Automatic Video Annotation Using Eye Gaze

ABSTRACT

This paper presents our efforts towards a framework for video annotation using gaze.
In computer vision, video annotation (VA) is an essential step in providing a ground-truth for the evaluation of object detection and tracking techniques.
VA is a demanding element in the development of video processing algorithms, where each object of interest should be manually labelled. Although the community has handled VA for a long time, the size of new data sets and the complexity of the new tasks pushes us to revisit it. A barrier towards automated video annotation is the recognition of the object of interest and tracking it over image sequences. To tackle this problem, we employ the concept of visual attention to enhance video annotation.
Human attention is naturally biased towards highly semantic objects providing valuable information for extracting the objects of interest which can be exploited to annotate videos. Under task-based gaze recording, we utilize an observer's gaze to filter seed object detector responses for a video sequence. The filtered boxes are then passed to an appearance-based tracking algorithm. We evaluate the gaze usefulness by comparing the algorithm with gaze and without it. We show that eye gaze is an influential cue for enhancing the automated video annotation.

1 Introduction

In the previous decades, manual annotation of videos has always been the only way to annotate videos: to track objects of interest in image sequences. Video annotation can be used in documentary film studies, enhancing video lectures by annotating them, and annotating language courses. Manual video annotation is performed by asking users to watch a video, and manually draw bounding boxes over objects of interest in each image frame [1]. The massive increase in the amount of multimedia content has made it very costly, in terms of human effort and time, to proceed in using manual annotation.

In the last decade, automated object trackers have been introduced to replace manual annotation. Different methods were used for object segmentation. One method is using Direct Acyclic Graphs (DAGs) for detection and segmentation of the primary objects in videos, then building an object model out of it [2].

Another method is using optical flow and motion boundaries to label background-foreground in image frames [3]. In spite of being faster in annotation, the trackers had noticeably lower accuracies than manual annotation.

A new approach to enhance the accuracy of automated annotation is using eye-gaze tracking data alongside the object trackers. Using eye-gaze data alongside object trackers has proven to be efficient in enhancing the accuracy of video annotation. Some implementations report an accuracy reaching up to 90% for object annotation [4]. New studies are still looking into novel approaches to efficiently use eye-gaze data. In this paper, we provide a new approach of using eye gaze data to enhance and filter bounding boxes presented as a seed for object tracking algorithms.

2 Related Work

Eye-gaze data has been used in different ways to enhance annotation results. One way is to perform a video-watching task [5]. Users are asked to freely watch a specific video. Meanwhile, eye-gaze data is constantly acquired using eye trackers. At the same time, object trackers are used to produce bounding boxes around detected objects in videos. These bounding boxes are then merged with the eye data and the annotation is refined.

Current techniques for this method are divided into two approaches. The first approach is applying image processing techniques to data from eye-gaze and images. In [5], an approach relying on image processing is implemented: after eye-gaze data is collected, trackers- path lines for eye fixations- are generated. Trackers are then merged with the bounding boxes generated from object trackers. The output is used to segment objects of interest from the image sequence and to refine the previously generated bounding boxes.

The second approach is modelling the eye-gaze data to later classify objects of interest in videos. In [6], eye-gaze data is collected and feature vectors are generated out of the data. The feature vectors are then used to train a classifier, and the eye-gaze is modelled. The built up classifier models the eye-gaze data of a user looking at an object of interest, and separates it from the rest of the image scene. Machine learning algorithms are then applied to refine the bounding boxes for these objects.

3 Approach

In this paper, we present an approach that mainly focuses on refining tracking results by filtering out false positive bounding boxes using eye gaze data. The approach assumes that users watching a video sequence will always be following an object of interest at any given time frame. Using this assumption, we filter out bounding boxes that are far away of eye gaze points; as they have high probability of being false positive detections. A detailed description of the approach is provided below.

