8-2010

Event-driven Similarity and Classification of Scanpaths

Thomas Grindinger
Clemson University, tgrindinger@gmail.com

EVENT-DRIVEN SIMILARITY AND CLASSIFICATION OF SCANPATHS

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Computer Science

---

by
Thomas Grindinger
August 2010

Accepted by:
Dr. Andrew T. Duchowski, Committee Chair
Dr. Donald H. House
Dr. Pradip K. Srimani
Dr. Anand K. Gramopadhye

Abstract

Eye tracking experiments often involve recording the pattern of deployment of visual attention over the stimulus as viewers perform a given task (e.g., visual search). It is useful in training applications, for example, to make available an expert's sequence of eye movements, or scanpath, to novices for their inspection and subsequent learning. It may also be potentially useful to be able to assess the conformance of the novice's scanpath to that of the expert. A computational tool is proposed that provides a framework for performing such classification, based on the use of a probabilistic machine learning algorithm. The approach was influenced by the need to compute similarity of eye fixations at single points in time, such as would be required for video stimuli. This method is also useful for eye movement analysis over static images and some interactive tasks. The algorithm employs a common qualitative comparison method, the heatmap, in a quantitative way to measure deviation from group aggregate behavior. This quantitative comparison is performed at individual events, defined by the stimulus, such as frame timestamps of video or mouseclicks of interactive tasks. The algorithm is evaluated and found to be more accurate and discriminative than existing comparison algorithms for the stimuli used in the examined experiments.

Acknowledgments

Special thanks belong to the members of my doctoral committee: Dr. Duchowski, Dr. Srimani, and Dr. House from the School of Computing, as well as Dr. Gramopadhye from the Industrial Engineering department. I appreciate the suggestions, criticisms, and communication I have received from you all.

I would like to especially thank Dr. Duchowski for serving as adviser for both my master's thesis and doctoral dissertation. He has given me mostly free reign to explore my area of interest and how it could be applied to eye tracking. He has also helped me to understand research better and to know what aspects of research I find most appealing.

Dr. Feng Luo certainly deserves acknowledgment, as well. Under his direction, I received my first conference publication. I experienced a different, and valuable, experience in research while working with him, which I have grown to appreciate and incorporate into my own style.

Finally, I would like to thank two colleagues at Clemson University: Adam Whitley and Bo Li. They have both been eager to discuss research and consider some of the hard questions we were dealing with. Whether the topic was personal or professional, they always had something useful to offer.

Table of Contents

Table of Contents (Continued)

Page

List of Figures

List of Figures (Continued)

Page

Chapter 1

Introduction

Eye tracking is a valuable method of visualization and analysis of a viewer's (or a group of viewers') distribution of visual attention. Fast eye movements (saccades) reposition the fovea, the area of highest resolvability, over objects in the visual field (or Regions Of Interest, ROIs) for closer scrutiny. In contrast to saccades, mainly stationary eye movement periods (fixations) are indicators of cognitive processing of the object under inspection. A sequence of fixations is known as a scanpath (Figure 1.1 shows the raw data and conversion to scanpath). It can be assumed that visual attention follows the fovea, although this is not always the case (one can covertly attend to an object in their periphery but must do so willfully; peripheral visual attention is immeasurable and unlikely without overt effort in most unrehearsed tasks [Kramer and McCarley 2003]). Consequently *scanpaths* are traces of what a viewer overtly attended to in a scene. They have been used for compelling visualizations since the early 1970s, but have as yet not been fully exploited for their quantitative potential.

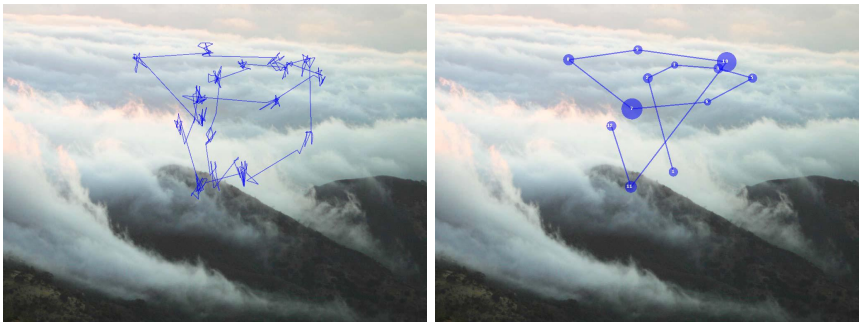


Figure 1.1: Raw scan data converted to a scanpath in which each circle represents a fixation with the diameter relative to the duration.

There are many applications in which tracking and quantifying individuals' reactions to visual stimulus is valuable. One such example is that of a complex training

task. A basic training paradigm consists of an expert overseeing the activity of one or more novices under his tutelage. The expert relies on many learning cues which give important feedback on future training direction. Nair et al. [2001] have shown that eye tracking is a useful cue for these kinds of tasks, as eye movement recordings afford process measurement along several different metrics (e.g., fixations, fixation durations, etc.). One comparative metric that has not been studied extensively is expert/novice similarity.

Expert/novice classification metrics are actively sought simultaneously by Computer-Human Interaction researchers, training practitioners, as well as eye tracking users and developers. The desire for a quantitative metric for expert/novice classification is driven mainly by process training applications. Within this context, proper evaluation of training patterns and development is essential to those who oversee an organization's training programs.

There are many ways in which expert and novice scanpaths can be compared. Depending on the task, certain metrics may be more useful than others. A robust comparison metric should reliably classify an input scanpath as a member of some specified group, such as experts or novices, with high accuracy. Since the most accurate way of classifying these two scanpath classes may vary per task, the most robust method of classification is one that adapts to the stimulus used, whether it be a still image, video, or an interactive task. The visual activity of the observers can be used to extract useful information, independent of the content of the stimulus itself.

Distinguishing between expert and novice can be accomplished via a two-step process. Given a predefined set of experts and novices, the scanpath in question is compared to each group to determine how similar it is to that group. These two

similarity scores are then compared to the distribution of expected similarity scores for each class distribution, from which we may determine the probability that the scanpath in question is truly expert or novice. This final value serves well as a "group-wise similarity".

The approach is thus two-fold. First, an appropriate method must be derived for comparing a single scanpath to a group of scanpaths. Second, an analysis framework must be constructed that will provide a robust mechanism for determining which scores belong to which class.

The first problem is approached from the perspective of a collection of samples "per event" for a given stimulus. At each event (e.g., display of a video frame), the fixation points for all scanpaths are collected. The scanpath in question is then compared to the collected fixation points for the different classes and weighted by the distance from the various fixations. The weights for the fixations from the scanpath in question are summed up over all the events to obtain a final similarity to all the predefined scanpaths.

The second problem is solved through use of Receiver Operating Characteristic (ROC) analysis. This statistical framework is commonly used in machine learning applications. The ROC curve, in particular, provides an elegant means of evaluating the probability of arbitrary values' membership in known classes. It also allows for a secondary indication of reliability, described in Section 3.1.4.

The derivation of a robust classification metric poses several challenges. First, the metric should be able to compute correlation-like values in terms of the features used, whether they be location, area, order, and/or duration. Second, the metric should provide levels of statistical significance. Third, this computation should be

automatic, relying on a robust clustering approach that does not rely on *a priori* estimation of the number of clusters. Fourth, cluster overlap should be flexible, allowing fixation cluster comparison in both space and time. Fifth, computation of an "average scanpath function" should also be automatic, providing an idealized function serving as the ground truth or reference for comparison. The proposed classification algorithm satisfies these criteria in a novel way. The development of completed classification and scanpath comparison computations is evaluated empirically.

Chapter 2

Background

Scanpath comparison can be classified as either *content-*, or *data-driven.* The former is largely based on Regions Of Interest, or ROIs, identified *a priori* in the stimulus and subsequently by associating those regions with fixations. Thus, any form of analysis or comparison between scanpaths is made in terms of regions or image elements fixated by the viewer. The latter approach, in contrast, is made on scanpaths directly, independent of whatever was presented as the stimulus. An important advantage of the latter form of analysis is that it can be applied to the (x, y, t) eye movement streams directly, without the need of establishing a reference frame within which the ROI stipulation must take place.

Consider two recent approaches to the scanpath comparison problem. Jarodzka et al.'s [2010] vector-based similarity measure is an example of a content-driven approach since it relies on the quantization of the stimulus frame into an arbitrarily-sized 5×5 grid which serves as the method's source of ROI labeling. A label is added to the scanpath stream whenever a fixation is present within a grid cell. In contrast, Duchowski et al.'s [2010] revisitation of Privitera and Stark's [2000] string-editing approach is an example of a data-driven approach since it operates directly on the scanpaths. String (ROI) labels are determined by overlapping fixation clusters. Both approaches consider fixation durations and are therefore potentially suitable for analysis of gaze collected over dynamic media, however, their means of scanpath aggregation are derived from pairwise vector or string comparisons. For groups of viewers, additional organization is required to derive aggregate statistics. Furthermore, Privitera and Stark's [2000] use of a string-editing procedure to compare the sequential loci of scanpaths did not distinguish between fixations of different du-

rations, which is useful for class discrimination. The present approach builds on this framework, but at a temporal level, introducing an *event-driven* concept, where events are points in time, allowing for comparison with groups of scanpaths.

The basic derivation of the string-editing metric follows. The string-editing distance, also referred to as the Levenshtein distance, is derived from a dynamic programming approach. A two-dimensional matrix is constructed to represent possible modifications to the two strings being compared, which is then traversed to find the optimal alignment (in terms of transformation cost). An example matrix for the strings *afbffdcdf* and *abcfeffgdc* is displayed in Figure 2.1. The values in the matrix entries represent the cost required to perform a transformation. Substituting, inserting, or deleting a character each have cost 1. The matrix is filled by rows from top to bottom, accumulating transformation costs. For instance, in the upper-left entry, the value is 0 because there is no transformation needed. In the entry immediately to the right, there is transformation cost 1, since that entry corresponds to an insertion of the character *f* in the second string. Similarly, the entry immediately below the upper-left 0 corresponds to an insertion in the first string. The diagonal entry corresponds to a possible substitution.

	a	f	b	f	f	d	c	d	f
a	0	1	2	3	4	5	6	7	8
b	1	1	1	2	3	4	5	6	7
c	2	2	2	2	3	4	4	5	6
f	3	2	3	2	2	3	4	5	5
e	4	3	3	3	3	3	4	5	6
f	5	4	4	3	3	4	4	5	5
f	6	5	5	4	3	4	5	5	5
g	7	6	6	5	4	4	5	6	6
d	8	7	7	6	5	4	5	5	6
c	9	8	8	7	6	5	4	5	6

Figure 2.1: Example of Levenshtein distance calculation.

The alignment procedure is demonstrated in Figure 2.2. On the final line, the real alignment is displayed, along with the total distance from the first string to the second. This is the most useful part of the string-editing distance algorithm. For character data, this alignment is sufficient. Hembrooke et al. [2006] used a multiple sequence alignment algorithm to create an average scan path for multiple viewers, providing some functionality lacking in the previous work. Unfortunately, their procedure was never explained in detail, and no objective results were provided. When calculating similarity of scanpaths, though, this alignment requires a high-level analysis to determine regions of interest, which are then labeled with a character. A scanpath, originally a fairly long sequence of coordinate pairs, is then converted into a much shorter sequence of ROI labels. This may be done through clustering techniques or designation of predefined, explicit ROIs.

One objective of the present work is to integrate this high-level alignment with the lower-level algorithm that will be described, allowing it to be used for a wider range of problems. Some visual tasks involve multiple steps that need to be completely separately. In some instances, the problems may take different lengths of time to complete for different individuals. The high-level string-editing alignment would provide a guideline, of a sort, within which to perform a more comprehensive, lower-level analysis. Also, since it is the most widely-accepted form of scanpath comparison, it serves well as a baseline to compare new algorithms to.

Goldberg et al. [2002] demonstrated several ways in which high-level scanpath data may be used to analyze stimuli. Their study was focused on the general characteristics of scanpaths around specific areas of a stimulus. Bednarik et al. [2005] used similar high-level data to construct a transition matrix that described the probability that a subject would look from one given area of interest to another. Their study, on eye tracking during program visualization, showed no significant correlation be-

$s_1 = abcfeffgdc$ $s_2 = afbffdcdf$	start	cost 0
$s_1 = abcfeffgdc$ $s_2 = afeffdcdf$	substitution of first b by e	cost 1
$s_1 = abcfeffgdc$ $s_2 = abcfeffdcdf$	insertion of bc after first a	cost 2
$s_1 = abcfeffgdc$ $s_2 = abcfeffdc$	deletion of last df	cost 2
$s_1 = abcfeffgdc$ $s_2 = abcfeffgdc$	insertion of g	cost 1
$s_1 = abcfeffgdc--$ $s_2 = a--fbff-dcdf$	alignment	total distance 6

Figure 2.2: Example of string editing. Adapted from Privitera and Stark [2000].

tween the high-level data they extracted and program comprehension. Fischer and Peinsipp-Byma [2007] combined both high-level data and transition matrices, extracted from a uniform grid sampling of a stimulus, to describe scanpaths. Their intention was similar to the objective of Goldberg et al. [2002], in that they were attempting to evaluate perception of elements of a stimulus. None of these works actually compare high-level data from one individual or group of individuals to another.

Duchowski and McCormick [1998] described a visualization which tracks fixations through time, referred to as "volumes of interest". They were able to visualize multiple scanpaths in two and three dimensions, using this temporal mapping. The two-dimensional visualization plots x or y components on the y-axis and time on the x-axis, while the three-dimensional visualization depicts scanpaths as uniform-width volumes, where time serves as the third dimension. Räihä et al. [2005] described a similar visualization in two dimensions, with the slight difference that fixations were displayed as variable-size circles, congruent with the typical visualization of

scanpaths. Heatmap visualizations, as described by Pomplun et al. [1996] and popularized by Wooding [2008], overlay attentional information onto a stimulus as colors, where hot colors correspond to regions of high interest and cold (or no) colors correspond to regions of low interest. This representation is highly informative, yet does not provide any quantitative information. The concept of heatmaps is utilized in the present algorithm, but they are not aggregated over time.

The present similarity measure resembles somewhat the Earth Mover's Distance used by Dempere-Marco et al. [2006] when considering cognitive processes underlying visual search of medical images. The approach is also similar to Galgani et al.'s [2009] effort to diagnose ADHD through eye tracking data. They created three classifiers, including a classifier based on Levenshtein distance, and discovered that Levenshtein's gave the best results among their chosen algorithms. To show relative improvement, the performance of the algorithm is compared to a similar Levenshtein classifier.

Torstling [2007] demonstrated an application of machine learning to eye tracking, where a generative model was constructed from individual scanpaths collected over multiple images. The classifier is then trained and evaluated on its ability to predict which image the scanpath was created from. This is quite similar to the intent of the present work, aside from the fact that the previous attempted to predict stimulus, whereas the present work attempts to predict subject attributes (e.g., expert/novice).

The current approach was initially described in Grindinger et al. [2010]. The basic framework of the algorithm has not changed since that initial publication, though several of the details have been adjusted, extended, and evaluated. The major missing element of that publication was analysis over video stimuli, which will

be evaluated in this work, as well as application to interactive stimuli.

