Classification of Gaze Fixations by Area of Interest

CS5525: Project Final Report

Swapnil Vekhande
Virginia Tech
Blacksburg, Virginia
svekhand@vt.edu

ACM Reference Format:
Swapnil Vekhande. 2019. Classification of Gaze Fixations by Area of Interest: CS5525: Project Final Report. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 PROBLEM STATEMENT

Eye tracking systems have proven invaluable to both the study of human interaction with visual displays. However, in spite of their obvious benefit to visual display research, they typically come with the caveat of requiring manual annotation and reduction of data (e.g., marking start and stop times of a task, assigning gaze fixations to an area of interest (AOI)). This caveat commonly requires many man-hours, and exists because these features simply are not available, as is usually the case with mainstream eye tracking applications, or because eye tracking itself is not accurate often enough to support these features. The issue of unreliable tracking typically arises from the method of eye tracking itself, or poor calibration of the eye tracking system. Infrared corneal reflection tracking [1] is a prime example of an unreliable method, and is common in systems available from mainstream vendors (e.g., SMI, Tobii). While we can not solve the issue of unreliable technology without producing a completely new method of tracking, the issue of automatic classification features simply being unavailable can be solved with proper data analysis. Over the course of the semester, we consulted with human-factors researchers in the Cognitive Engineering for Novel Technology (COGENT) Laboratory who frequently make use of eye-tracking technology in an effort to identify primary bottlenecks caused by requisite manual coding of the dataset and the optimal analytical solution based on their stated needs.

1.1 Response to Proposal Comments

As per suggestion supervised and un-supervised classification techniques like Support Vector Machine and K-means have been implemented in this project.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 DATA DESCRIPTION

The data for this project is exported from BeGaze eye tracking software as a CSV where each data point represents an eye tracking event. The entire data set is chronologically ordered relative to a video recording captured by the eye tracking glasses[2, 3]. Eye tracking events are primarily classified within the dataset by behavior (i.e., blinks, saccades, or fixations). These classifications determine what quantitative data is available for the event (e.g., blink events do not report pupil size). Events are quantified temporally both absolutely, according to epoch time; and relatively, according to video time. Temporal data is reported at millisecond accuracy, and positional data is reported as pixel location (~1280x1280 resolution) down to the thousandth of a pixel. Although each event is quantified absolutely and relatively in terms of time; positional data is only reported as relative to the video recording. This shortcoming is addressed during pre-processing of the raw data.

3 DATA PROCESSING

Pre-processing of the data consists of trimming ubiquitously irrelevant data from the set. During this step, irrelevant eye tracking events are removed from the resulting data set (i.e., blinks and saccades). Further, fixation events are reduced to only time and coordinate attributes. Finally, time-stamp attributes are converted to plain millisecond values to allow faster comparisons when assigning a fixation event to a frame of the video.

Primary processing of data occurs during video processing, wherein paired frames are checked for homography and their perspectives warped accordingly [4–6]. This action is supplemented by a dynamic set of reference frames to account for the inevitable lowering of accuracy that occurs when handling a set of images this large and updating homography in a pairwise fashion. The features are detected from the video using scale-invariant feature transform (SIFT). The features generated from each image are matched using a brute-force technique, and filtered using Lowe's ratio test. Using this data, we compare match quality between the current frame, the set of reference frames, and the previous frame. Based on this comparison we warp the perspective of the current frame and add it to the panorama and map associated fixation coordinate data. This continues until the video data has been completely processed. The algorithm for this process is outlined in Algorithm 1.

4 DATA EXPLORATION

The output of any successful fixation classification system for human factors research will allow arbitrary statistical testing of the processed eye tracking data. While there are common statistical tests that may be performed with eye tracking data, researchers employing novel technology may want to employ obscure statistical

Load eye tracker data
Load video
Initialize feature detector and matcher
Initialize panorama with first frame
Initialize set of reference frames as set containing first frame
for *all pairs of frames* **do**
  Remove distortion from frame using camera distortion values
  Calculate blurriness factor of current frame
  Get brute-force matches for frame pair
  Get brute-force matches for current frame and all reference frames
  if *No good matches* **then**
    Add current frame to set of reference frames
  end
  if *Good match with reference frame* **then**
    Calculate homography with reference frame
  else
    Calculate homography with previous frame
  end
  Warp perspective of current frame using calculated homography matrix
  Add to panorama
  Map fixation coordinates that lie within frame's timespan
  Assign {key-points, descriptors, frame data} of current frame to previous frame
end

Algorithm 1: Data Preprocessing: Feature Extraction and Projective Transformation

Figure 1: Resulting fixation points marked in green

methods to identify the most relevant metrics. As such, we provide two primary outputs: a simple visualization of fixation clusters, color coded according to their associated cluster as reported by the k-means clustering algorithm; and a raw CSV output of the mapped x and y coordinates of a fixation along with a *cluster* attribute, again as reported by k-means.

