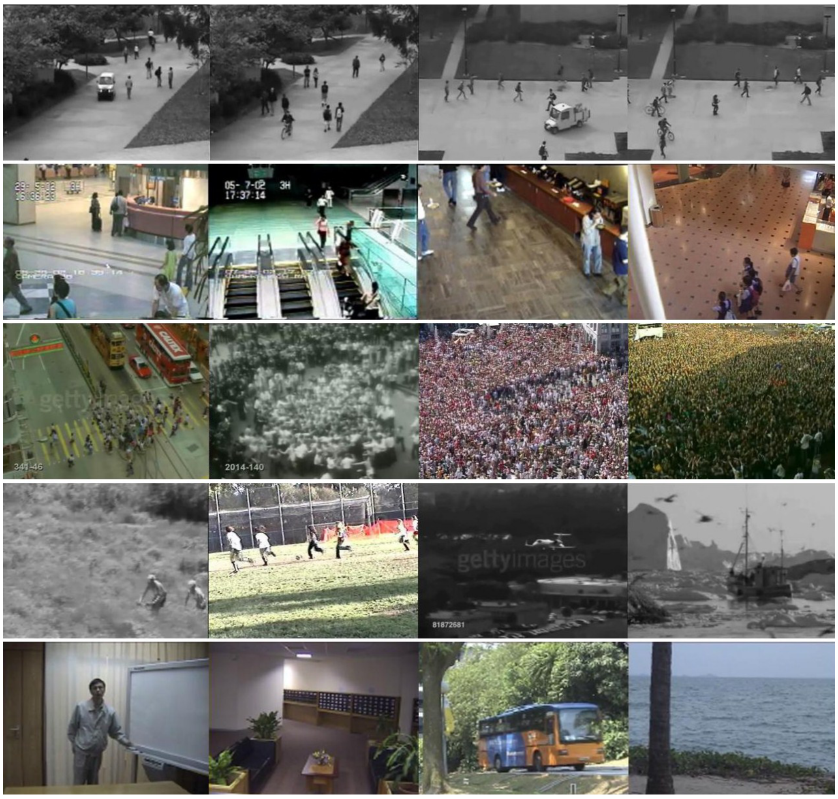


Figure 2. The five classes of videos from the ASCMN database. First line represents ABNORMAL motion in terms of speed and direction with bikes and cars which are faster than people for example. The second line shows SURVEILLANCE classical motion with nothing really salient in terms of movement. The third line shows CROWD motion with increasing density from left to right. Line four shows MOVING camera videos. Line five finally displays videos with long periods of no activity and NOISE (frames 2 and 4) and sudden salient object (frames 1 and 3).

The eye gaze positions were recorded and superimposed on the initial video for all the viewers as it can be seen in the first column of Figure 3. This representation does not correspond to the classical saliency maps produced by most models, which attempt to describe the overall saliency of the points in the frame. To obtain such a representation the maps need to be low-pass filtered to obtain a heatmap, such as that shown in the middle of the Figure 3, which represents the saliency aggregated over all viewers. The peaks of the heatmap correspond to the most salient points. To get a binary mask that would classify the points in the frame between salient and non-salient, one needs to normalize the heatmap and apply a threshold (θ). The final result is illustrated in the right column of Figure 3.