Chapter 3

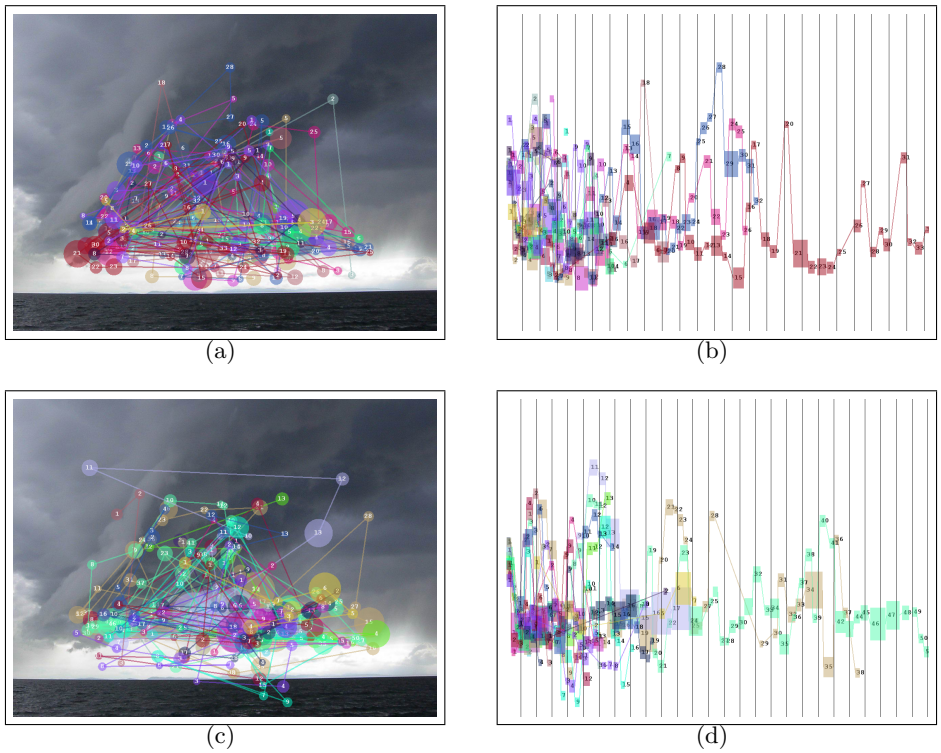
Still Image Stimulus Analysis and Classification

The scanpath classification algorithm takes as input two collections of fixation-filtered scanpaths. An example image is presented in Figure 3.1, displayed in 3.1(a) with all novice scanpaths and in 3.1(c) with all expert scanpaths. From a simple visual examination, there is no obvious characteristic that stands out for either collection. A procedure is then needed to perform a deeper statistical analysis of each collection.

3.1 Classification Framework

A classifier is a function that accepts training data and a similarity measure as input and produces a means of discriminating between classes of data present in the training data. Most classifiers produce some form of threshold, which is used to determine whether new data are members of the class the specific classifier was trained for, using the similarity measure. There are three steps to building and evaluating the classifier for this particular problem. Other forms of classifiers may be constructed differently. First, similarity scores need to be extracted from the training data. Second, the threshold for optimal discrimination needs to be computed, based on the similarity scores computed in the first step. Third, the classifier must be validated to determine its reliability.

A classifier may be defined as a function $C(X_t, X_r)$, where X_t is the set of test data to be used in evaluating the reliability of the classifier constructed with the set of



(a) (b)

(c) (d)

Figure 3.1: Collections of scanpaths for a single stimulus. Typical novice scanpaths visualized in (a), experts in (c). Collection of time-projected novice scanpaths in (b), and experts in (d), which can be considered side views of the three-dimensional data.

training data X_r. These two sets are necessarily disjoint. The output of this function varies as needed, but at its most simplistic, yields the accuracy of classifications of instances in the X_t data set. Accuracy may be measured in multiple ways, but the most common and intuitive measurement is the ratio of correctly-classified instances to total instances.

3.1.1 Eye Movement Analysis

Comparison of scanpaths captured over static media depends on the identification of fixations within the raw gaze point data stream. Raw eye movement data tends to be noisy, as it represents a conjugate eye movement signal, composed of a rapidly changing component (generated by fast saccadic eye movements) with a comparatively stationary component representative of fixations (the eye movements generally associated with cognitive processing). In most diagnostic applications, all but fixations are removed from the signal by either of two leading methods for fixation detection: the *position-variance* or *velocity-based* approaches [Duchowski 2007].

The former defines fixations spatially, with centroid and variance indicating spatial distribution [Anliker 1976]. If the variance of a given point is above some threshold, then that point is considered outside of any fixation cluster and is considered to be part of a saccade. Fixation identification can be implemented by a variant of the mean-shift algorithm [Santella and DeCarlo 2004], where fixations $\mathbf{s}(\mathbf{x}_i)$ can be thought of as ellipsoids in time with spatial and temporal extent, weighted by clusters of nearby raw gaze points, where $\mathbf{s}(\mathbf{x}_i)$ is iteratively determined by repeatedly shifting it to a new location based on a kernel function K. The kernel can be modeled by a zero-mean spatiotemporal Gaussian kernel,

$$K([\mathbf{x}_i, t_i]) = \exp\left(\frac{x_i^2 + y_i^2}{\sigma_s^2} + \frac{t_i^2}{\sigma_t^2}\right)$$

with the i^{th} raw gaze point denoted as $\mathbf{x}_i = (x_i, y_i, t_i)$, and σ_s and σ_t determining local support of the kernel in both spatial (dispersion) and temporal extent. The user-adjustable parameters σ_s and σ_t can be epistemically set to match the extent of the foveolar dimension of the human retina (e.g., $\sigma_s = 50$ pixels at screen resolution 1280×1024 constitutes 1.5° visual angle at 50 cm viewing distance with $\sigma_t = 500$ ms set to an expected average fixation duration).

The latter (velocity-based) approach, which could be considered a dual of the former, examines the velocity of a gaze point, e.g., via differential filtering, $\dot{\mathbf{x}}_i = \frac{1}{\Delta t} \sum_{j=0}^{k} \mathbf{x}_{i+j} \mathbf{g}_j,\ i \in [0, n-k)$, where k is the filter length, $\Delta t = k - i$. A 2-tap filter with coefficients $\mathbf{g}_j = \{1, -1\}$, while noisy, can produce acceptable results. The point $\mathbf{x}_i$ is considered to be a saccade if its velocity $\dot{\mathbf{x}}_i$ is above threshold [Duchowski et al. 2002]. It is possible to combine these methods by either checking the two threshold detector outputs (e.g., for agreement) or by deriving the state-probability estimates, e.g., via Hidden Markov Models [Salvucci and Goldberg 2000].

Presently, a variant of the position-variance algorithm is used with a spatial deviation threshold of 30 pixels and the number of samples set to 5 (implying a temporal threshold of 100 ms at a 50 Hz sampling rate). The fixation analysis code is freely available on the web.[1]

3.1.2 Scanpath Comparison

The similarity measure is the backbone of the classifier. The framework outlined by Grindinger et al. [2010] was motivated by the need for analysis of eye tracking

[1]The position-variance fixation analysis code was originally made available by LC Technologies. The original `fixfunc.c` can still be found on Andrew R. Freed's eye tracking web page: <http://freedville.com/professional/thesis/eyetrack-readme.html>. The C++ interface and implementation ported from C by Mike Ashmore are available at: <http://andrewd.ces.clemson.edu/courses/cpsc412/fall08>.

data collected over video stimuli. The motivation of the approach is to track the deviation of an individual's eye movements from the average of two or more sets of scanpaths that have already been classified as members of well-defined classes. The first experiment that is used to demonstrate validity involves two such classes, namely experts and novices, and uses the approach to show that class members could be reliably classified by their eye movements alone. Although intended for analysis of video stimuli, the approach is first utilized for static images. The algorithm is described in relation to the video paradigm, however. For the purposes of this first experiment, it may be helpful to conceptualize a stimulus consisting of a static image to be a video clip in which the image frame is periodically repeated, or extended, in time.

The approach to video stimuli encapsulates an "event-driven" paradigm. Namely, the events consist of display of individual video frames. This event-driven approach is expected to be useful for other forms of stimuli, as well. The defining event for a given stimulus may be defined on a case-by-case basis. In the case of image stimuli, the events are defined to be intervals of time, specifically 33 millisecond intervals, in accordance with current video playback speeds. A study involving tasks consisting of multiple steps treats the intermediate steps as events, as described in Section 5.

Dynamic stimuli, such as videos, may each be considered to be simply a collection of individual stimuli, namely frames. Scanpath similarity metrics developed for static stimuli can thus be applied on a frame-by-frame basis and aggregated in some way (e.g., averaged). The trouble with prior vector- or string-based approaches, however, is that aggregation is based on pairwise comparisons. This leads to rather complicated bookkeeping requirements for organizing pairs, e.g., labeling each pair as *local*, *repetitive*, *idiosyncratic*, or *global* based on the dyadic permutations of viewer and stimulus (idiosyncratic, for example, refers to scanpaths made

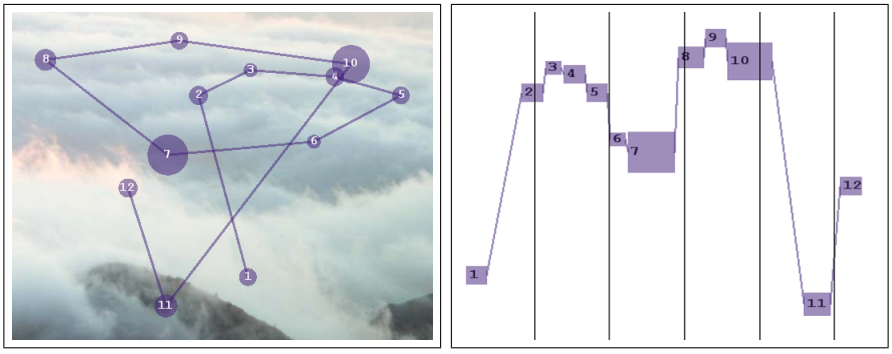


Figure 3.2: Typical scanpath visualization at left. Time-projected scanpath visualization at right, where the y-axis denotes vertical gaze position, but the x-axis denotes time. Fixation labels are common between the two. Markers denote one-second intervals.

by the same viewer over different stimuli).

The projection of fixations $\mathbf{x}$ onto individual frames and clustering produces disks of variable radius, as is commonly seen in traditional scanpath visualizations, an example of which is depicted in Figure 3.2. Alternatively, depositing each fixation $\mathbf{x}$ onto the video frame by once again using a Gaussian kernel to weight the pixel intensity produces the well-known heatmap, an alternative to scanpath visualization (see Figures 3.3 and §4.2 below).

The present approach projects gaze data onto video frames as heatmaps, sampling scanpaths recorded over video frames, and measures the deviation of a scanpath of unknown classification from a set of scanpaths which has already been classified. Each frame is composed of a sampled set of fixations, with as many sets as there are scanpath classes defined. A per-frame similarity measure is then derived and averaged over the duration of the video sequence to compute the total similarity of an unclassified scanpath to the two or more sets of classified scanpaths. Statistical analysis determines which class a scanpath is a member of, based on its similarity score to each defined class.

Using video as the temporal reference frame, a scanpath is parametrized by a two-dimensional function $f(s,t)$, where s is the scanpath defined by its collection of fixations, and t is the frame number at which the data was collected. The function f simply returns the fixation at frame t. This definition is trivially extended to $f(S,t)$, where S is a set of scanpaths, t is still the frame number, but f now returns a collection of fixations from the scanpath set.

Following heatmap generation, with each fixation conceptualized as a Gaussian region of interest, where μ denotes the mean of the Gaussian coinciding with the location of the fixation and σ models the error of the eye tracker, the similarity of a single fixation to one of these Gaussian regions is given as:

$$g(\mathbf{x},\mu) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\mathbf{x}-\mu)^2}{2\sigma^2}\right),$$

where $\mathbf{x}$ denotes the unclassified fixation. Denoted here as the Gaussian similarity, g can thus be thought of as the probability of a fixation being a member of the region of interest, as illustrated in Figure 3.3. This value does not directly describe the probability that the fixation is a member of the class. Instead, it provides an intuitive description of the similarity of that fixation with the class it is being compared to. A fixation far away from any fixations in the class would be expected to have low similarity, whereas a fixation close to fixations in the class would be expected to have much higher similarity. This Gaussian function returns values that directly correlate with this concept of similarity.

The similarity of a single fixation to a set of fixations is:

$$d(s,S,t) = \sum_{\mathbf{x}\in f(S,t)} w(\mathbf{x}) g(f(s,t),\mathbf{x}),$$

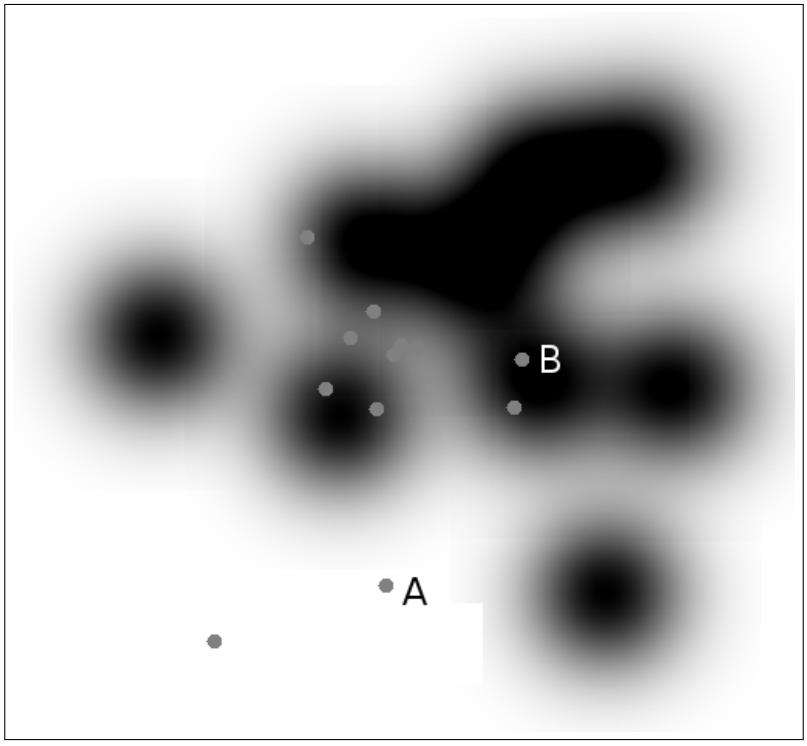


Figure 3.3: Mixture of Gaussians for a classified set of fixations at a discrete time-stamp. Displayed unclassified fixations (labeled gray circles) were not used in the heatmap generation. Note that the fixation labeled 'A' is far from any Gaussian center, and thus has lower similarity than fixation labeled 'B'.

where s is the unknown scanpath, S is the set of classified scanpaths being compared against, t is the frame number, and $w(\mathbf{x})$ is a weighting factor that is set to $1/|S|$ as a simple means of similarity score normalization.

The measure $d(s, S, t)$ is evaluated over the entire video sequence to estimate the mean similarity of a scanpath to the set of scanpaths, $d(s, S) = \overline{d(s, S, t)}, \quad t \in T$, where t represents individual frame numbers and T is the collection of frame numbers for the entire sequence. The similarity of a single scanpath to a set of scanpaths is the average frame similarity over the duration of the video. The resultant score lies between 0 and 1, and tends to fall near 0. The value of the score, however, is not as important as the probability that the score lies within the expected distribution of scores for a specific class.