As a secondary output, some simple metrics are reported. These metrics include the ratio of total time fixating to total video time, time spent fixating per cluster, and number of fixation per cluster. These metrics throw light on the relative distribution of the data

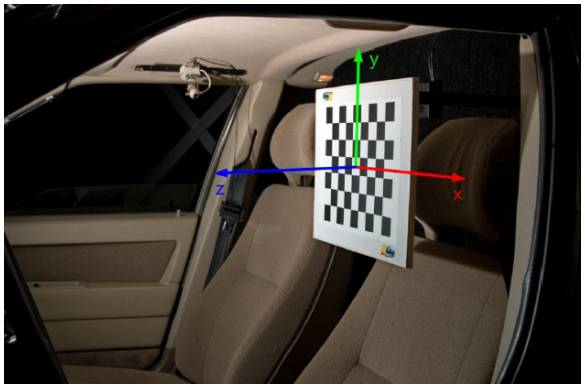

Figure 2: Transformation of video to world coordinates in data-preprocessing.

over the course of the experimental session as captured by the video file.

5 MODEL BUILDING

Two primary algorithmic systems must be implemented: one for the extraction of relevant temporal segments of the data set, and another for classification of fixations within those segments in terms of AOIs using machine learning techniques. Both systems relied on positional data transformations performed during preprocessing.

Relevant temporal segment extraction can be achieved by noting instances of unsuccessful homographical mapping and confirming it is due to the scene not being visible within the ego-centric video; or by referencing predictions generated by machine learning techniques. To this effect, outliers were removed from the detected features by using Lowe's ratio test. The descriptors and matching point sets were there subjected to homographic projections which enabled the transformation of the two dimensional eye gaze tracker data points into three dimensional space so they could be mapped. The reference frame are checked for relevance from time to time. If found obsolete they are discarded and replaced with suitable best frames. Finally temporally relevant gaze fixations were extracted by using a gaze ration. The learning classifier used for this extraction considered the behavioral categories of surrounding events weighted by temporal proximity on a per-set basis.

If either of the conditions for irrelevance are met, we can begin to conclude events associated with the video frame in question to be useless. When sufficiently large segment of video were associated with useless eye tracking events, we safely discarded this temporal segment from the data set so that it did not interfere with statistical testing.

The machine learning algorithms that were used are discussed below:-

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.

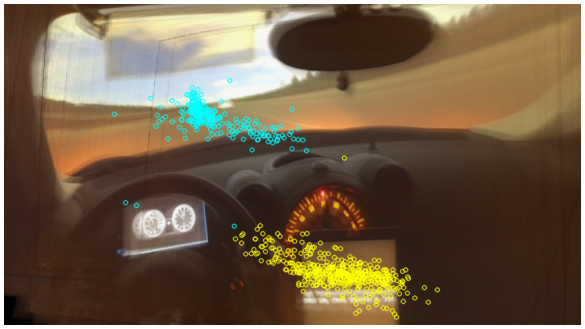

Figure 3: AOIs as per Support Vector Machine.

Figure 4: AOI classified as per K-means

Classification by most relevant AOI employed Support Vector Machine. The Linear SVM classifier was employed to classify the training dataset points. Let us assume that we have n labeled examples $(x_1, y_1), \ldots, (x_n, y_n)$ with labels $y_i \in \{1, -1\}$. We want to find the hyperplane $< w, x > +b = 0$ (i.e. with parameters (w, b)) satisfying the following three conditions:

(1) The scale of (w, b) is fixed so that the plane is in canonical position w.r.t. $\{x_1, \ldots, x_n\}$. i.e.,

$$\min_{i \leq n} | < w, x_i > +b| = 1$$

(2) The plane with parameters (w, b) separates the $+1$'s from the the -1's. i.e.,

$$y_i(< w, x_i > +b) \geq 0 \text{ for all } i \leq n$$

(3) The plane has maximum margin $\rho = 1/|w|$. i.e., minimum $|w|^2$.