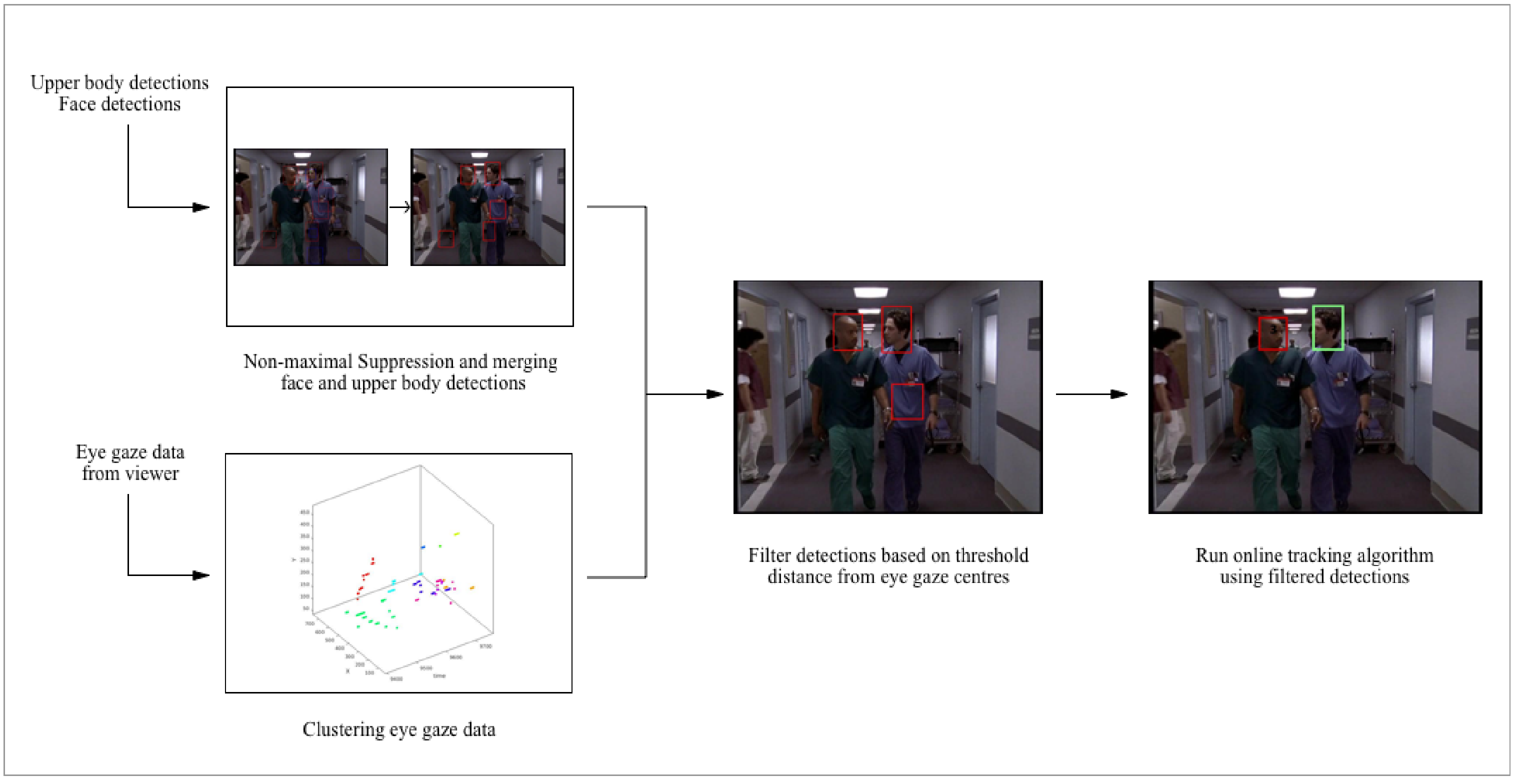

Upper body detections
Face detections
Non-maximal Suppression and merging face and upper body detections
Eye gaze data from viewer
Clustering eye gaze data
Filter detections based on threshold distance from eye gaze centres
Run online tracking algorithm using filtered detections

Figure 1: A general outline of the proposed algorithm.