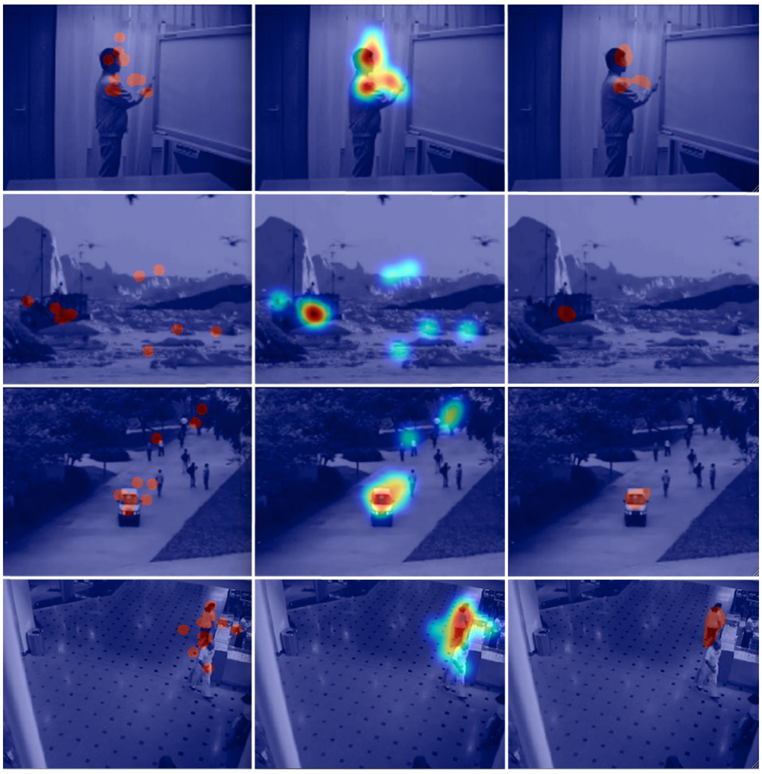


Figure 3. Left column: aggregated eye tracking results: each red dot is the position of the eye gaze of one viewer. The middle column contains smoothed gaze location producing "heatmpas" superimposed to the corresponding frame. Right column shows a thresholded version of the heatmap.

4. DATA-DRIVEN SALIENCY MODELLING

Figure 4 shows a block diagram of the proposed approach.

The key component of the approach is data sampling methodology. Our input are the eye-fixations and the original visual stimulus. We deal with videos as a more general case, but the methodology can be easily adapted to still image saliency.

As discussed in Section 3, the output of the eye-tracker is represented as a set of circular regions which contain the focus users' gaze for a certain frame. To create our dataset, that can be used for training and testing of the machine learning algorithm, features need to be extracted first. Any type of features related to visual saliency can be used. The features used in our experiments were the dynamic-saliency features proposed by Culibrk *et al.*[12]

To create the dataset used for classifier training and testing, we randomly sample pixels using both the eye-tracking binary mask, indicating the focus regions of the viewers and the set of extracted saliency features. For each sample, a set of pixel coordinates (x_i, y_j) within a frame is randomly generated. Extracted features for this