The number of clusters were determined empirically as well as based on the domain expertise. Th readily available, labeled sets of raw eye tracking data was used to partition the space into two clusters.

K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. This unsupervised learning is helpful to get idea about the number of clusters when prior intuition is absent.

$$J(C, X) = \sum_{x \in X} min_{c \in C} ||x - c||^2 \quad (1)$$

Here , $x \in \mathbf{X}$ is the data points and $c \in \mathbf{C}$ are the cluster centers. The goal is to find C that minimizes $J(C, X)$.

To validate the result unsupervised clustering was also employed using K-means and the results were found to be convincing the hypothesis.

6 MODEL EVALUATION

Models was validated by comparing results to a manually annotated sets of raw eye tracking data. COGENT boasts access to many raw and annotated data sets gathered from multiple experiments so labeled data thus was reused again

Relevance of temporal segments was compared to manually coded start and stop times for the same data set. Using the error rate of these comparisons, the models were fine tuned to more closely fit the manual coding. During adjustment, conservative approximations were weighted more heavily when compared to aggressive approximation in the effort to always provide usable data post-processing.

The model was tested for various statistical measures like time spent fixating gaze came out to be 167 sec out of the total run time 259.875s. Total number of fixations were 725. Out of these 445 belonged to the first cluster where driver watched the road.Remaining 280 belonged to the other cluster. Time spent in the first AOI was 90.103 sec. The video was traversed for each temporal frame and the matches for the current with respect to previous were found using K-nearest neighbor algorithm. The driver was primarily interested in the road ahead and in the instrument panel(marked in blue and red respectively).

7 REAL-WORLD INSIGHTS

Through this project we learned how to automate the analysis of eye tracking data, which is collected frequently during human factors research. Automation of this analysis will expedite the process of human factors research by alleviating its primary bottleneck of manual annotation. Applicability of methods implemented for this project will extend beyond the scope eye tracking. That is, other forms of biological data gathering (e.g., heart-rate, brain activity) will benefit from an automated approach to data preparation by similarly alleviating bottlenecks associated with research areas such as mental health and consumer experience.

8 CONCLUSION

In summary, by automatically classifying raw eye tracking data by temporal relevance and weighted proximity to areas of interest, the workload of research using eye tracking as a metric can be significantly reduced. Eye-tracking intrinsically poses challenge as it is difficult in directly attributing attention to what is being looked at. In this project a framework has been developed which not only extracts and maps the eye gaze data to a plane but also provides useful insights into the interests. In doing so we can significantly expedite the process of visual display research, allowing novel technology such as vehicular AR displays to more quickly approach

commercial readiness. The same techniques could well be used across various other AR applications like design, training, education and health care industry.

8.1 Lessons Learned

The most important aspect of any data analysis is the data pre-processing. It takes a lot of effort to make the data suitable to be operated upon by any statistical learning technique. In this project the sophisticated methods of computational photography described at length in the earlier sections hold key to calibration accuracy, that is, how exactly the position of participant 's gaze is projected into world coordinates.

If given an opportunity again author would utilize the time to carry out detailed experiments with multiple AOIs. The exercise will go a long way to differentiate performance of k-means with respect to multi-class SVM.

REFERENCES

[1] Alex Poole and Linden J Ball 2006,Eye tacking in HCI and usability research. ***Encyclopedia of Human Computer Interaction*** IGI Global,211-219
[2] BeGaze Manual Version 3.0. http://twiki.cis.rit.edu/twiki/pub/MVRL/SmiTracker/begaze2.pdf
[3] Driving Simulator Manual .https://www.nads-sc.uiowa.edu/minisim/
[4] J.Black et. al. 2002, Multi view image surveillance and tracking ,Workshop on Motion and Video Computing.
[5] Kenichi Kanatani 1998. Optimal Homography Computation with a Reliability Measure , IAPR Workshop on Machine Vision Application 426-429
[6] Zhongfei Zhang and Allen Hanson . 3D Reconstruction Based on Homography Mapping