3.1.3 Temporal Normalization

In the case of still images, wherein the participant has control over how long he wishes to view the stimulus before continuing on, scanpaths of different length must be handled properly. Portions of a scanpath that do not contain any fixations cannot reasonably be compared to a set of scanpaths that do contain fixations. It is possible to assign some "penalty" value to these "blank" portions of the scanpaths, but it is not immediately justifiable why this should be done. In this study, only portions of scanpaths which contain fixations are used in the comparison. It follows that a scanpath of a specific duration may not be compared beyond its duration.

Some scanpaths last significantly longer than other scanpaths. It would not be reasonable to determine classification of a scanpath by comparing it against a very small minority of scanpaths in a group. There needs to be some cut-off point to prevent effectively over-training the classifier. The average scanpath length serves

this purpose. Given a scanpath that is longer than the average scanpath length of a class of scanpaths, the similarity algorithm will only use data up to the average length in its calculations. Any information beyond that length is discarded. Similarly, given a scanpath that is shorter than the average length, only data up to the length of that scanpath is used. This heuristic is only used in the case of still image stimuli. Scanpaths collected over video have the same length, assuming the subjects actually watched the entire video clip.

3.1.4 Classification

The similarity measure provides a means of estimating the deviation of a single or set of scanpaths from a group of scanpaths with known classification. These similarity scores serve as input to the classification mechanism that is responsible for estimation of an optimal threshold that determines whether a scanpath of unknown classification is accepted by that classifier or not. Given a scanpath's similarity, scored against a set of scanpaths of known classification, the scanpath is accepted by the classifier if the similarity score is higher than the computed threshold or rejected if it is lower.

The training data for the classifier generally consists of multiple classes, very often two, but possibly more. The classifiers generated by the current approach are each trained to a single class. Non-class training data is still used as input, however. Each classifier is specialized to recognize members of the class it was trained for. Multiple classifiers may be used together to reinforce the classification decisions, as discussed below.

The classification approach estimates the distribution of similarity scores for a given class. To do so, the similarity of each known member of the class is computed

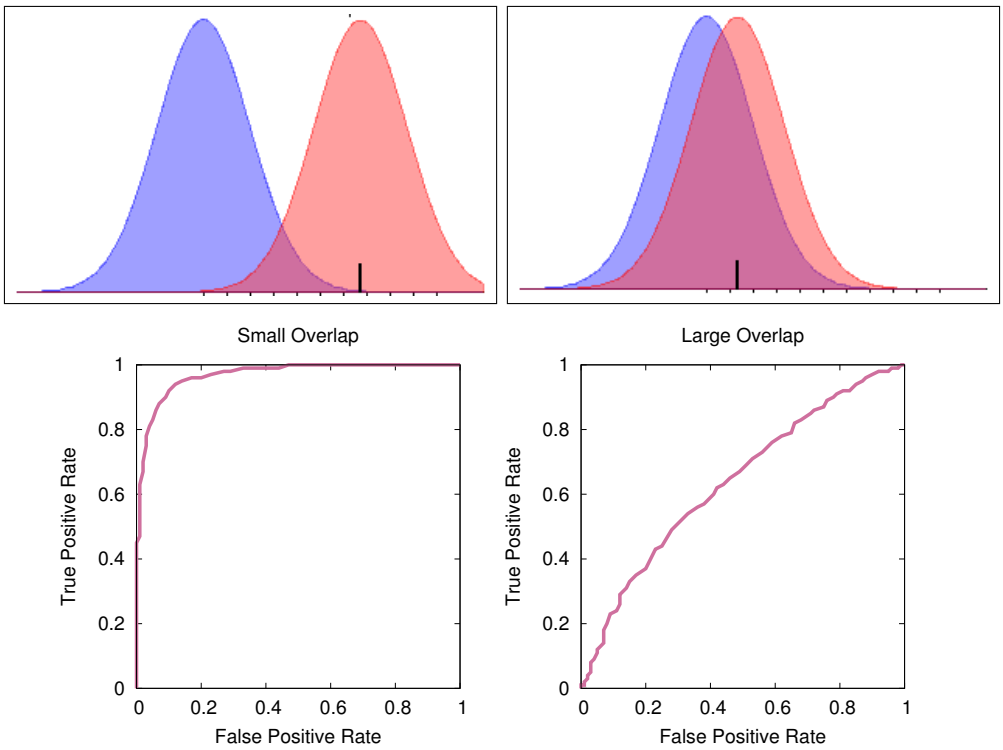


Figure 3.4: Examples of small and large distribution overlap with ROC curves that are similar to what these distributions would be expected to yield. The ROC curve is extracted by sliding threshold along x-axis and calculating true and false positive rates.

in relation to the rest of the class. These scores are compiled together along with scores for known non-members of the class. It is expected that non-members will produce lower similarity than members. For unclassified data, the new score is likely to fall within either the known member distribution or the known non-member distribution. This property is visualized in Figure 3.4. Classification results tend to be significantly better for distributions with small overlap than for those distributions with large overlap.

Analysis relies on the use of Receiver Operating Characteristic, or ROC, curves. For a score distribution, such as either of those depicted in Figure 3.4, an ROC curve is constructed by setting a threshold for class acceptance at the left end of the scale

and gradually sliding it to the right. Every time a threshold is evaluated, the true positive and false positive rates are recorded.

The ROC curve provides two convenient facilities. First, it easily facilitates the choice of an optimal threshold by picking the point closest to $(0, 1)$, where the balance of false positives to true positives is optimal. The area under the curve (AUC) also indicates the discriminative capability of a classifier. Ideally, the AUC should equal unity (1), while a completely random classifier would yield AUC close to 0.5. According to Swets [1988], values between 0.5 and 0.7 are generally considered to be uninformative, while values above 0.9 are considered highly informative. Values between 0.7 and 0.9 are not optimal, but are very common and far more acceptable than values that are lower. The AUC value represents the probability that some arbitrarily-chosen class member is given a similarity score higher than some arbitrarily-chosen non-class member.

It is possible, and somewhat common, for classifiers to have AUC value less than 0.5, such as in Parker et al. [2007]. Since a random classifier should have AUC value 0.5, a classifier with AUC value less than 0.5 could be said to be "worse" than random. This means that the classifier is consistently making the wrong decisions, instead of randomly correct and incorrect decisions. A common cause of this occurrence is that the distribution of positive class member scores has a lower mean than the negative distribution. In training data, this can be trivially accounted for by inverting the results of the classifier. If some instance would have been given classification score 0.7, that score then becomes 0.3. For cross-validation purposes, an AUC value of less than 0.5 for the testing data is unable to be corrected, aside from modification of the classification algorithm. The purpose of cross-validation is to estimate the accuracy of the classifier, given unknown data. In practice, such as in a study intended to determine whether unlabeled subjects are experts or not, it is im-

possible to compute an AUC value at all, since the *a priori* classification is unknown.

The ROC curve provides appropriate thresholds for class acceptance and rejection, but in cases where more than one class exists, multiple classifiers may reinforce the classification decisions. Two classifiers are used for the experiment described in this chapter: one for experts and another for novices. The only difference between the two classifiers is how the data are labeled. These separate classifiers are combined into one "meta-classifier," resulting in a single classifier with better accuracy and discriminative ability than either of its components. Instances of the class a classifier is intended for are considered "positive," while data from other classes are considered "negative."

The labeling of instances as positive or negative provides the ability to evaluate the performance of the classifier, based on its ability to correctly predict positive instances (true positive rate) and avoid accepting negative instances (true negative rate). In addition, the inaccurate behavior of a classifier may be described by its false negative rate and false positive rate. A comprehensive discussion on the following metrics may be found in Olson and Delen [2008]. The true positive rate of a classifier is also referred to as the classifier's recall or sensitivity and is equal to the number of true positives divided by the number of total positive instances, or equivalently, the number of true positives divided by the sum of the number of true positives and false negatives:

$$Recall = \frac{tp}{tp + fn}.$$

The precision of a classifier describes the rate at which instances predicted as positive are actually a member of the positive class. This value is calculated by

dividing the number of true positives by the sum of the number of true positives and false positives:

$$Precision = \frac{tp}{tp + fp}.$$

The true negative rate, also referred to as the specificity, is similar to the precision, in that it describes the rate at which instances predicted as negative are actually negative. It is computed similarly to the precision:

$$Specificity = \frac{tn}{tn + fp}.$$

Finally, the accuracy of a classifier is the rate at which instances are classified correctly. Since true positives and true negatives describe the number of correct classifications, the accuracy is simply the sum of the true positives and true negatives divided by the sum of all classifications:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}.$$

This metric will be used most often in the results, in addition to the AUC value described earlier. Accuracy is the most intuitive metric for evaluating classifiers, though it can be misleading. The AUC gives a comprehensive impression of the precision, recall, and specificity in a single value, though it is slightly more difficult to intuitively grasp than simple accuracy.

If only one "sub-classifier" were used, classification decisions would be trivial. If the similarity score exceeds the computed optimal threshold, the instance is accepted. Otherwise, it is rejected. The results of the acceptances or rejections of both classifiers must be interpreted correctly. When one classifier accepts and the

other rejects, the final classification decision is trivial. If both classifiers reject or both accept, the similarity score is divided by the threshold value. Classification is then attributed to the class with the highest value that resulted from the previous operation. Analogously, a prediction value may be constructed by dividing the output of the positive classifier by the sum of the output of the positive and negative classifiers:

$$Pred(s, C) = \frac{Pred(s, P)}{Pred(s, P) + Pred(s, N)},$$

where $Pred(s, X)$ is the prediction value of s, with respect to classifier X. C, P, and N are the combined, positive, and negative classifiers, respectively. Values greater than 0.5 are then classified as positive, while values lower than or equal to 0.5 are classified as negative. In this way, it is possible to estimate the reliability of a classification, since values close to 0 or 1 indicate large differences between results for each of the sub-classifiers and values close to 0.5 indicate very small differences.

3.1.5 Cross-Validation

Establishing the validity of classification generally uses some subset of all available data as training data and a smaller subset as testing data. One-third of the training data is often used as test data, and all instances are chosen randomly. This would be the preferable means of establishing validity of the classifier trained on the eye tracking data collected in this study, had the data set not been so small. Classes have approximately twenty members each, smaller than most other classification problems, which tend to operate on data sets numbering in the thousands. Setting aside one-third of this data would most likely affect the reliability of the classifications. Cross-validation is meant to address this problem.

In order to establish validity of classifications, cross-validation is performed. In

this step, results are validated by executing the algorithm multiple times on different combinations of data. For a given trial, if there are n records, one is left out and used to evaluate accuracy, based on the remaining $n-1$ records. Subsequently, another record is rotated out for testing. In this way, one data set may be recombined into n data sets. Accuracy is then the percentage of trials that successfully classified the test case left out. This describes the traditional leave-one-out cross-validation, or LOOCV for short.

Although the goal of cross-validation is the estimation of the reliability of the classification threshold, which relies on the selection of the point on the ROC curve closest to $(0,1)$ (where the ratio of false positives to true positives is balanced) for the training data, unfortunately, LOOCV precludes computation of an ROC curve for the testing data. Nevertheless, it is still possible to estimate the AUC for the testing data, without explicit computation of the ROC curve, since the AUC represents the probability that some arbitrary positive class member will be given a higher similarity score, also called a prediction in machine learning terminology, than some arbitrary negative class member.

To allow estimation of the AUC, LOOCV is replaced by leave-pair-out cross-validation, or LPOCV, following Airola et al. [2009]. Instead of holding one item out of the training set, two are held out: one from the positive class and one from the negative class. The AUC is then defined:

$$AUC = \frac{1}{|X_+||X_-|} \sum_{s_i \in X_+} \sum_{s_j \in X_-} H(C_{\overline{\{i,j\}}}(s_i) - C_{\overline{\{i,j\}}}(s_j)),$$

where $X_+ \subset X$ and $X_- \subset X$ are the positive and negative instances in the training set X_r, and $C_{\overline{\{i,j\}}}(s_i)$ is a classifier trained without the i^{th} and j^{th} training examples, and $H(x)$ is the Heaviside step function, which returns 1 when $x > 0$, 0.5 when

$x = 0$, and 0 when $x < 0$.

In this way, each cross-validated test case contributes to an approximation of the probability that an arbitrary positive instance will be scored higher than an arbitrary negative instance. This modification artificially lowers the AUC estimate, since the classifiers are being trained with less data than LOOCV. This effect is expected to be insignificant, since the impact on the training set is still small. Airola et al. [2009] conclude that this method of approximating the AUC is more reliable than other approximation techniques, especially for small data sets with low signal-to-noise ratio.

In experiments consisting of multiple stimuli, all of which are intended for the same visual task, the multiple classifiers may be used to reinforce the classification for a given individual, similar to the boosting mechanism described in Kearns [1988]. For instance, given an expert/novice study, in which subjects are asked to perform the same visual task for a number of stimuli, a single classifier may have a certain level of expected accuracy, based on testing data, perhaps 70%. Reinforcing the classification of an individual with several classifiers, each with 70% expected accuracy, would then yield a decision that is accurate with probability much greater than 70%, depending on how many extra classifiers are used and how independent the classifiers are.

Unfortunately, the dependencies of the classifiers are unknown and will have an effect on the ability to reinforce the results. If the classifiers were completely independent, it might be expected that the error rate (inverted accuracy) will be reduced exponentially for every classifier added. The 70% accuracy classifiers would each have 30% or 0.3 error rate, decreasing to 0.3^n for n classifiers, or $1 - 0.3^n$ accuracy. This is a best-case estimation, assuming independence. Actual accuracy would

be expected to fall between 70% and the optimum estimated reinforced accuracy.

3.2 Empirical Validation

In this study, a sequence of static images was used to elicit scanpaths during a visual training task. Experts and novices for the particular training task were recruited to provide data. This data served as the basis for the machine learning mechanism for the purposes of training (expert) and evaluation (expert and novice), and is described in more detail in Sawyer [2009].

Participants. This study involved 60 participants recruited from Clemson University in Clemson, SC and the surrounding areas. The participants were divided into three equal groups of 20 based on the total number of flight hours each participant had accumulated. The first group consisted of non-pilots who had no prior flight training or experience. The second group consisted of low-time pilots who had accumulated less than 500 total flight hours. The third group included high-time pilots who had accumulated over 500 total flight hours.

Non-pilot subjects were recruited through word of mouth advertising around the Clemson University campus. Of the 20 non-pilot subjects there were nine male and eleven female subjects. The participants were an average age of 25.5 years old with a standard deviation of 4.8 years. The maximum age was 42 and the minimum age was 21. None of the non-pilot subjects had accumulated any flight hours or certifications.

The low-time and high-time pilots were recruited through the Clemson University Flight Club and from flyers at local airports, as well as through word of mouth advertising. The low-time flight group consisted of 17 male and 3 female subjects.

Low-time pilots were an average age of 33.58 years old with a standard deviation of 13.3 years. The maximum age was 53 and the minimum age was 19. The high-time pilot group consisted of 19 male and 1 female subjects. The average age of high-time pilots was 53.4 years old, with a standard deviation of 12.59 years. The maximum age was 75 and the minimum age was 23. A detailed breakdown of mean flight experience is provided by [Sawyer 2009].

Apparatus. A Tobii ET-1750 eye tracking monitor was used to collect all eye tracking data. The ET-1750 provides non-invasive eye tracking on a 17″ monitor. The ET-1750 is able to take samples at a rate of 50 Hz with 0.5° accuracy. For this study the resolution of the monitor was set at 1280 × 1024 pixels. The eye tracking monitor was powered by a Sun W2100z PC with a 2.0 GHz AMD Opteron 246 processor and 2 GB of RAM. Eye tracking data was collected using the software program ClearView 2.7.1 developed by Tobii Technology. Data was then exported as text files for further analysis.