3.1 Collecting and Processing Eye Gaze

Eye movements can be generally divided into: fixations, saccades, and scanpath [7]. These movements are recorded using trackers. Eye gaze data has proved efficiency in computer vision problems. An example of that is using this data in the detection of actions in image frames using a weakly-supervised classifier [8].

We use the eye gaze data differently: we filter out saccades, as probably the viewer is moving his eyes from one object of interest to the other. We then keep fixations and scanpaths. Both of them have high probability of being objects of interests, as the viewer fixes his eyes on an object, and follows it as it moves. At this point we have eye gaze data, which includes fixations and scanpaths. We then use this data to filter out false positive bounding boxes.

3.2 Filtering Bounding Boxes Using Eye Gaze Data

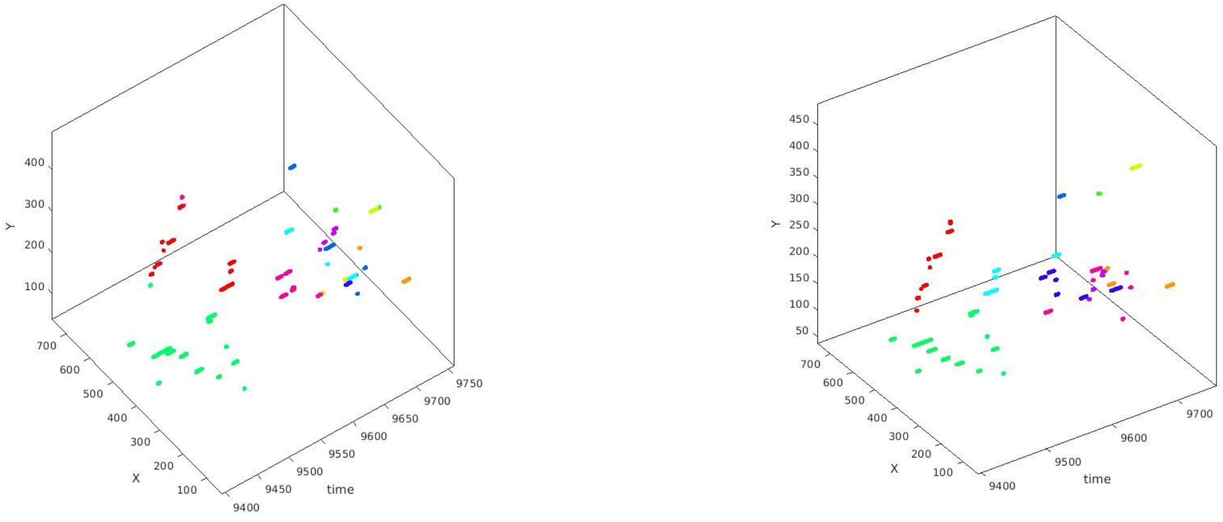
Filtering bounding boxes collected from object trackers is done in two steps. First, preliminary face and body bounding boxes for the whole sequence are collected. Face bounding boxes are collected using Viola-Jones Object Detection algorithm [9]. Upper body bounding boxes are collected using Oxford upper body detector [10]. At this point we have eye gaze data of the whole sequence, face bounding boxes and upper body bounding boxes. Non-maximum suppression is then performed on upper body bounding boxes using their scores, this helps in filtering redundant bounding boxes. After getting the filtered upper body boxes, the approach merges them with the face bounding boxes using a threshold overlap ratio.