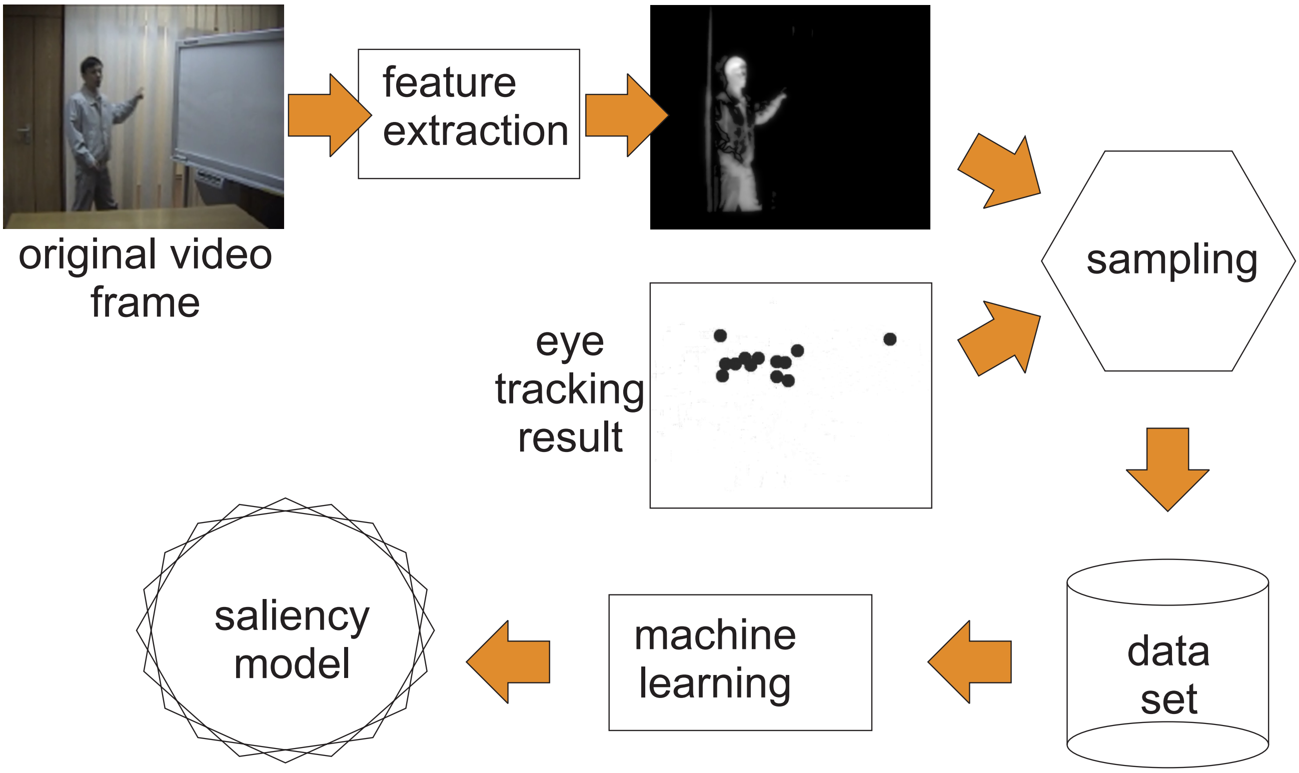


Figure 4. Data-driven saliency modelling

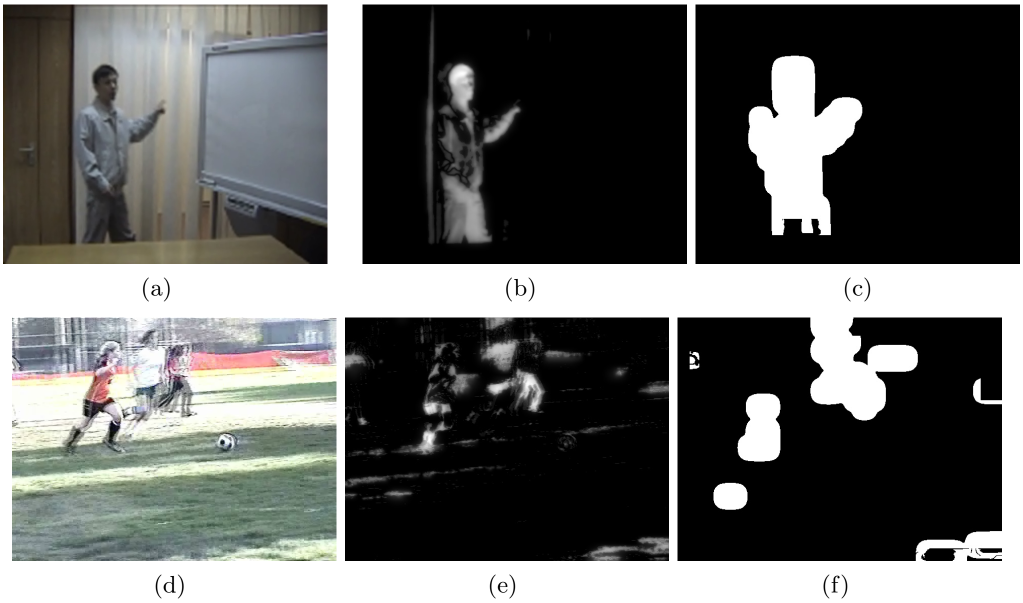
point are stored in the dataset, and a label is attached based on the examination of the eye-tracking result at those coordinates. If the pixel corresponds to a focus region it is assigned a class label of 1 and 0 otherwise. We constrain the process to create a balanced dataset, as this benefits most learning algorithms,[3] so the process is repeated until we have a predefined and equal number of samples of each class.