Hypothesis. The experimental hypothesis tested was whether the new machine learning mechanism is capable of classifying the actions recorded during a complex visual search task as expert or novice.

Experimental Design. To test the hypothesis, a visual search task was presented to both novices and experts, whose set of scanpaths served as the independent variable. This particular search task involved evaluation of images of weather of varying degrees of severity.

Procedure. Participants were first given an initial briefing about the nature and goals of this study. They then read and signed an informed consent form. Subjects then completed a basic demographic questionnaire.

Participants were then given an introduction to general aviation and weather decision making. They were then given a description of visual flight rules, including the specific requirements for daytime flight in class G airspace. Participants were then told to assume they were on a cross-country daytime VFR flight as they were shown a series of weather pictures. For each picture participants were told to verbally respond either "yes", signifying that the conditions were above VFR minimums and they would continue their flight, or "no", signifying that the conditions are below VFR minimums and they would divert from their current flight path.

Before beginning the task, subjects were assigned to either view group A or group B images first. Each group contained 10 different images of weather conditions of varying degrees of severity. Both groups had 5 images above VFR minimums and 5 images below VFR minimums.

To begin, the eye-tracking monitor was first calibrated to the task subjects by having them fixate on a blue circle as it moved through a series of nine places on the screen. Subjects were then shown a practice image and asked about the weather conditions to ensure they fully understood the task. After successfully completing the practice attempt, subjects were shown the series of images from the assigned image group. The image order was randomized for each subject within each group. Each image was displayed on the monitor for as much time as was needed for the subject to make a decision. Once the subject responded "yes" or "no", the eye tracking was stopped and their answer was recorded.

Participants then completed the computer based training program *Weatherwise*. Participants were encouraged to take as much time as needed to thoroughly complete the program. Once the program was completed subjects were given a brief

break.

The final portion of the study involved subjects viewing another series of images on the eye tracker. Subjects were given the same briefing and information as in the first task. Participants were first shown the 10 images from the opposite group of the first task, followed by the 10 images from their original group. The order that the images were displayed was randomly generated for each subject. For each image subjects again responded either "yes", that VFR minimums were met, or "no", VFR minimums were not met. When the subject finished the 20th image, the task was complete. Subjects were then thanked for their time, compensated and dismissed.

Expected Outcomes. The expectation was of course that experts and novices could be classified as such with a probability significantly greater than random (0.5). The training effect experienced by the novices during the study was not considered to have a significant effect on the classification results, since the duration was quite small and the experts had extensive experience with such tasks. Furthermore, while accuracy of novice results may have increased, novices did not have sufficient experience to develop automatic search strategies as experts had most likely done.

3.2.1 Results

In order to evaluate the present approach, the results of the study were analyzed, wherein 20 high-time pilots (experts), 20 low-time pilots, and 20 non-pilots (novices) were presented with 20 different images of weather. Subjects were asked to determine whether they would continue their current flight path or if they needed to divert. Their eye movements were recorded by a Tobii ET-1750 eye tracker (their verbal responses were ignored in the analysis). The objective was to produce a classifier that could predict whether a subject is expert or novice, based solely on their

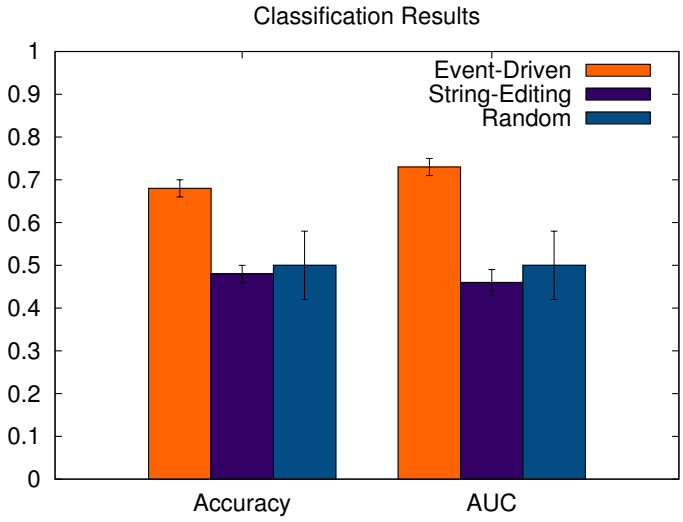


Figure 3.5: Results of classification cross-validation for the new event-driven method, string-editing similarity, and a random classifier's expected performance.

eye movements. The scanpaths for the low-time pilots were not used for the evaluation, since the intention was to demonstrate discriminability, and it is expected that low-time pilots would be less discriminable from the other groups than high-time pilots and non-pilots.

With two classes, a random classifier would be expected to produce 0.5 accuracy and 0.5 AUC values. Evaluation results for the classifier are listed in Figure 3.5. Both accuracy and area under the ROC curve appear significantly higher than both the random classifier and the string-editing approach. The string-editing approach does not appear significantly more accurate than a random classifier would be expected to be.

Results show the classifier's discriminative ability averaged over all stimuli. Multiple classifiers for each subject, over all the stimuli, would increase the confidence level of the individual classifiers by combining them. Therefore, a "majority vote" is used, where one vote is drawn from each stimulus. If more than half the votes indi-

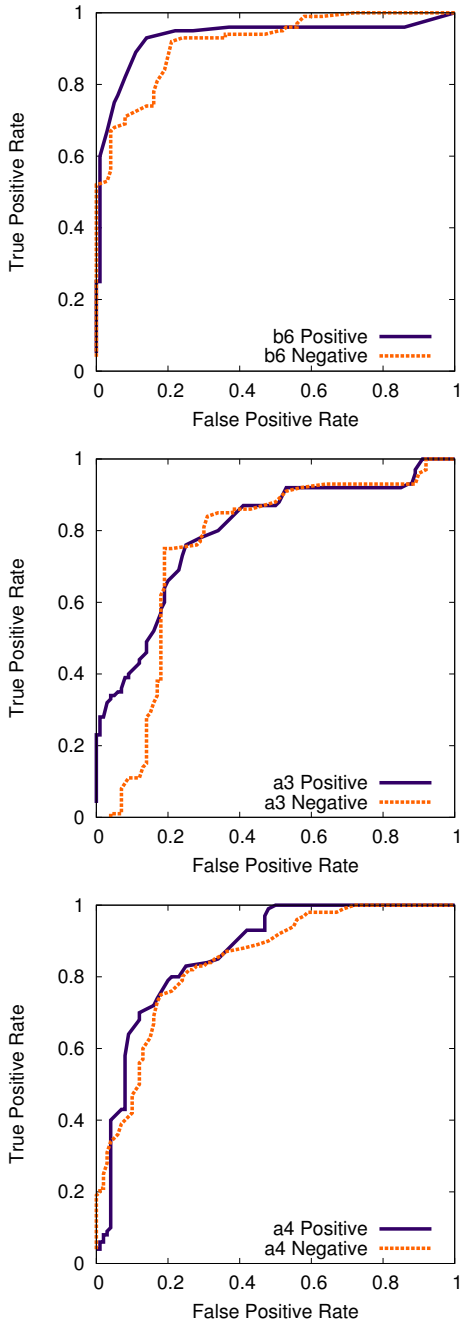


Figure 3.6: Examples of best, worst, and average ROC curves, respectively. Each example shows the ROC curve for the training data for both the positive and the negative classifiers, for that stimulus. Best case is stimulus 'b6'. Worst case is stimulus 'a3'. Average case is stimulus 'a4'.

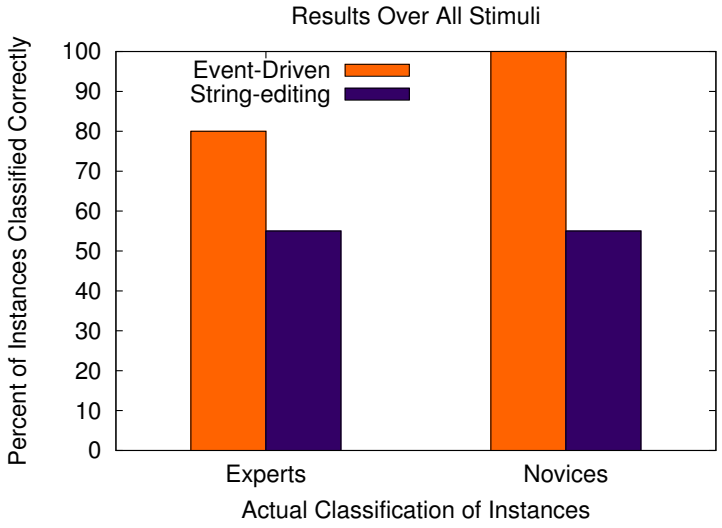


Figure 3.7: Results of cross-stimulus validation. Accuracy is determined by counting the number of experts/novices with expert ratio greater than 0.5 in the case of experts and less than or equal to 0.5 in the case of novices.

cate that a subject is expert, that subject is then classified as conclusively expert. Otherwise, a subject is classified as novice. Accuracies for this voting mechanism are listed in Figure 3.7. The new event-driven method is much more accurate than the string-editing approach. The cross-stimulus validation for string-editing is only slightly better than a random classifier would be expected to give. The event-driven method is 100% accurate for novices and 80% accurate for experts, yielding a total accuracy of 90% for the event-driven method. The string-editing method has only 55% total accuracy.

Originally, an approach similar to that of Torstling [2007] was used, attempting to use a generative model to predict group membership. That approach failed to achieve sufficient reliability, though it should be noted that this was a different problem. Torstling attempted to identify stimuli from eye movement data, while the present approach attempts to identify subject attributes, e.g., expertise. The new approach achieved greater accuracy than the generative model, though they

both were influenced by the machine learning paradigm.

Chapter 4

Video Stimulus Analysis and Classification

The scanpath classification algorithm takes as input two collections of scanpaths, in the form of lists of gaze points. Some example frames are presented in Figure 4.1, with heatmaps on the top obtained from tasked viewers and those on bottom from "free" viewers. Contrary to the expert/novice pilot study, there is an obvious difference between the two. With only a single stimulus for each classifier, there is no way to reinforce the classification with cross-stimulus validation. From the still images study, it was found that a single classifier would have just under 70% accuracy, on average, while accuracies for classification across many stimuli improved to 90% across both groups. It would be conservative to hypothesize that the accuracy of a classifier for the given video example would be expected to be somewhere between 70% and 90% accuracy, since the data appears more discriminable during cursory examination, but lacks the reinforcement of multiple stimuli (i.e., only a single video is viewed, in place of a number of images).

4.1 Classification Framework

The same classifier $C(X_t, X_r)$, as defined in Chapter 3, is employed for video stimulus analysis, except that classes are operationalized differently. Instead of experts and novices, two groups of viewers are given the same video stimulus, but with different instructions. In one instance, they are asked to view the video as they would watch a movie at home, e.g. this is the "free viewing" task. In the second instance, they are given specific instructions, pertaining to the video clip being viewed.

Figure 4.1: Heatmap visualization of gaze recorded over a video sequence (labeled sequence C) viewed by either "free viewing" (above) or following instruction to avoid faces (below), the latter artificially simulating reduced face gaze exhibited by autistic observers. Frames in both strips were rated highly as perceptually salient according to the level of attentional dispersion detected.

4.1.1 Eye Movement Analysis

Comparison of scanpaths captured over static media depends on the identification of fixations within the raw gaze point data stream. In this experiment, however, no fixation detection algorithm is applied for fear of removal of saccades made to sudden onset stimuli often present in dynamic media. Removal of these saccades could potentially remove reflexive eye movements to such events. More importantly, the third class of eye movements, smooth pursuits (or fixations on moving entities), are able to be completely ignored in still images, since they are impossible, whereas they may occur when viewing video stimuli. It is likely that a number of these smooth pursuits would be removed or corrupted by fixation filters. The algorithm thus operates on raw gaze points $\mathbf{x} = (x, y, t)$ recorded by the eye tracker. Although the fixation filter, as described previously, is hypothesized to be less useful for eye tracking data collected from video stimuli, its effect is reported in the evaluation to confirm this hypothesis.

4.1.2 Scanpath Comparison

The similarity measure is the backbone of the classifier. The framework outlined by Grindinger et al. [2010] was specifically motivated by the need for analysis of

eye tracking data collected over video stimuli. The motivation of the approach is to track the deviation of an individual's eye movements from the average of two or more sets of scanpaths that have already been classified as members of well-defined classes. The study that was previously used to demonstrate validity involved two such classes, namely experts and novices, and used the approach to show that class members could be reliably classified by their eye movements alone. The stimulus for that study was static. The present study applies the classification method to eye tracking data collected over dynamic stimuli. In the present situation, viewers are not initially sampled from expert or novice populations, but rather their scanpaths are grouped following task instruction. The study thus replicates Yarbus's [1967] classic work but provides an automatic means of distinguishing the resultant sets of scanpaths. Visualization is provided to show aggregate behavior of eye movements between groups defined by viewing behaviors. Classified aggregate scanpaths are then used to automatically select perceptually salient video frames where scanpath differences are evident (see Section 4.3).

4.2 Dynamic Heatmap Visualization

The behavior of the algorithm may be visualized, to some extent, by overlaying a heat map on the video frames while the clip is being played. This mechanism facilitates understanding of both the mechanics of the algorithm and the visual behavior of the viewers of the different classes. This visualization may aid in future applications, such as evaluation of whether a film audience's attention is focused on what the director intended. A director may be attempting to focus the audience's attention to certain aspects of a given scene, but it is uncertain whether the audience is drawn to that aspect or to some other unintentional object of interest. This visualization would provide a means of determining whether this was occurring and

what was causing the distraction.

Pixel intensity $I(x, y)$ at pixel coordinates (x, y) can be efficiently computed via $I(x, y) = \exp\left(-(x^2 + y^2)/(2\sigma^2)\right)$ by truncating the kernel beyond 2σ [Paris and Durand 2006] and setting σ to an arbitrary locus of influence, e.g., $1/6^{th}$ the screen dimensions, or error of the eye tracker (as done presently, see Figure 3.3). Note that the heatmap representation of scanpaths eliminates order information.

The above specification for a static heatmap generated at a single frame (the temporal parameter is implied), does not produce pleasing visualizations over dynamic media due to the potentially rapid appearance and disappearance of gaze points atop video frames. This is especially noticeable whenever the scene changes, e.g., due to camera movement. The situation is somewhat analogous to the sudden onset and termination of an audio signal—a more pleasing effect is produced by a fade-in and fade-out of the signal.

To achieve temporal visual decay, and generate dynamic heat maps [Daugherty 2009], the pixel intensity $I(x, y)$ can simply be accumulated via linear interpolation between video frames, e.g., $I(x, y, t) = (1 - h)I(x, y, t - 1) + hI(x, y, t)$. In practice, a value of $h = 0.4$ appears to work well. After the accumulation buffer is calculated for a given frame, it is mapped to the R,G,B color space. Suitable color threshold values were picked so that the accumulated intensity I produced a color gradient "increasing" from green, to yellow, to red.