Second, we use eye gaze data to filter out false positive bounding boxes. The filtering is done as follows:

- Initializing eye gaze data centers by clustering eye gaze data with time as a 3 dimensional feature vector using spectral clustering [11]. We use self-tuning spectral clustering as it allows us to cluster without knowing the number of clusters (candidate objects).
- After running the spectral clustering algorithm to get the number of candidate objects and initial centers, we run a clustering algorithm using resulted centers as seeds to enhance clustering. We used both k-means and Gaussian Mixture Models for clustering. We discuss results of using both approaches in section 4. Figure 2 shows a sample of the clusters.
- At this point, we have centers of 3 dimensions each, which are probably results of the eye fixating and tracking an object of interest.
- We use the resulted centers to filter out false positive bounding boxes. The paper implements two approaches for that. First approach is calculating distance of all bounding boxes across the whole sequence from each center. For each center, we get either mean or median of distances and save it as threshold distance, and then filter out bounding boxes with distances greater than that threshold. Second approach takes

into consideration the activation time frame of each center: the time period in which the cluster exists. We calculate the distance between each center and its time-activated bounding boxes. We then filter out bounding boxes for each frame using same as the previous approach (mean or median distance threshold). We discuss results from using both approaches in section 4.

- At this point, we have filtered bounding boxes across the whole sequence. We use these bounding boxes to feed the tracking algorithm.



(a) Clustering Eye Gaze with Gaussian Mixture Models

(b) Clustering Eye Gaze with k-means

Figure 2: The two diagrams show the results of clustering eye gaze data using spectral clustering first for determining number of cluster, and then clustering using (a) Gaussian Mixture Models and (b) K-means.

3.3 Using Filtered Bounding Boxes in Tracking

After filtering bounding boxes, we use them as seed boxes for an online multi-object tracking algorithm from S.-H. Bae and K.-J. Yoon paper “Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning” [12]. The paper proposes a robust online multi-object tracking method in consideration of limitations existing in other tracking methods. The method is based on tracklet confidence to handle track fragments due to occlusion or unreliable detections, and online discriminative appearance learning to handle similar appearances of different objects in tracklet association. The two parts can be explained as follows:

- The tracklet confidence is calculated based on detectability and continuity of a tracklet. Three factors are measured to calculate confidence: occlusion, length and affinity of tracklet. This helps in handling frequent occlusions by clutter or other objects. After calculating the confidence, and to formulate the problem of multi-object tracking, tracklets of high confidence are associated with the detections provided by our gaze-filtration algorithm. On the other hand, tracklets with low confidence are globally associated with other tracklets and detections. This strategy allows tracklets to grow sequentially with gaze-filtered detections and fragmented tracklets can be linked to them without expensive associations.
- Association process is performed by using appearance modeling. Appearance modeling is crucial to associate tracklets and detections of the same object and distinguishing different objects. The paper proposes a novel online discriminative appearance learning which takes into consideration two main issues in multi-object tracking: online learning using tracking results to update appearance models, and online training sample collection for discriminating appearances of multiple tracked objects. The method considers both issues to learn discriminative appearance models using an incremental linear discriminant analysis (ILDA). This allows to distinguish each object and incrementally update learned appearance models with online tracking results.

The system provides frame-by-frame detections, along with identification of each detection in each frame. An overview of the system can be seen in Figure 3.