Unfortunately, due to the rather large diameter of regions marked in the eye-tracking maps, the values of saliency features extracted for large numbers of pixels within these regions were equal to zero. Since such data samples would not benefit the learning algorithm, these points have been discarded, and our methodology modified to create a dataset containing equal number of non-salient and salient features, with the latter constrained to points where at least one feature had a value largen than zero.

Once the dataset was created we can apply an arbitrary machine learning algorithm. We used a grafted decision tree algorithm available within the data-mining and machine learning tool Wakaito Environment for Knowledge Analysis (WEKA).[3] WEKA is an open-source tool that contains implementations of a large number of machine learning algorithms and allows one to test their performance easily. We opted for a decision tree algorithm, as the final classifier is fairly simple and easy to implement. WEKA is able to generate Java code for the trained decision tree, which we translated to C++, as the used feature extraction code is written in this programming language. The decision tree is represented as a series of conditional clauses and has a low computational cost. This is important, as the approach we are attempting to extend is designed for real-time performance, which we would not like to affect.

The decision tree learning algorithm used in our experiments is the WEKA implementation of Quinlan's C4.5 methodology,[29] with the grafting as proposed by Webb.[30] The dataset used in our experiments (in WEKA format) and the binaries of the implemented saliency detection can be obtained from `http://www.dubravkoculibrk.org/ddsaliency`. We selected 10 salient and 10 non-salient points from each frame of the videos in the ASCMN database. First 10 frames were discarded to ensure that the saliency features are stable. The final dataset contains environ 198,000 samples.

Figure 5 (c,f) shows saliency detection results for sample frames of sequences 7 and 19.


(a) (b) (c)

(d) (e) (f)

Figure 5. Sample frames for sequences 7 and 19 and the proposed approach: a,d-original frames, b,e-features (sum), c,f-detection result

5. EXPERIMENTS AND RESULTS

The basic performance statistics achieved by the proposed decision tree machine learning algorithm for the dataset created can be obtained directly from WEKA. To collect them, the default model validation procedure, 10-fold cross-validation is used.[3] The procedure involves holding out 10% of the data in the set for testing and using the rest for model building. The 10% are selected randomly and the procedure is iterated 10 times. The final performance statistics are the average values achieved for all iterations.

The overall accuracy of the proposed approach achieved is 84%. The salient points were accurately detected in 95% of the cases, but the non-salient points were detected accurately in just 73% of the cases. This is probably due to the fact that the database used cannot separate the bottom-up and top-down attention, while the features used are just stimulus driven (bottom-up). This causes the approach to erroneously classify larger part of the non-salient points as salient, as the feature values are significant but the people are focusing just on a different subset of such points. E.g. although all moving parts of a person in a video are equally salient in terms of features, the people tend to focus on the face and hands.

In order to validate the effectiveness of the proposed approach with respect to the state of the art, we compare the proposed approach (Culibrk model), with the state-of-the-art bottom-up saliency model of Seo *et al.*[15]

The comparison was made using a database of diverse videos, which was recently created for dynamic-saliency model testing, and for which the eye-tracking data of 13 viewers had been collected.

Several similarity measures have been proposed to compare saliency models with the eye tracking data. Two widely used measures were selected for the comparisons done within this study, because of their complementary nature: the Normalized Scanpath Saliency (NSS)[31] and the Correlation Coefficients (CC).

The NSS metric is the average of the response values at human eye positions in a model's saliency map. The second, CC metric, is a classical comparison method. For the CC and the NSS metric, we used the freely available Matlab functions implemented by Ali Borji and Laurent Itti.[32]

The output the proposed (Culibrk) model was first preprocessed in the same manner, as was done with the eye-tracking results, to obtain a saliency (heat) map. The saliency maps from the original eye-tracking data, the proposed and SEO model have been normalized and thresholded to obtain binary masks used for comparison. The threshold was set to 0.7. Finally, mathematical morphology operations were performed on the two saliency