4.3 Perceptual Saliency of Video Frames

Over the span of a long video sequence, some frames are likely to be more perceptually salient than others. The perceptual saliency of an individual frame may be estimated by measuring the inter- and intra-class dispersion of classified gaze points over the frame. For instance, perceptual saliency could describe to what extent or what percentage of the audience is being distracted. Dispersion of two classes is measured by calculating the standard deviation of the distance of all gaze points of one class to all gaze points of the other class. This measure is similar to how Daugherty [2009] displayed gaze point grouping during dynamic heatmap visualization. With the Euclidean distance between two gaze points at frame t, $r_{i,j}(t) = |\mathbf{x}_i - \mathbf{x}_j|_t$, with $\mathbf{x}_i \in s_i$ and $\mathbf{x}_j \in s_j$, and mean inter-class distance $\omega_{i,j}(t) = 1/(mn)\sum_i^m \sum_j^n r_{i,j}(t) \ \forall \ i,j, \quad i \neq j$, inter-class dispersion of two scanpaths is estimated by the standard deviation,

$$\mathrm{SD}(s_i, s_j, t) = \sqrt{\frac{1}{mn-1}\sum_{i}^{m}\sum_{j}^{n}\left(r_{i,j}(t) - \omega_{i,j}(t)\right)^2} \ \forall \ i,j, \quad i \neq j.$$

Intra-class dispersion is calculated similarly with $s_i = s_j$.

Perceptual saliency, as defined here, depends on the dispersion of visual attention. Intuitively, one may expect small dispersion to be a better indicator of a frame's perceptual saliency than large dispersion. This could indicate, for example, a focal object of interest in the frame, e.g., as may be evoked by a close shot of a face. On the other hand, the perceptual saliency of a frame may depend on the visual task being performed, as well as the class of viewers (e.g., expert/novice). For instance, were a group of viewers instructed to look away from a specific type of object in a scene, they would likely choose a variety of different objects to fixate. The resulting large dispersion may indicate a highly perceptually salient frame for these viewers, e.g., autistic viewers [Leigh and Zee 1991] (this point is further addressed

in experimental procedures below).

4.4 Empirical Validation

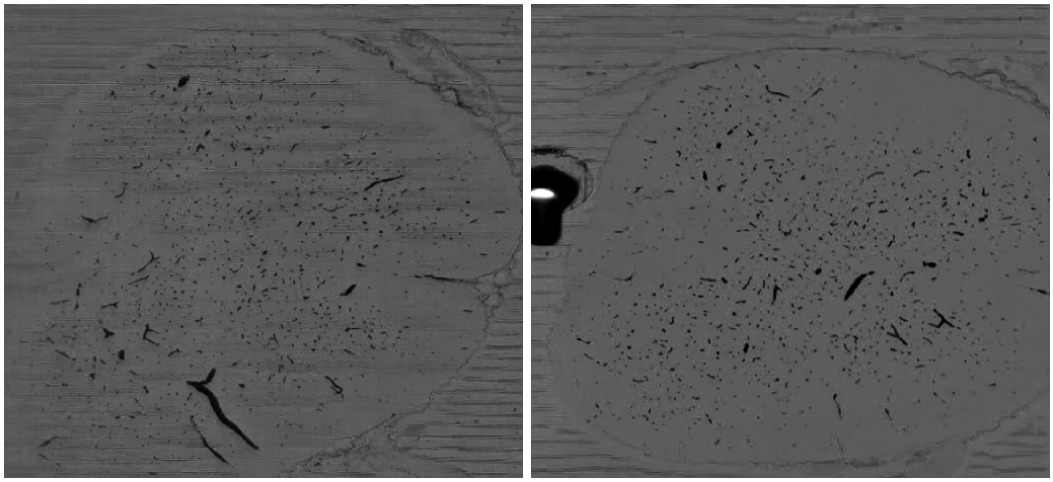
Eye tracking data was recorded for repeated viewings of three video sequences. Participants' first viewing was natural (amounting to "free viewing"), while they were given a task to perform during the second viewing. It was hypothesized that the data could be reliably classified by the described approach given differing instruction to viewers (see below).

Stimulus. Stimuli consisted of three video sequences, named A, B, and C, and shown in Latin square counterbalanced order, with approximately each third of the viewers seeing the sequences in order $\{A, B, C\}$, $\{B, C, A\}$, or $\{C, A, B\}$. Sequence A contained a misplaced modern pair of sneakers in an 18^{th} century setting, while a modern popular song played in the background. Sequence C was from the same feature film, with scenes containing a large number of human faces. Sequence B was a sequence of CT-like scans of the mouse vasculature in the spinal cord. Select frames from all clips are shown in Figure 4.2.

Participants. Twenty-seven college students volunteered in the study (seven male, twenty female). Ages of the participants ranged from 18 to 21 years old.

Procedures. Participants sat in front of the eye tracker at about 60 cm distance. Calibration required visually following nine targets. Following calibration, subjects were asked to watch the first of two viewings of each of the three sequences naturally. They then received viewing instructions prior to the second viewing of the same sequence.

(a) Sequence A, chosen for its misplaced pair of modern sneakers.



(b) Sequence B, chosen for its expected unfamiliarity.

(c) Sequence C, chosen for its large number of prominent faces.

Figure 4.2: Frames from stimulus sequences. Sequences A and C were excerpts from Sofia Coppola's *Marie Antoinette* © 2006, Columbia Pictures and Sony International, obtained with permission for research purposes by the Universitat Autònoma de Barcelona. Sequence B shows the mouse vasculature in the spinal cord at 0.6×0.6×2 μm resolution with blood vessels stained black, as obtained by a knife-edge microscope (courtesy of Texas A&M).

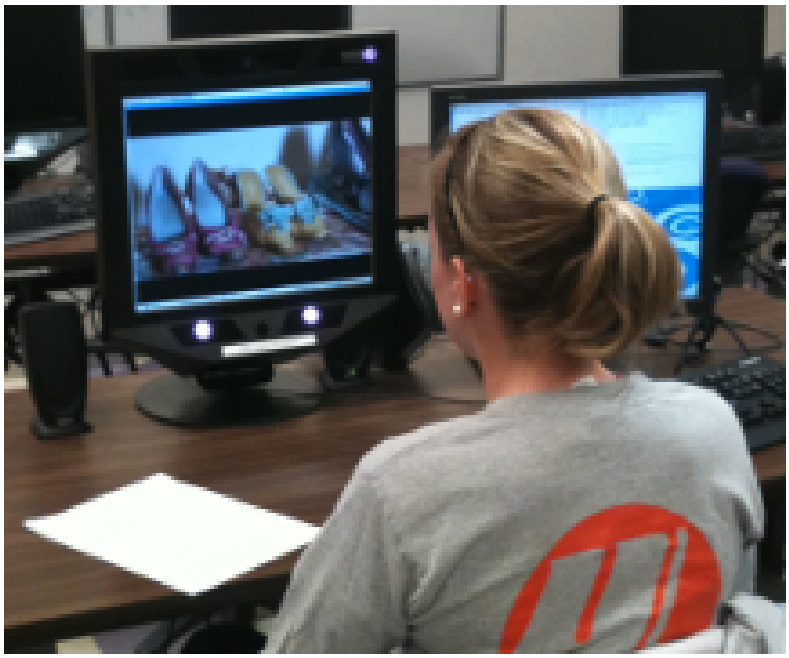


Figure 4.3: Tobii eye tracking hardware setup.

For sequence A, participants were asked to look for anything unusual. It is hypothesized that viewers would notice the sneakers given this instruction. For sequence B, the contents of the clip were revealed only after the first viewing, when they were asked to visually follow the path of blood vessels as they viewed the spinal cord sections. It is hypothesized that in this case viewers would avoid the aberrant artifacts at the sides of the frames and focus on the vascular stains.

For sequence C, participants were asked to avoid looking at faces. It is hypothesized that this instruction could artificially simulate autism, since autistic individuals have been shown to exhibit reduced face gaze [Leigh and Zee 1991].

Apparatus. Eye movements were captures by a Tobii ET-1750 eye tracker (see Figure 4.3), a 17 inch (1280×1024) flat panel with built-in eye tracking optics. The eye tracker is binocular, sampling at 50 Hz with 0.5° accuracy.

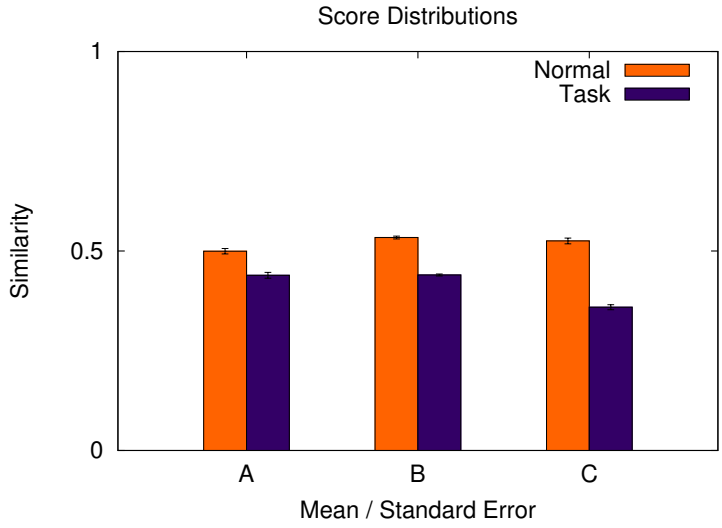


Figure 4.4: Mean similarity score and standard error for video stimulus scanpaths.

4.4.1 Results

Eye movement data from "natural" (free) viewings act as one class (the "negative" class in this study), while data from tasked viewings serves as another class (the "positive" class). The accuracy and reliability of the classification approach is evaluated with this specification for each of the three sequences.

Results for viewing of stimulus A may be found in Figure 4.5. Stimulus A is one to which the described classifier does not respond positively. Accuracy and AUC are not even above the bare minimum for a classifier of 0.7. Even though there appears to be significant difference between the distributions of positive and negative scores for stimulus A in Figure 4.4, the accuracy and AUC are both close to random, which relates that the data for this stimulus could not be classified reliably. This seems counter-intuitive, since, if there is significant difference in the distribution of scores, it would stand to reason that a split could be made between the two which would yield some discriminability. These are not typical normal distributions for this stimulus, however. In the case of the positive scores, a number of the instances

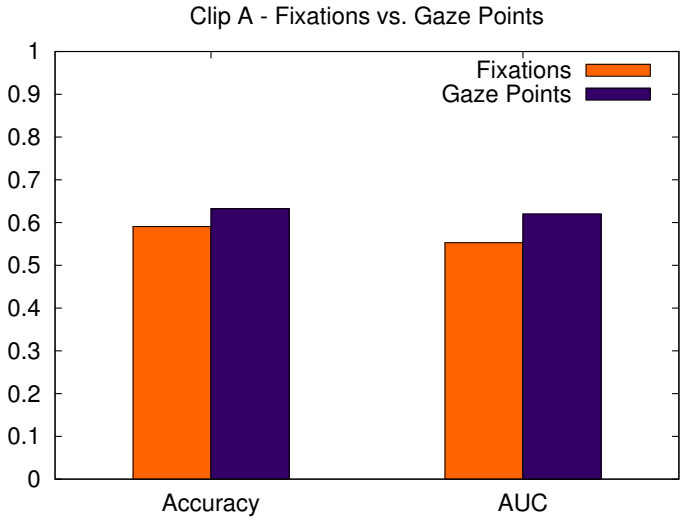


Figure 4.5: Results of experimental analysis for clip A. Columns indicate classification accuracy (detecting tasked or natural viewing) and AUC.

actually score lower than most of the average negative scores. Most of the negative score distribution overlaps closely with the positive instances, while the low-scoring positive instances give the appearance of significant difference. Five video frames during normal and tasked viewings are shown with heatmap visualization in Figure 4.9(a). These frames were chosen for their greater disparity between classes than other frames, yet it is similarity between the two classes that is more apparent. Accuracy and AUC value for fixations is slightly higher than for gaze points in this stimulus, though neither appear significantly better than random.

Results for viewing of stimulus B may be found in Figure 4.6. Results for stimulus B indicate much better classifiability than stimulus A. Both accuracy and AUC are higher than 0.7, the bare minimum for a classifier. The AUC value is actually closer to 0.8. Video frames for normal and tasked viewings are shown in Figure 4.9(b). The differences between the classes are much more apparent for this stimulus than for stimulus A, as reflected in the classification results. The algorithm that uses fixations has lower accuracy and AUC than the algorithm using gaze points.

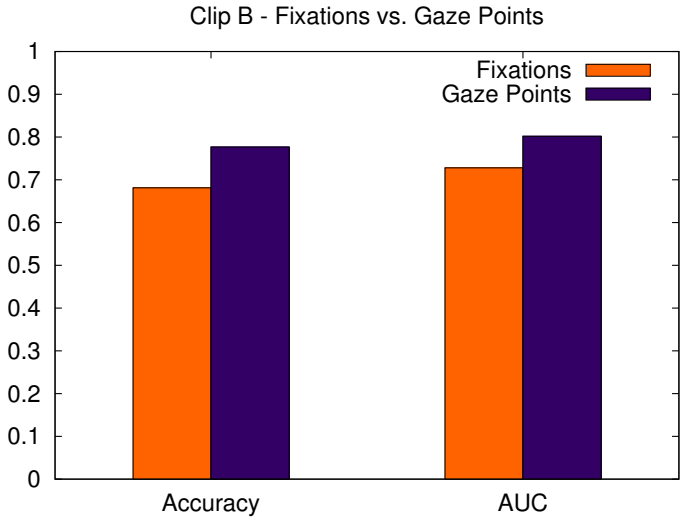


Figure 4.6: Results of experimental analysis for clip B. Columns indicate classification accuracy (detecting tasked or natural viewing) and AUC.

Results for viewings of stimulus C may be found in Figure 4.7. For stimulus C, results are quite similar to stimulus B, but accuracy and AUC are slightly lower. They are still higher than the results for stimulus A, however, and above the bare minimum classifier accuracy and AUC. Although results for stimulus C were slightly worse than for stimulus B, the task given to the participants required far more distinct eye movement behavior for specific portions of the clip. A large portion of stimulus C, however, did not require modified eye movement behavior from natural viewing. A segment of the clip was selected that was expected to produce highly discriminable behavior between the two groups. The specific portion selected was a fifteen second window, including the first two frames displayed in Figure 4.1. The results for this short excerpt may be found in Figure 4.8. The accuracy and AUC value for this excerpt are much higher for this short excerpt than for the entire clip. Both the accuracies and AUC values for the entire clip and the excerpt show that use of fixations yields lower accuracy and AUC than gaze points, for this stimulus.

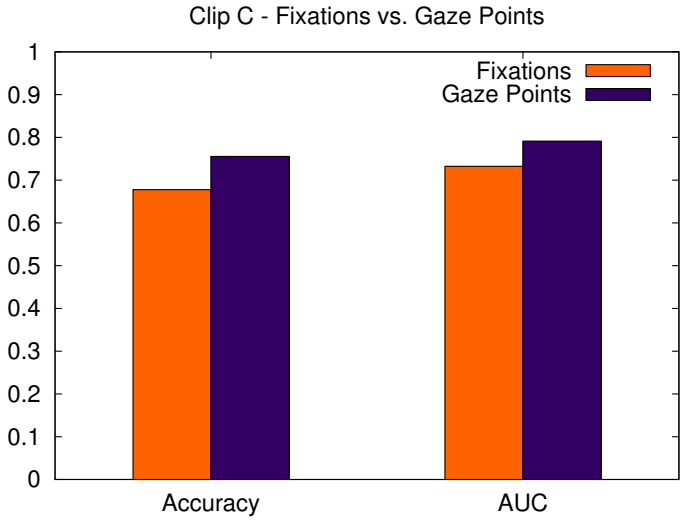


Figure 4.7: Accuracy and AUC results of execution with fixations or gaze points for stimulus C.