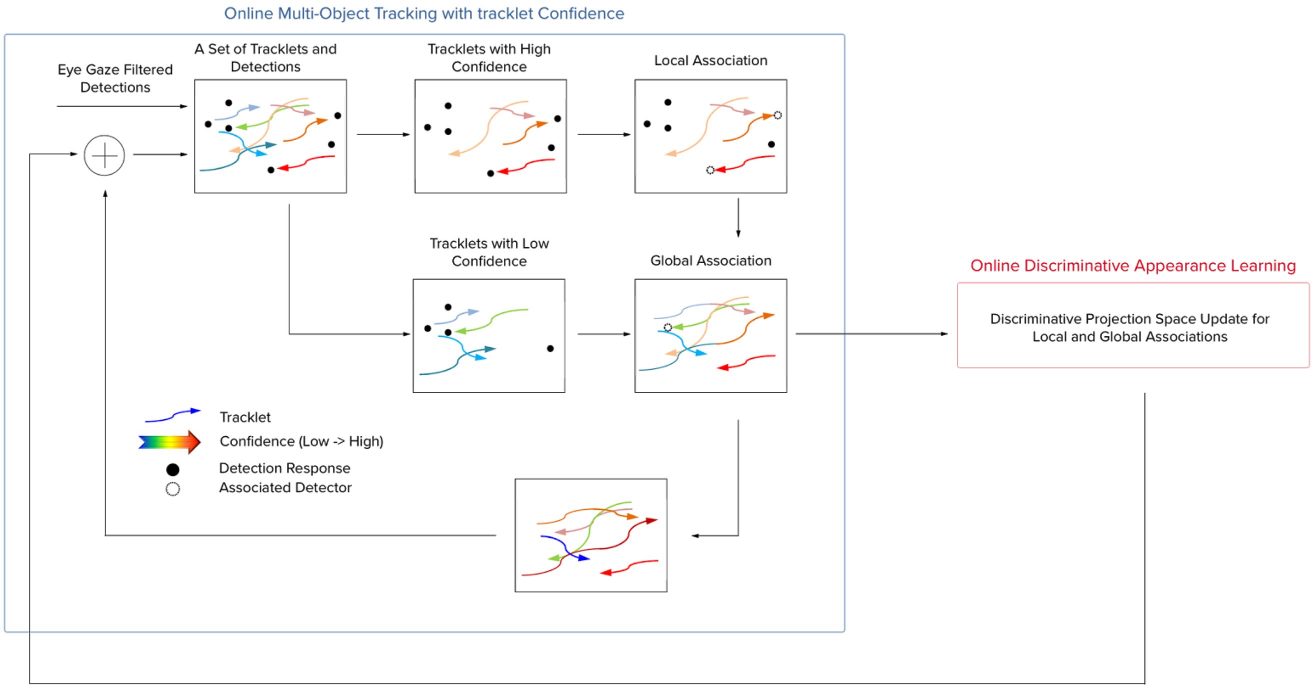

Online Multi-Object Tracking with tracklet Confidence
Eye Gaze Filtered Detections
A Set of Tracklets and Detections
Tracklets with High Confidence
Local Association
Tracklets with Low Confidence
Global Association
Online Discriminative Appearance Learning
Discriminative Projection Space Update for Local and Global Associations
Tracklet
Confidence (Low -> High)
Detection Response
Associated Detector

Figure 3: Overview of the online tracking algorithm. Adapted From "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning." by S.-H. Bae and K.-J. Yoon, in CVPR 2014.

4 Experiment and Results

4.1 Implementation

We have implemented the proposed system using MATLAB.

Eye gaze Data: A scene from "Scrubs" was used as the sequence to test, and the viewer was asked to watch it and follow objects of interest as the sequence plays. Tracking data was gathered by using iView X eye trackers. Data from one viewer was used as a proof of concept of the validity of the approach. Saccade data was ignored.

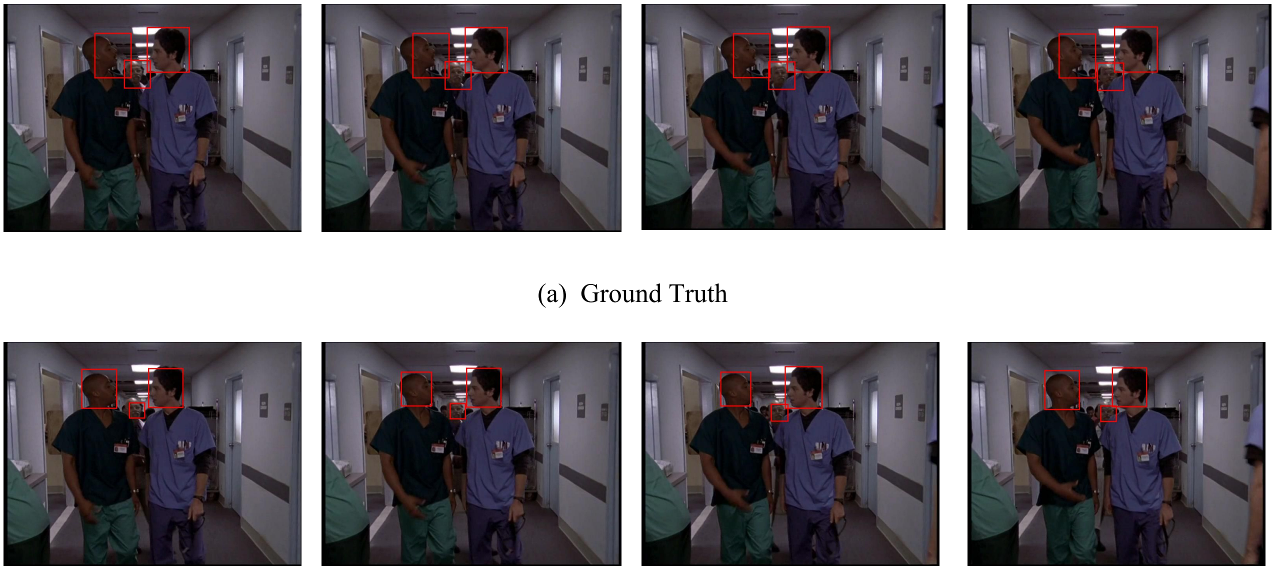
Detections: Detections for the whole image sequence were collected and then processed. Upper body detections with low confidence were ignored. Non-maximum suppression was then applied on the remaining upper body boxes. Merging with face detection data was done using a threshold overlap ratio (0.5). If the overlap was bigger than 0.5, the two boxes were merged.