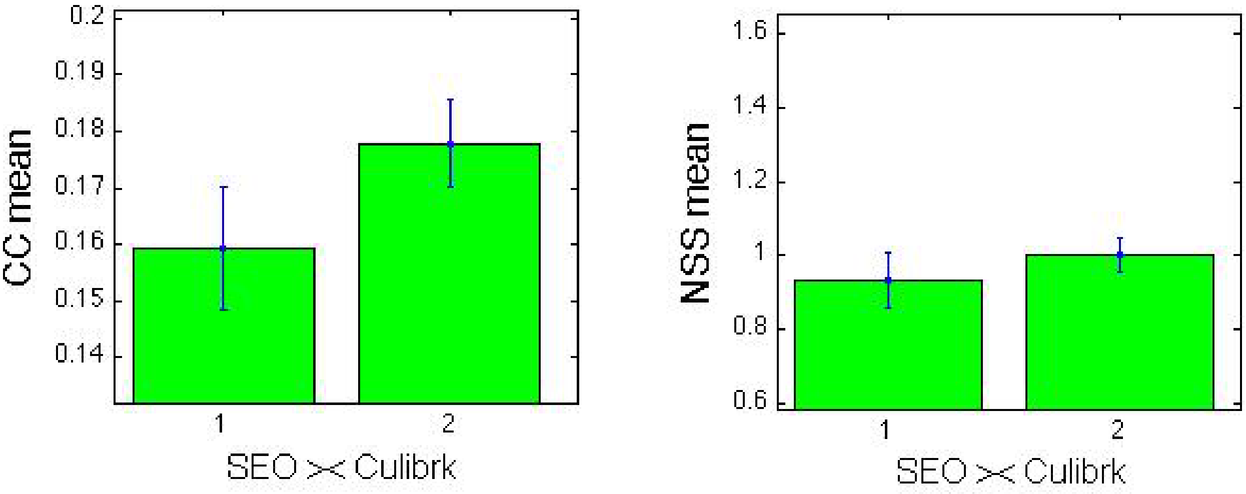


Figure 6. Average CC and NSS values obtained for the proposed (Culibrk) and SEO saliency models.

maps to make them have the same object size. The parameters of these operations were selected to maximize the performance of the approaches.

Figure 6 shows the mean and standard error for the two metrics achieved for the whole dataset. Figure 7 shows the results achieved for each sequence. The proposed methodology is able to surpass the performance of the SEO model when the whole dataset is concerned. When separate classes of videos are considered, the only class for which the SEO model achieves better result is the abnormal motion class. This can be attributed to the fact that these sequences contain a lot small moving objects motion dispersed throughout the frame. Our model is trained based on a rather coarse map of focus points that is obtained from the eye-tracking software. Thus the approach generates larger saliency regions than SEO. Even after the morphology operations the SEO model is favored for the fact that it generates smaller saliency regions.

6. CONCLUSION

The paper presents a novel approach to saliency detection. Rather than using eye-tracking data just to verify the effectiveness of the saliency model, as is usually done, a framework is proposed for learning the saliency model directly from the data.

The key technology is the data sampling methodology that allows for the extraction of a representative set of pixel saliency-related feature values and their saliency label from a database of videos for which the eye-tracking data is available. Arbitrary saliency-related features can be used for this purpose and diverse machine learning algorithms can, subsequently, be trained to achieve saliency detection.

Using features form a simple dynamic saliency model and a decision tree classifier, we compare the performance of the proposed approach against that of a state-of-the-art saliency model and show that the approach is able to surpass the model compared against.

REFERENCES

[1] Itti, L. and Koch, C., "Computational modelling of visual attention," *Nature Reviews Neuroscience* **2**, 194–203 (Mar 2001).

[2] Olveczky, B. P., Baccus, S. A., and Meister, M., "Segregation of object and background motion in the retina," *Nature* **423**, 401–408 (2003).

[3] Witten, I. H. and Frank, E., [*Data Mining: Practical machine learning tools and techniques, 2nd Edition*], Morgan Kaufmann, San Francisco (2005).