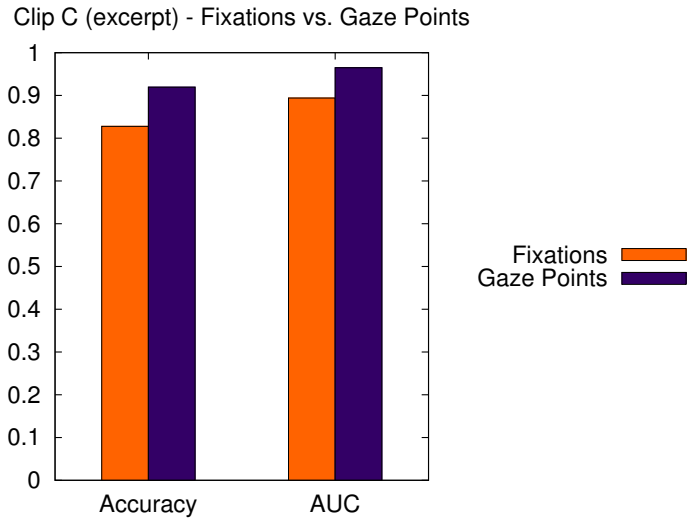
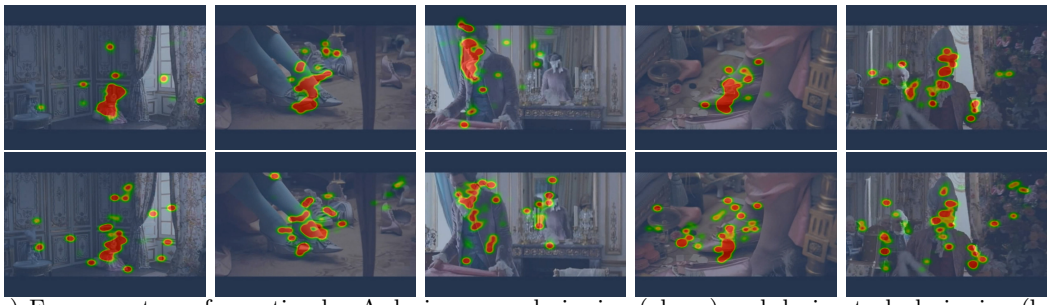
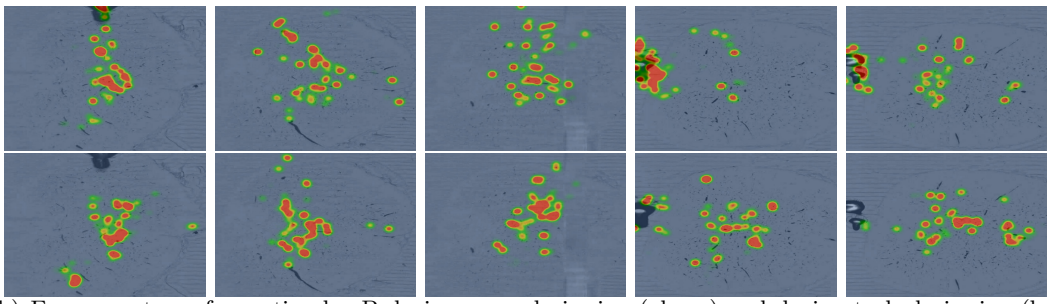


Figure 4.8: Accuracy and AUC results of execution with fixations or gaze points for an excerpt of stimulus C.



(a) Frame captures from stimulus A during normal viewing (above) and during tasked viewing (below).



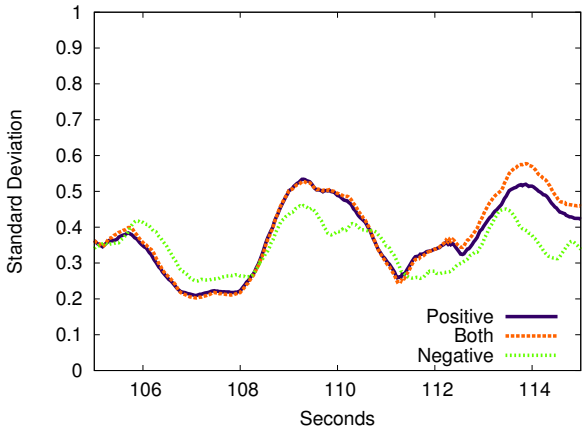
(b) Frame captures from stimulus B during normal viewing (above) and during tasked viewing (below).

Figure 4.9: Dynamic heatmap visualization of gaze over video sequences.

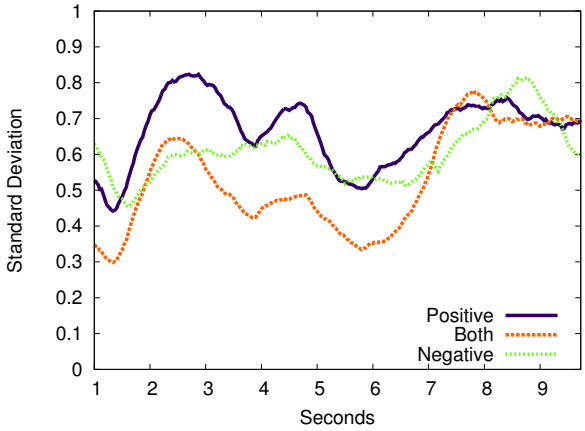
4.4.2 Perceptual Saliency

The video frames selected for visualization in Figures 4.1 and 4.9 were rated particularly well by the perceptual saliency estimation outlined above. This notion of perceptual saliency refers to the degree of dispersion of gaze points aggregated atop video frames. Small dispersion, or tight grouping, tends to identify close shots, particularly ones where a prominent face is present. It is likely that this metric could be used to automatically select video frames possessing similar artistic direction. Alternatively, the approach could identify problems in artistic direction, where the dispersion of attention is unintentionally large (loose grouping of gaze points).

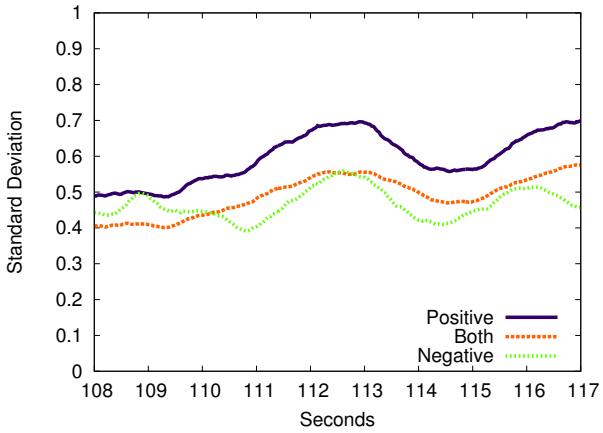
Saliency graphs for short excerpts of the video clips are shown in Figure 4.10, roughly centered on the frame shown at center in the strips rendered in Figures 4.1 and 4.9. Stimulus A curves appear to follow each other closely and this is typical of



(a) Sequence A saliency graph.



(b) Sequence B saliency graph.



(c) Sequence C saliency graph.

Figure 4.10: Saliency graph curves are mean filtered over a two-second window and display about a 10 second excerpt of the entire clip.

the entire sequence.

Curves for stimulus C tend to move independently of each other, though there appears to be some correlation. Standard deviation of the positive class tends to be higher, in general, than the negative class, which is to be expected when participants scatter their focus away from a specific object. There are two means of determining which frames are more salient: pick the peaks of the cross-class standard deviation (labeled "Both" in the figure) or pick the frames for which the difference between the positive and negative curves is largest. For this particular stimulus, frames which have either of these properties likely contain faces.

The saliency curves of the positive and negative class from stimulus B tend to mirror each other until the end of the video, where a large number of the free viewing subjects and only a handful of the tasked subjects were attracted to the large black spot in the video.

4.4.3 Embedded Figures Test

Subjects were given a short embedded figures test at the end of the experiment. They were tasked with finding specific figures (e.g., a circle, triangle, square, etc.) in an image with many different figures inside it, but only one of the figure they were assigned to find. This portion of the experiment did not involve eye tracking. The intention was to provide a means of filtering out any data from individuals that might be particularly unusual. The average length of time to complete a single trial would then be an indicator as to whether the individual's data should be included or not.

In practice, the time to completion was very close, with an average of $2,695$ milliseconds. The standard deviation was 595 milliseconds, indicating that time

to completion did not vary much across subjects. The longest average time to completion was $4,255$, and the shortest was $1,700$. No subjects were filtered out by the results of this test.

4.4.4 Discussion

Classification of sequences B and C are similar to expectation, while results for stimulus A are mixed, with accuracy similar to what would be expected from a random classifer and only slightly higher AUC. The natural and tasked viewings appear similar to each other. There may be several explanations. Stimulus A was the only one with an audio track, which may have been distracting. The task given to participants for the second viewing of stimulus A was too general, with subjects merely being told to "stay alert". Only three viewers reported seeing the sneakers in the sequence. One of the viewers reported that some of the shoes looked modern, but when asked which ones, they cited instances other than the sneakers. Other objects of interest mentioned were birds in the hairpiece and dogs with jewelery.

Participants were asked to guess what the second clip was depicting after viewing it the first time. The most common response was "something under a microscope". Some participants noted a resemblance to tadpoles, bacteria, or cells. After explaining the true nature of the sequence (a type of CT scan of the mouse spinal column), twenty viewers reported being able to understand the clip better. Almost every viewer deemed the large black spots on the top and left sides of the clip eye-catching, and some participants had trouble avoiding them for the tasked portion of the trial.

Sequence C was intended to demonstrate the classifier's ability to distinguish between eye movement behaviors of normal viewers and those suffering from neuro-developmental disorders such as autism. The particular task of avoiding faces is

characteristic of autistic individuals. Although none of the participants indicated that they had autism, they were essentially being asked to simulate autistic behavior. The high accuracy of the classifier for this clip appears to indicate that it may be a reliable aid to this type of diagnosis. After normal viewing of this sequence, viewers noted that people looked upset or angry, that no one was smiling but Marie Antoinette, and that she was being stared at. Of course, when they were instructed to avoid faces, none of these observations occurred. They tended to look at the sky, the floor, and the architecture. Some mentioned that it was difficult for them to avoid faces, and that they had to concentrate to do so.

It is expected that short clips tend to produce more discriminative scanpaths than longer clips. Stimulus C certainly appears to be more discriminable, at first glance, than stimulus B does. The actual results show very similar discriminability, though, which is most likely due to the length of stimulus C being much longer than stimulus B. There are several possible causes for this. The most likely cause is extended periods of both very similar and very different viewing behaviors between classes. Both classes tended to look at the horses and birds. Including this conformant data in the analysis would lower the discriminability over time. Another cause could be deteriorating calibration, which is a problem in all eye tracking applications, since participants tend to fidget and change position while watching the clips. Another possible factor could be the tendency to settle into natural viewing patterns during a visual task of extended length, i.e., forgetting the task instructions.

Fixations appear to consistently produce lower accuracies and AUC values for discriminable video stimuli. This is in line with the hypothesis that fixation filtering affects data collected during video stimuli negatively, artificially interrupting smooth pursuits and removing reactions to sudden onset stimuli. The fixation filter may still be worthwhile, but its functionality will need to be adapted to dynamic

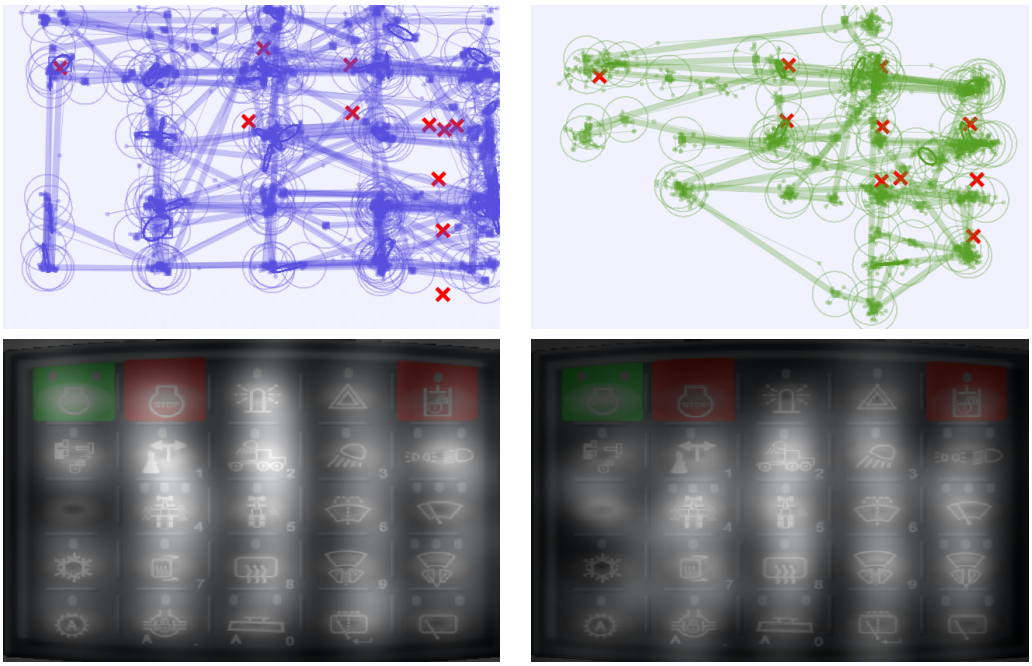
stimuli. Instead of discarding smooth pursuits or breaking them apart into multiple fixations, they may be classified and analyzed independently, using a mechanism such as the Kalman filter, described in Grindinger [2006].

Chapter 5

Interactive Stimulus Analysis and Classification

The study used for evaluation in this chapter consists of a set of visual tasks that were given to two classes of subjects: those of western descent and those of eastern descent. A visualization of exemplar scanpaths used in this study is presented in Figure 5.1, displayed on the top-left with eastern scanpaths and on the top-right with western scanpaths. These scanpaths appear very different, but the aggregate heatmap shows less difference. The task given to the subjects was to select a specific sequence of buttons in the interface. Subjects took very different lengths of time to locate and select each of the buttons required. A temporal adaptation prealignment procedure will now be described that will attempt to provide the functionality necessary to analyze this type of data, namely a step-wise method of normalization, allowing the use of the new classifier algorithm. The definition of "event" for this study is a fixation on a particular button. In this study, an event cannot be a fixed interval of time, since each step of the task takes a variable amount of time to complete, which differs for each subject. This particular definition of an event allows them to be aligned in such a way as to allow the use of the similarity measure originally described on commonly-occurring events. The performance of this modification will be evaluated against the original string-editing algorithm.

Presently, a variant of the position-variance algorithm is used with a spatial deviation threshold of 30 pixels is used and the number of samples set to 5 (implying a temporal threshold of 100 ms at a 50 Hz sampling rate). The fixation analysis code is freely available on the web.



(a) Easterners' (left) and Westerners' (right) eye movements

Figure 5.1: Representative scanpaths and aggregate heatmaps (exp. 1).