Ground truth data is collected using a semi-automated annotation software. We used the “Video Annotation Tool from Irvine, California (VATIC)” software to annotate objects of interest over the sequence and use it as ground truth data.

Clustering: We tested both Gaussian mixture Models and K-means to cluster eye gaze data. We first tried clustering the data with spatial features only. This approach gave less accurate results, as it tended to cluster eye gaze points of different objects together,. After using spatio-temporal data for eye gaze, results got better. We only report results for spatio-temporal features.

Tracking: The online tracking algorithm code from [12] was then run, with filtered detections as input seed detections.

Results from sample frames can be seen in Figure 4.



(a) Ground Truth

(b) Tracking Results

Figure 4: Comparison between (a) ground truth detections and (b) results from the proposed approach on four sample consecutive frames

4.2 Performance Metrics

We used some of the performance metrics provided by [13]. We used the accuracy metric, which measures how well the bounding box predicted by our tracker overlaps with the ground truth bounding box. The tracking accuracy of a detected bounding box at time frame t is defined as the sum of overlap between predicted bounding box **PB$_t$** and ground truth bounding box **GT$_t$** at time t [Eq. 1].

$$a_t = \frac{PB_t \cap GT_t}{PB_t \cup GT_t} \qquad \text{Eq. 1}$$

If this value is greater than 0.3 for any of ground truth data, the box is marked as a true positive. We report the average overlap between true positive detected boxes and their corresponding ground truth boxes. We then use the Multiple-Object Tracking Accuracy (MOTA) from [14], but without the mismatch factor. We calculate the overall frame error rate as the ratio between the sum of false positives **fp$_t$**, misses **m$_t$** over total number of objects in a frame **g$_t$**. Error rate of the whole sequence is measured as the average of error rates of all frames [Eq. 2]. Overall accuracy is measured as 1 - Overall error rate [Eq. 3]. We also report the ratio of false positives to the total number of detections. This metric is to see whether our approach decreased the number of false positives detected by the tracker or not [Eq.4].

$$E = \frac{\sum_t m_t + fp_t}{\sum_t g_t} \qquad \text{Eq.2}$$

$$A = 1 - E \qquad \text{Eq. 3}$$

$$E_d = \frac{\sum_t fp_t}{\sum_t d_t} \qquad \text{Eq.4}$$

4.3 Results

First, we used the boxes from Viola-Jones face detection algorithm and Oxford upper body detectors without filtering them using gaze. Pre-processing steps were applied on the data (non-maximum suppression, merging upper body and face boxes based on a threshold overlap ratio). This was our baseline results to compare our algorithm with. Since the detected bounding boxes are not yet filtered, there was noise, which affected the results. After running the online tracking algorithm on the data, and with using both upper body and face detection boxes, precision was 41.1%. Recall was lower at 39.2%, while overall accuracy was 15.1%. On running the online tracker on only upper body data, precision (50.1%) was higher while recall was a little lower (37.2%). Overall accuracy was 23.3%. Results can be found in Table 1.