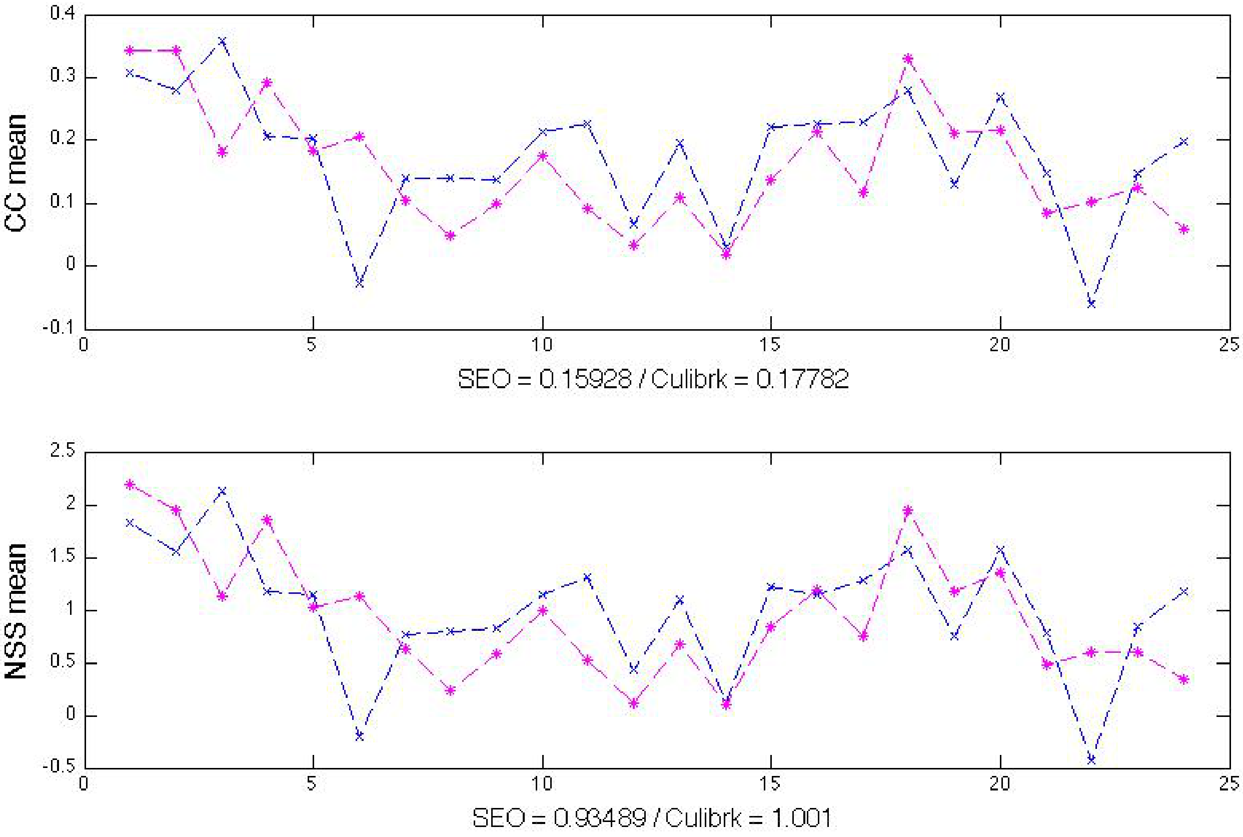


Figure 7. NSS and CC values obtained for the proposed (Culibrk) and SEO saliency models.

[4] Marques, O., Mayron, L. M., Borba, G. B., and Gamba, H. R., "An attention-driven model for grouping similar images with image retrieval applications," *EURASIP Journal on Advances in Signal Processing* **2007** (2007).

[5] Styles, E. A., [*Attention, Perception, and Memory: An Integrated Introduction*], Taylor & Francis Routledge, New York, NY (2005).

[6] James, W., [*The Principles of Psychology, Vol. 1*], Dover Publications (June 1950).

[7] Wolfe, J. M., "Visual attention," in [*Seeing*], 335–386, Academic Press (2000).

[8] Connor, C., Egeth, H., and Yantis, S., "Visual attention: bottom-up versus top-down," *Current Biology* **14**(19), R850–R852 (2004).