5.1 Temporal Adaptation

The scanpath classification approach was intended to work well for static images and was formulated to be particularly effective for video tasks, but it is not, in its current state, appropriate for variable-duration interactive tasks. For instance, suppose a sequence of numbers are distributed onto an image. The task is to look at each number in order. Some individuals may take a fraction of the time that it took others, but the order of the fixations is very similar. One would expect the sequence of fixations to be very similar, while the actual timestamps are quite different.

In this case, it is possible to manipulate the duration of fixations to map one scanpath onto another of different length by using the Levenshtein alignment of the two scanpaths as a reference. Here is an example of two scanpaths of different lengths:

```
0  1  2  3  4  0  5

0 11 12  6  1 13  9  9 14  0 15  1 16  8 17
```

This is the Levenshtein alignment of those two scanpaths:

```
0  -  -  -  1  -  2  3  4  0  -  -  -  -  5

0 11 12  6  1 13  9  9 14  0 15  1 16  8 17
```

The numbers in each of the examples refer to labels that have been arbitrarily chosen for each discrete fixation. Each label denotes a region of interest in the stimulus. Labels repeated in both scanpaths indicate commonality between the two scanpaths. A '-' character indicates a gap in the paths. These blank pieces of the alignment will not be used in the scanpath similarity measure, since they effectively stretch a single point in time to cover an extended portion of the larger alignment. Scanpaths which do not have a blank piece at a particular point will still be used in the similarity calculation for that point, though, even if other scanpaths do have blank pieces.

The string-editing alignment may be used as a preprocessing procedure (hereafter referred to as prealignment) for group-wise scanpath similarity. For instance, a scanpath that is being compared to a group could first be compared to each member of that group. A Levenshtein alignment with the best-matching scanpath within the group would then be expected to serve well as a template with which to adapt the new scanpath to the group. The timestamps of the fixations in the new scanpath's alignment and those remaining in the group would then be adjusted to coincide with the timestamps from the best-matching scanpath within the group. In this way, the scanpath being analyzed and the group it is being compared to are normalized, such that they all have the same internal structure (i.e., they have the same number of "pseudo-fixations"). The similarity and classification algorithm would then be executed normally. A simple visualization of two scanpaths, before and after alignment, may be seen in Figure 5.2.

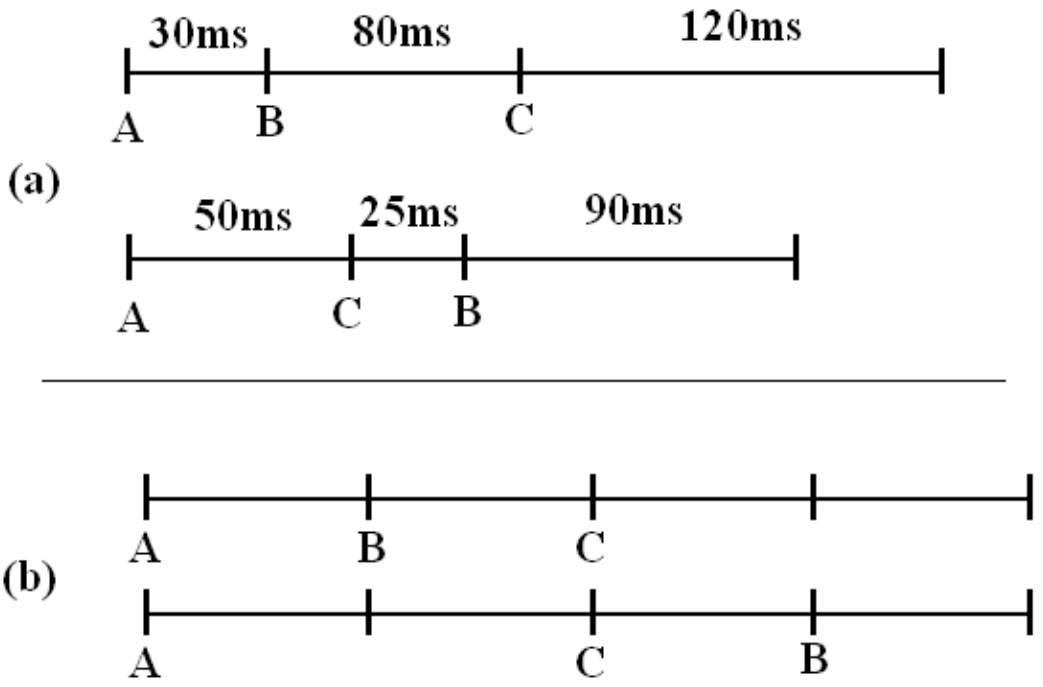


Figure 5.2: Simple pairwise scanpath alignment. Original scanpaths in (a) and aligned scanpaths in (b).

To accomplish this, a multiple sequence alignment algorithm must be utilized. The particular algorithm used for this purpose was Clustal, described by Higgins [1988], initially intended to align sequences of nucleotides in DNA and RNA or amino acids in proteins. In the case of eye tracking classification, a sequence of ROI labels is used. Scanpaths may be recorded in multiple formats. It is common to label various areas on a stimulus, such as buttons on interfaces or clearly-defined objects in images.

The Clustal algorithm consists of two steps. The first step involves constructing a phylogenetic tree from the pairwise similarity of all sequences that are being aligned. This tree is constructed by joining pairs of sequences in order of similarity, from highest to lowest. Pairs of sequences may also be joined with other groups of sequences. In the case of joining two groups of sequences together, the group-wise similarity of the two groups is used to determine when they are to be merged. Group-wise similarity, in respect to the Clustal algorithm, is obtained similarly to pair-wise similarity. Instead of using a fixed cost of substitution, such as in string-editing distance, however, the average cost of string operations is used. For instance, suppose one group of sequences has the labels $ABBB$ at some position. If this group of sequences were being compared to another group of sequences at a particular position, having the labels $BBCD$, the substitution cost of aligning the two groups of sequences at that position would be 0.5, since half of the labels are shared between the two groups at that particular alignment position. Aside from this small difference, the similarity matrix may be visualized similarly as in Figure 2.1.

The second step in the Clustal algorithm is the construction of the multiple sequence alignment. Groups of sequences are aligned in much the same way as pairs of sequences are aligned. Instead of inserting blank positions in single sequences, though, blank positions are inserted in all the sequences in the group if the align-

ment algorithm needs to do so.

Given a single group of scanpaths that have all been aligned with each other, it is then possible to utilize the event-driven scanpath comparison method described previously. The prealignment then serves as a list of events, from which similarity is derived. It may be argued that this approach could be improved by subdividing the events even further. This subdivision would necessarily require the divisions to be variable lengths of time, depending on the fixation being subdivided. A particular column in an alignment consists of a list of fixations, each of which has a specific duration attached to it. These individual durations would have to be subdivided into an equal number of divisions, in order to apply the similarity measure to them, which would cause the intervals of time between each subdivision to vary drastically between scanpaths. For instance, some fixation could have length 30 milliseconds, while another has length 50 milliseconds, such as the two 'A' fixations in Figure 5.2. Nevertheless, the results of such a subdivision will be explored in the empirical validation.

5.2 Empirical Validation

With the temporal adaptation (prealignment) mechanism, it is now possible to attempt to compare and classify groups of scanpaths with the previously-described classification mechanism. A study was previously conducted to gauge difference in performance on an interactive visual search task between different cultures, which serves as validation for the temporal adaptation mechanism. Similar to the video study described in the last section, this study does not measure expert/novice similarity. Instead, it is used to distinguish differences in visual search between two cultures.

Two experiments were conducted to gauge cultural differences during interaction. The first study required participants to complete four tasks: two visual search and icon localization tasks and two menu navigation tasks. So that scanpath visualization was not cluttered by eye movement during menu navigation, gaze point recording was turned off whenever menus were visible. A second follow-on experiment was conducted to compare cultural differences during a visual search task with symbolic icons replaced by randomly generated numerals. The second study thus serves as a type of baseline for comparison of scanpath similarity metrics.

Apparatus. A Tobii ET-1750 video-based corneal reflection (binocular) eye tracker was used for real-time gaze measurement (and recording). The eye tracker operates at a sampling rate of 50 Hz with an accuracy of about 0.3° over a ±20° horizontal and vertical range [Tobii Technology AB 2003]. The eye tracker's 17″ LCD monitor was set to 1280 × 1024 resolution and the stimulus display was maximized to cover the entire screen (save for its title bar at the top). The eye tracking server ran on a dual 2.0 GHz AMD Opteron 246 PC (2 G RAM) running Windows XP. The client display application ran on a 2.2 GHz AMD Opteron 148 Sun Ultra 20 running the CentOS operating system. The client/server PCs were connected via 1 Gb Ethernet (switched on the same subnet). Participants sat at a viewing distance of about 50 cm from the monitor, the camera's focal length. The same eye tracking apparatus was used for both experiments.

Experiment 1. The first experiment concerned navigation and selection of iconic and menu elements in the simulated interface shown in Figure 5.3(a).

Subjects. 20 college students participated in the first study. All participants had at least some college experience. Ten participants were of South Asian (Indian)

(a) assigned icon button tasks (b) relabeled numerical buttons

Figure 5.3: Iconographic and random numerical button tasks.

descent (7 M, 3 F, mean age 23.8) composing what will be referred to as the Eastern group in the remainder of the paper. The other ten US participants composed the Western group (6 M, 4 F, mean age 24.2). All of the Eastern participants spoke English as a secondary language (primary language Hindi, Malayalam, Marathi, or Tamil), having moved to the US within the last five years. All of the Western participants spoke English as a primary language. All 20 participants had at least four years of college education and all but one possessed a college degree. Participants were compensated $8 for participation in the first experiment.

Procedure. Participants were greeted and instructed to sit in front of the eye tracker (see Figure 5.4). A brief introduction was given, signature and responses were then obtained from an informed consent form and a brief demographic questionnaire. Next, participants were told they would be completing basic computer-based tasks using the mouse. Before proceeding, the eye tracker was calibrated to each participant. To do so, participants were told that they would see two gray circles on the screen representing the location of their eyes. If these gray circles were not centered, participants should adjust their position accordingly, e.g., if the circles were too high, the chair should be lowered. In addition to the gray circles, a

Figure 5.4: Participants at eye tracker.

roving yellow dot was shown moving to each of five programmed screen locations. Participants were asked to visually follow this dot. Following calibration, the experimenter brought up the button panel and menu interface. A short familiarization task was completed from which the data was not recorded. The experimenter then read through the four recorded tasks, with order counterbalanced by a partial Latin square design.

Participants were asked to press the green Start Engine button located at the top left of the interface to begin each of the four tasks (button A seen in Figure 5.3(a)). The first task (or first recording, R01), required participants to turn on the Air Conditioning (button I), Autoshift (button K), and Lever Steering (button B). Next, participants were asked to press the reset button before continuing on to task two (or recording two, R02). Again, participants were asked to start the engine, and then were told to find the Diagnostics menu and check the Hydraulic Oil Temperature. The reset button was then pushed again before moving on to task three (R03). Task three started with the Start Engine button, and then location of the Machine Settings where participants were asked to change the Reversing Fan

Cycle to 30 minutes and to turn on the Manual Fan Reversal. Once again, the reset button was pressed before proceeding to the fourth and final task (R04). Following the Start Engine button press, participants were asked to turn on all buttons associated with exterior work lights (buttons C, D, E, F, and G) followed by turning on all windshield wiper buttons (buttons H, J, L). To complete the task sequence, participants were asked to press the finish button at the top of the screen.

Experiment 2. The second experiment only required visual search of icons relabeled with random two-digit numerals (see Figure 5.3(b)).

Subjects. 20 college students participated in the second study. All participants had at least some college experience. Ten participants were of South Asian descent composing the Eastern group (8 M, 2 F, mean age 23.7), with seven returning from the first experiment. The ten remaining US participants composed the Western group (6 M, 4 F, mean age 24.5), with seven returning from the first experiment. All of the Eastern participants spoke English as a secondary language (primary language Malayalam, Marathi, Tamil, or Telugu), having moved to the US within the last five years. All of the Western participants spoke English as a primary language. All 20 participants had at least four years of college education and all but one possessed a college degree. Participants were paid $5 for participation in the second experiment.

Procedure. The procedure in the second study was similar to that in the first, with the exception that all four of the trials involved visual search of icons. Only the first and fourth runs were used in scanpath comparison, matching the analysis performed in the first experiment. No menu search was involved in the second study. Each of the four tasks (counterbalanced via a Latin square) required search for and selection (clicking) of icons replaced by two-digit numerals. The

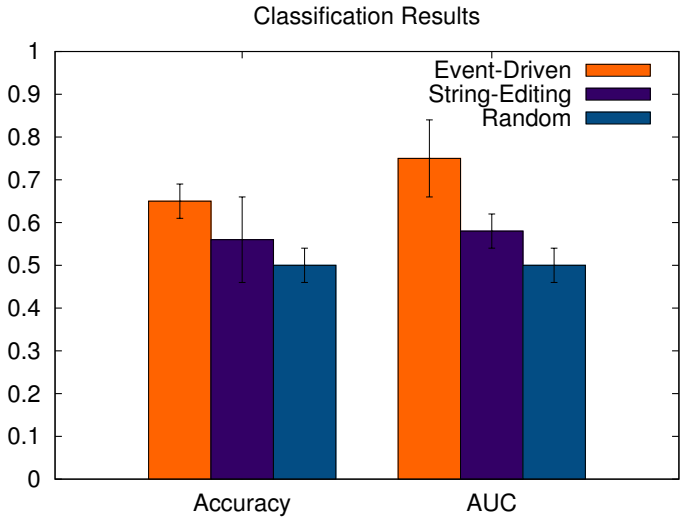


Figure 5.5: Mean accuracy and AUC for event-driven, string-editing, and random classifiers for symbol search tasks.

first task (or first recording, R01), required participants to find the following sequence: $\{13, 20, 15, 28, 16, 11, 26\}$. The second recording (R02) used the sequence $\{25, 17, 31, 24, 28, 13, 16\}$. The third and fourth recordings used the following sequences: $\{26, 14, 29, 15, 31, 27, 19\}$ and $\{20, 10, 13, 32, 24, 19, 15\}$, respectively. Each of the tasks required pressing the Start Engine button as in the first study.

5.3 Results

The results, both with and without the subdivision concept (described in section 5.1), were evaluated. The results with subdivision actually deteriorated the accuracy to become indistinguishable from random. Thus, the non-subdivided results are reported in this section. The average accuracy and AUC for the symbol selection experiment are presented in Figure 5.5. The accuracy and AUC for the event-driven similarity measure appear to be significantly greater than random. The accuracy does not appear to be significantly greater than the string-editing algorithm, though.

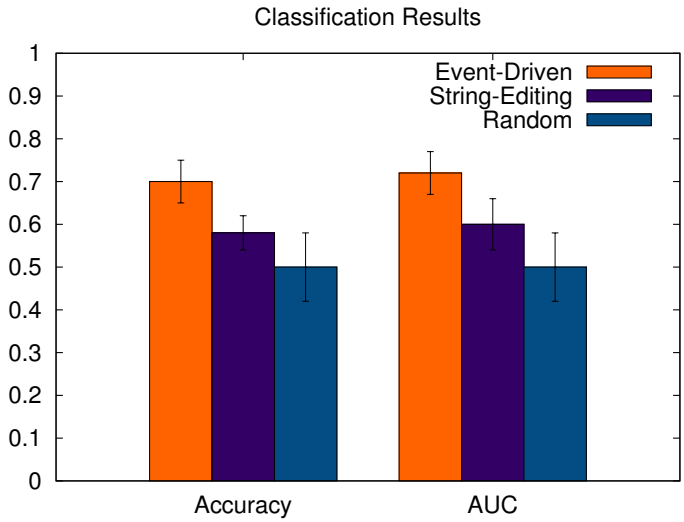


Figure 5.6: Mean accuracy and AUC for event-driven, string-editing, and random classifiers for number search tasks.