After that we used our filtering approach. Same pre-processing steps were applied on the face and upper body bounding boxes. Our filtering algorithm was applied to the new set of boxes, and then the filtered boxes were fed into the tracking algorithm. We report the results while using different parameters. We report the full results in Table 2.

As we can see from the results, precision was enhanced by at least ~20% at 61.3% precision, and at most by ~59% at 79.1% precision. Average overlap is more related to the original detectors' accuracy (Viola-Jones and Oxford detectors), and it is between 48% to 61.7%. A small variation between best results using time activated clustering versus non time activated clustering can be noticed, in favor of non time-activated clustering. This is because in non time-activated clustering, boxes from far away frames in times from the original cluster can be classified as belonging to it; because the (X,Y) position is close. Time activated clustering gives us more accurate results; as the output in the other case is a little misleading. Average results using GMM (Precision: 73.84%, Overall Accuracy: 30.21%) tend to be higher than using K-means (Precision: 71.78%, Overall Accuracy: 28.49%) for clustering. Missed ground truth boxes (and consequently recall) tend to decrease when boxes from the face detection algorithm are used alongside upper body boxes. This indicates that viewers focus on both face and upper body of objects of interest while viewing a video.

With Face Detections				Without Face Detections				Face only			
Precision	Recall	Accuracy	Average Overlap	Precision	Recall	Accuracy	Average Overlap	Precision	Recall	Accuracy	Average Overlap
41.1%	39.2%	15.1%	49.3%	39.1%	37.2%	23.3%	48.7%	36.23%	17.95%	6.8%	39.9%

Table 1: Baseline results of running the tracking algorithm on all bounding boxes.

Clustering Method	Threshold Method	Time activation for Clusters	With face Boxes	Boxes taken/filtered	Average Overlap	Precision	Recall	Overall Accuracy	Misses/ False Positives
K-means	Mean	Yes	Yes	1216/383	49.5%	67.6%	56.5%	29.2%	374/233
K-means	Mean	No	Yes	1213/386	49%	63.1%	56.1%	23.1%	377/282

K-means	Mean	Yes	No	951/339	58.7%	70.5%	50%	28.2%	434/182
K-means	Mean	No	No	967/323	59.9%	74.5%	50.4%	32.3%	427/149
K-means	Median	Yes	Yes	843/756	49.4%	74.5%	50.2%	32.8%	428/148
K-means	Median	No	Yes	844/755	50.4%	68.6%	46%	25.2%	462/181
K-means	Median	Yes	No	676/614	58.1%	76.3%	43%	29%	493/116
K-means	Median	No	No	676/614	60.5%	79.1%	38.6%	28.1%	522/87
GMM	Mean	Yes	Yes	1124/475	48.3%	71.7%	57.8%	33.7%	368/200
GMM	Mean	No	Yes	1168/431	49%	68.5%	56.8%	30.6%	371/225
GMM	Mean	Yes	No	867/423	58.4%	75.8%	47.5%	31.58%	455/132
GMM	Mean	No	No	945/345	60.2%	77.1%	50.3%	34.6%	431/130
GMM	Median	Yes	Yes	839/760	48%	72.3%	47.3%	29.1%	453/156
GMM	Median	No	Yes	833/766	51.2%	72.4%	47%	28.9%	437/148
GMM	Median	Yes	No	677/613	57.9%	74.1%	39.6%	25.1%	522/120
GMM	Median	No	No	675/615	61.7%	78.8%	39.5%	28.1%	499/88