[9] Siagian, C. and Itti, L., "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**, 300–312 (Feb 2007).

[10] Siagian, C. and Itti, L., "Biologically inspired mobile robot vision localization," *IEEE Transactions on Robotics* **25**, 861–873 (July 2009).

[11] Ma, Y.-F., Hua, X.-S., Lu, L., and Zhang, H.-J., "A generic framework of user attention model and its application in video summarization," *Multimedia, IEEE Transactions on* **7**, 907–919 (Oct. 2005).

[12] Culibrk, D., Mirkovic, M., Zlokolica, V., Pokric, M., Crnojevic, V., and Kukolj, D., "Salient motion features for video quality assessment," *Image Processing, IEEE Transactions on* (99), 1–1 (2010).

[13] Itti, L., "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE Transactions on Image Processing* **13**, 1304–1318 (Oct 2004).

[14] Bruce, N. and Tsotsos, J., "Saliency based on information maximization," *Advances in neural information processing systems* **18**, 155 (2006).

[15] Seo, H. and Milanfar, P., "Static and space-time visual saliency detection by self-resemblance," *Journal of Vision* **9**(12), 1–12 (2009).

[16] Zhang, L., Tong, M., Marks, T., Shan, H., and Cottrell, G., "Sun: A bayesian framework for saliency using natural statistics," *Journal of Vision* **8**(7), 1–20 (2008).

[17] Mancas, M., Riche, N., and J. Leroy, B. G., "Abnormal motion selection in crowds using bottom-up saliency," in [*IEEE ICIP*], (2011).

[18] Mahadevan, V. and Vasconcelos, N., "Spatiotemporal saliency in dynamic scenes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **32**(1), 171–177 (2010).

[19] Itti, L., Koch, C., and Niebur, E., "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**, 1254–1259 (Nov 1998).

[20] Farnebäck, G., "Two-frame motion estimation based on polynomial expansion," in [*SCIA*], 363–370 (2003).

[21] Li, L., Huang, W., Gu, I., and Tian, Q., "Statistical modeling of complex backgrounds for foreground object detection," in [*IEEE Trans. Image Processing, vol. 13, pp. 1459-1472*], (2004).

[22] Rosin, L., "Thresholding for change detection," in [*Proc. of the Sixth International Conference on Computer Vision (ICCV'98)*], (1998).

[23] Hodge, V. J. and Austin, J., "A survey of outlier detection methodologies," *Artificial Intelligence Review* **22**, 85–126 (2004).

[24] Rajashekar, U., van der Linde, I., Bovik, A., and Cormack, L., "Gaffe: A gaze-attentive fixation finding engine," *Image Processing, IEEE Transactions on* **17**(4), 564–573 (2008).

[25] Judd, T., Ehinger, K., Durand, F., and Torralba, A., "Learning to predict where humans look," in [*Computer Vision, 2009 IEEE 12th International Conference on*], 2106–2113, IEEE (2009).

[26] Machine, S., "Facelab commercial eye tracking system." `http://www.seeingmachines.com/product/facelab/`.

[27] Itti, L., "Crcns-orig video and eye tracking database." `http://crcns.org/data-sets/eye/eye-1`.

[28] Mahadevan, V. and Vasconcelos, N., "Spatiotemporal saliency in dynamic scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**, 171–177 (2010).

[29] Quinlan, R. J., "Learning with continuous classes," in [*5th Australian Joint Conference on Artificial Intelligence*], 343–348 (1992).

[30] Webb, G., "Decision tree grafting from the all-tests-but-one partition," in [*International Joint Conference on Artificial Intelligence*], **16**, 702–707, Citeseer (1999).

[31] Peters, R. J., Iyer, A., Itti, L., and Koch, C., "Components of bottom-up gaze allocation in natural images," *Vision Research* **45**, 2397–2416 (Aug 2005).

[32] Borji, A., "Evaluation measures for saliency maps: Cc and nss." `https://sites.google.com/site/saliencyevaluation/evaluation-measures`.