Table 2: Complete results for all experiments. Clustering method indicates the method used to cluster eye gaze points. Threshold method indicates the method used to label the detector bounding boxes and assign them to an eye gaze cluster. Time activation indicates that bounding boxes that are only present in the cluster time frame are considered. Face boxes indicates using detections from the face detector alongside upper body detector. Boxes taken/filtered indicates the number of boxes that were filtered out using the clustering algorithm before feeding the boxes to the online tracker. Average overlap indicates the average overlap between ground truth boxes and valid online tracking boxes.
Results are reported after that. Precision indicates the ratio between valid online tracking boxes to the whole online tracking boxes. Recall indicates the ratio between true positives and total ground truth boxes. Overall accuracy indicates the result of the ratio between false positives and misses to total ground truth boxes subtracted from 1. Misses/false positives further investigates the exact number of missed boxes from ground truth and false positives from online detected boxes.
Green highlighting indicates best results for trials **with** time activation for clusters. Red highlighting indicates best results for trials **without** time activation for clusters.

5 Conclusion

Using eye-gaze data has significantly enhanced Human-Computer interaction tasks. Video annotation, being one of those tasks, has been affected positively. The introduction of the usage of this data has sped up the process of manual annotation of videos, and reduced the amount of interaction needed to annotate videos. In this paper, we proposed a filtering approach that improved tracking accuracy for automated object trackers. The filtering approach used eye gaze data from viewers to label salient objects and objects of interest. The proposed approach can be further refined to work as an individual tracking algorithm, by just using eye gaze data from multiple viewers.

References

[1] Carl Vondrick , Deva Ramanan , Donald Patterson, Efficiently scaling up video annotation with crowdsourced marketplaces, Proceedings of the 11th European conference on Computer vision: Part IV, September 05-11, 2010, Heraklion, Crete, Greece

[2] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 628–635. IEEE, 2013.

[3] Anestis Papazoglou, Vittorio Ferrari, Fast Object Segmentation in Unconstrained Video. ICCV 2013: 1777-1784

[4] Zsolt Palotai, Miklos Lang, Andras Sarkany, Zoltan Toser, Daniel Sonntag, Takumi Toyama, András Lörincz, LabelMovie: Semi-supervised machine annotation tool with quality assurance and crowd-sourcing options for videos. CBMI 2014: 1-4

[5] S. Karthikeyan, Thuyen Ngo, Miguel Eckstein and B.S. Manjunath, "Eye tracking assisted extraction of attentionally important objects from videos", IEEE International Conference on Computer Vision and Pattern Recognition, Boston, MA, Jun. 2015

[6] Stefanos Vrochidis, Ioannis Patras, Ioannis Kompatsiaris, Exploiting gaze movements for automatic video annotation. WIAMIS 2012: 1-4

[7] Poole, A. and Ball, L. J. 2005. Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects. In Ghaoui, C. (ed.) Encyclopedia of Human Computer Interaction. Hershey, PA: Idea Group. 211-219

[8] Nataliya Shapovalova, Michalis Raptis, Leonid Sigal, Greg Mori, Action is in the Eye of the Beholder: Eye-gaze Driven Model for Spatio-Temporal Action Localization. NIPS 2013: 2409-2417

[9] Paul Viola, Michael J. Jones, “Robust Real-Time Face Detection”, International Journal of Computer Vision, Volume 57 Issue 2, May 2004, Pages 137 - 154

[10] Ferrari, V., Marin-Jimenez, M. and Zisserman, A., Progressive Search Space Reduction for Human Pose Estimation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2008)

[11] Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In L. K. Saul, Y. Weiss, and L. Bottou, editors, Advances in Neural Information Processing Systems 17, pages 1601–1608. MIT Press, Cambridge, MA, 2005.

[12] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In CVPR 2014.

[13] Kristan, M., Pflugfelder, R., et al.: The visual object tracking vot2013 challenge results. In Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV (2013)

[14] Bernardin, K., Elbs, A., Stiefelhagen, R.: Multiple object tracking performance metrics and evaluation in a smart room environment. In: Sixth IEEE International Workshop on Visual Surveillance, in conjunction with ECCV2006, Graz, Austria (2006)