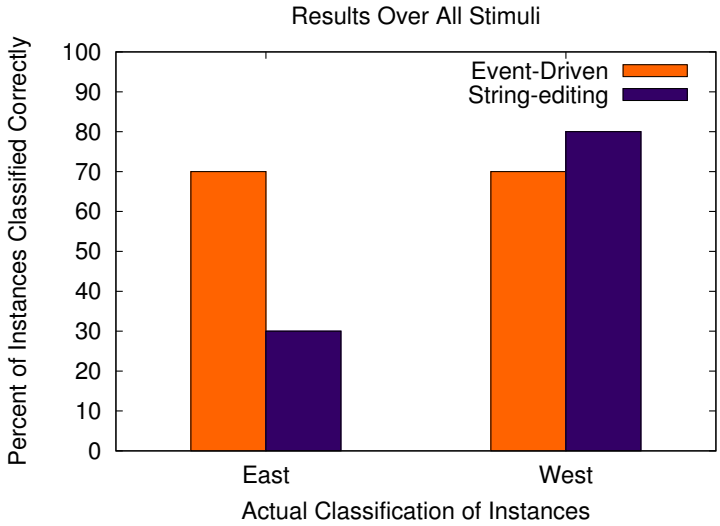


Figure 5.7: Cross-stimulus classification results for event-driven and string-editing classifiers.

Neither the accuracy nor the AUC appear to be better than random for string-editing.

Figure 5.6 reports the accuracy and AUC for the number selection experiment. In this case, the accuracy and AUC for the event-driven similarity measure both appear significantly higher than both string-editing and random. Similar to the symbol experiment, the string-editing measure does not appear significantly different from random.

Cross-stimulus classification reinforcement is possible in this experiment, similar to the expert/novice study. Results for the reinforcement are shown in Figure 5.7. Reinforced accuracy are not as improved as in the expert/novice case, with accuracy of 70% for both eastern and western subjects. The string-editing results are 30% for eastern subjects and 80% for western subjects, averaging to 55% total accuracy, unimproved from random.

5.4 Discussion

The data set for this study was quite small. Ten data points for each group would make it difficult to establish significance for many types of experiments. The fact that significance was achieved appears to imply that, for this problem, cultures performed the visual search task very similarly within cultures and very differently between cultures. The fact that a numeric search task yielded even greater significance than a symbolic search task is especially unexpected.

Chapter 6

Discussion

6.1 General Discussion

The event-driven similarity algorithm is very customizable. There are multiple details that may be modified if different behavior is desired. For instance, the Gaussian frame similarity method may have different weighting schemes. The current weighting scheme merely averages the similarities of fixations. It is also possible to choose the maximum similarity, instead. The implementor may also choose to cluster or not to cluster the fixations before evaluating the Gaussian frame similarity. The advantage to using clustering would be reduction of "noise" in the frame, and each cluster would be more meaningful than a simple fixation or gaze point. This is especially important in studies with high volumes of data. The advantage of not using clusters is that the mean shift algorithm tends to be the bottleneck, so removing the clustering algorithm would speed up the algorithm, at the risk of uninformative information affecting the result.

As described in the section on interactive stimuli, the user of the algorithm may not desire for every frame to be evaluated. Using the perceptual saliency approach or some other weighting scheme, informative frames may be automatically retrieved from the stimulus, leading to a faster evaluation and possibly more discriminative classification. There is no substitute, however, for designing the experiment in such a way that a maximal amount of informative data is retrieved, while a minimal amount of useless data is generated. In retrospect, the video clips in the second study were longer than optimal. Video clips of less than a minute length would likely have produced more discriminable data, since portions of the clip that were

expected to be highly discriminable could be selected.

The running time of the algorithm is quite high for these extensive evaluation and validation studies. For incremental classification of a single, unknown scanpath, the algorithm takes $O(mn * p(m))$, where m is the number of known scanpaths, n is the number of frames to be analyzed, and $p(m)$ is the running time of the mean shift algorithm on the collection of fixations, which is less than or equal to m for each frame. The runtime of $p(m)$ varies, depending on how many iterations are needed for convergence, which Santella and DeCarlo [2004] observed to be generally between 5 and 10 iterations for eye tracking data. The runtime for a single iteration of the mean shift algorithm is $O(m^2)$. The mean shift step may be skipped if time is at a premium.

For the validation studies, the incremental classification step must be executed m times, in order to construct an ROC curve and extract the optimal classification threshold. For a production system, this classifier construction step only has to be run once. Once the classifier is built, provided no scanpaths are added to the training data, all classifications only take the time needed for an incremental classification step. Most eye tracking applications are analyzed offline. In this case, running time is far less important. If an eye tracker classifier were desired for use as a type of biometric mechanism, some heuristics may need to be adopted to ensure acceptable response times, such as only sampling frames that have been chosen, ahead of time, as informative.

6.2 Future Work

Future work may include work on various kinds of adaptation. It may be useful to weight the training data in different ways and on different scales. First, subjects' data may be weighted. Supposing some subjects used for training data score more highly than others when performing intra-class comparison, those subjects' data may be given more weight than lower-scoring instances. Also, adaptation over time may be useful. For example, data collected within and between different scenes in a movie would be expected to weight the final classification differently. Data collected inside a single scene would be expected to have higher similarity than data from two different scenes. Perhaps only a few intra-scene timestamps need be evaluated. This would be a method of both speeding up the algorithm and possibly increasing reliability.

The cross-stimulus classification reinforcement has been shown to be useful in making classification more reliable for individual subjects. An improvement here would be to apply the cross-validation approach. Currently, a constant greater than 50% stimulus classifiers must claim it. It may be that the training data would yield insight into whether that value should be shifted. Perhaps some classes would require much more than half or even less than half. Optimally, some form of boosting would be used for this computation, such as the Adaboost algorithm (Freund and Schapire [1995]).

Another aspect of the research that could be expanded is analysis of which tasks lend themselves better toward classification than others. In the case of the video study, the general task of "watch for something unusual" appeared to be less classifiable than "avoid looking at faces." Perhaps a different instruction, such as "count the number of shoes" would have yielded more discriminable data. It would also be

useful to evaluate the effects of audio on visual behavior.

The metrics described in the section on video classification do not adequately relay certain aspects of the video clips used in the experiment, such as the differing lengths of the clips and what ratio of each clip was either discriminable or conformant. Ideally, both of these aspects could be merged into an overall "expected discriminability" metric and then compared to the accuracy of the classifier, itself. Such a metric would be useful for evaluating the relative performance of the classifier on clips of differing lengths. This "confidence metric" would need to be able to handle such extreme cases as two groups yielding highly conformative behavior throughout most of a stimulus, except for a very brief period of time in which the groups greatly diverge. The saliency metric is a first step toward this type of mechanism.

The current approach is specifically oriented toward two-class classifiers, though it is able to be extended to more than two classes through a complicated and time-consuming procedure of evaluating the results for individual classes against all other classes and between all pairs of classes. There is likely a simpler and more elegant approach to multi-class classification.

A topic of interest to the eye tracking community is the use of computation models of human visual attention, such as the model defined by Itti et al. [1998]. This particular model encompasses a bottom-up methodology, estimating information content of groups of pixels in an image, rather than a top-down approach, which might attempt to infer what objects are present in a stimulus and which objects a human observer might specifically be interested in. Preliminary results appear to show that scanpaths constructed from the described model on the film clips described in chapter 4 are different enough from scanpaths collected from hu-

man observers that the classifier is able to achieve complete accuracy. Future work could include defining which types of stimuli are or are not easily classifiable, which would indicate the convergence or divergence of the visual attention model with real human visual attention recordings.

6.3 Conclusion

In this dissertation, a novel method of event-driven aggregate scanpath comparison has been proposed and empirically validated. This metric calculates similarity by quantifying the overlap of Gaussian fixations over multiple events. These events are defined by the type of stimulus used for the experiment, such as individual frames for video stimuli. The similarity metric is then used to drive an ROC-based classifier framework. This allows the automatic classification of scanpaths into predefined groups for certain visual tasks. Results indicate far better discriminative ability than previous metrics, specifically string-editing distance, in most visual tasks. In addition to the original expert/novice pilot study used as proof of concept with still images, three developments have been evaluated. The first is an application of this new method to its intended purpose, mainly analysis of scanpaths over video. This development has been empirically validated. Although not capable of discriminating all types of tasks, it was able to reliably classify two out of the three stimuli and visual tasks performed. The task it was not able to discriminate implies that the instructions given were not specific enough to alter visual behavior as much as the other two tasks were. The second is a modification intended to expand its applicability to some interactive tasks, which was utilized to attempt to distinguish differences in visual search strategies over different cultures, the third task. The extension intended for interactive tasks showed improved classification capability over string-editing in this task, similar to the expert/novice pilot study. The classifier

was able to discriminate between cultures for two visual search tasks: symbol search and numeric search.

Bibliography

Airola, A., Pahikkala, T., Waegeman, W., De Baets, B., and Salakoski, T. 2009. A Comparison of AUC Estimators in Small-Sample Studies. In *Proceedings of the 3rd International workshop on Machine Learning in Systems Biology*. 15–23.

Anliker, J. 1976. Eye Movements: On-Line Measurement, Analysis, and Control. In *Eye Movements and Psychological Processes*, R. A. Monty and J. W. Senders, Eds. Lawrence Erlbaum Associates, Hillsdale, NJ, 185–202.

Bednarik, R., Myller, N., Sutinen, E., and Tukiainen, M. 2005. Applying Eye-Movement Tracking to Program Visualization. In *VLHCC '05: Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing*. IEEE Computer Society, Washington, DC, USA, 302–304.

Daugherty, B. 2009. Ocular Vergence Response Over Anaglyphic Stereoscopic Video. M.S. thesis, Clemson University, Clemson, SC.

Dempere-Marco, L., Hu, X.-P., Ellis, S. M., Hansell, D. M., and Yang, G.-Z. 2006. Analysis of Visual Search Patterns With EMD Metric in Normalized Anatomical Space. *IEEE Transactions on Medical Imaging 25,* 8 (August), 1011–1021.

Duchowski, A., Medlin, E., Cournia, N., Gramopadhye, A., Nair, S., Vorah, J., and Melloy, B. 2002. 3D Eye Movement Analysis. *Behavior Research Methods, Instruments, Computers (BRMIC) 34,* 4 (November), 573–591.

Duchowski, A. T. 2007. *Eye Tracking Methodology: Theory & Practice*, 2nd ed. Springer-Verlag, Inc., London, UK.

Duchowski, A. T., Driver, J., Jolaoso, S., Ramey, B. N., Robbins, A., and Tan, W. 2010. Scanpath Comparison Revisited. In *Eye Tracking Research & Applications (ETRA)*. ACM, Austin, TX.

Duchowski, A. T. and McCormick, B. H. 1998. Gaze-Contingent Video Resolution Degradation. In *Human Vision and Electronic Imaging III*. SPIE, Bellingham, WA.

Fischer, P. and Peinsipp-Byma, E. 2007. Eye Tracking for Objective Usability Evaluation. In *European Conference on Eye Movements (ECEM)*. ECEM.

Freund, Y. and Schapire, R. E. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT '95: Proceedings of the Second European Conference on Computational Learning Theory*. Springer-Verlag, London, UK, 23–37.

Galgani, F., Sun, Y., Lanzi, P., and Leigh, J. 2009. Automatic analysis of eye tracking data for medical diagnosis. In *Proceedings of IEEE Symposium on Computational Intelligence and Data Mining (IEEE CIDM 2009)*. IEEE.

Goldberg, J. H., Stimson, M. J., Lewenstein, M., Scott, N., and Wichansky, A. M. 2002. Eye tracking in web search tasks: design implications. In *ETRA '02: Proceedings of the 2002 symposium on Eye tracking research & applications*. ACM, New York, NY, USA, 51–58.

Grindinger, T. 2006. Eye Movement Analysis & Prediction with the Kalman Filter. M.S. thesis, Clemson University, Clemson, SC.

Grindinger, T., Duchowski, A. T., and Sawyer, M. 2010. Group-Wise Similarity and Classification of Aggregate Scanpaths. In *Eye Tracking Research & Applications (ETRA)*. ACM, Austin, TX.

Hembrooke, H., Feusner, M., and Gay, G. 2006. Averaging Scan Patterns and What They Can Tell Us. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, San Diego, CA, 41.

Higgins, D. G. 1988. Clustal : A package for performing multiple sequence alignment on a microcomputer. *Gene 73*, 237–244.

Itti, L., Koch, C., and Niebur, E. 1998. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 20,* 11, 1254–1259.

Jarodzka, H., Holmqvist, K., and Nyström, M. 2010. A vector-based, multidimensional scanpath similarity measure. In *ETRA '10: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. ACM, New York, NY, 211–218.

Kearns, M. 1988. Thoughts on hypothesis boosting. *Unpublished manuscript*.

Kramer, A. F. and McCarley, J. S. 2003. Oculomotor Behaviour as a Reflection of Attention and Memory Processes: Neural Mechanisms and Applications to Human Factors. *Theoretical Issues in Ergonomics Science 4,* 1–2, 21–55.

Leigh, R. J. and Zee, D. S. 1991. *The Neurology of Eye Movements*, 2nd ed. Contemporary Neurology Series. F. A. Davis Company, Philadelphia, PA.

Nair, S., Medlin, E., Vora, J., Gramopadhye, A., Duchowski, A. T., and Melloy, B. 2001. Cognitive Feedback Training Using 3D Binocular Eye Tracker. In *Proceedings of the Human Factors and Ergonomics Society*.

Olson, D. L. and Delen, D. 2008. *Advanced Data Mining Techniques*. Springer.

Paris, S. and Durand, F. 2006. A Fast Approximation of the Bilateral Filter using a Signal Processing Approach. Tech. Rep. MIT-CSAIL-TR-2006-073, Massachusetts Institute of Technology.

Parker, B., Gunter, S., and Bedo, J. 2007. Stratification bias in low signal microarray studies. *BMC Bioinformatics 8,* 1 (September), 326+.

Pomplun, M., Ritter, H., and Velichkovsky, B. 1996. Disambiguating Complex Visual Information: Towards Communication of Personal Views of a Scene. *Perception 25,* 8, 931–948.

Privitera, C. M. and Stark, L. W. 2000. Algorithms for Defining Visual Regions-of-Interest: Comparison with Eye Fixations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 22,* 9, 970–982.

Räihä, K.-J., Aula, A., Majaranta, P., Rantala, H., and Koivunen, K. 2005. Static Visualization of Temporal Eye-Tracking Data. In *INTERACT*. IFIP, 946–949.

Salvucci, D. D. and Goldberg, J. H. 2000. Identifying Fixations and Saccades in Eye-Tracking Protocols. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, Palm Beach Gardens, FL, 71–78.

Santella, A. and DeCarlo, D. 2004. Robust Clustering of Eye Movement Recordings for Quantification of Visual Interest. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, San Antonio, TX, 27–34.

Sawyer, M. W. 2009. Scanpath Comparison of the Salient Features of Weather for General Aviation Pilots based on Training and Experience. Ph.D. thesis, Clemson University, Clemson, SC.

Swets, J. A. 1988. Measuring the accuracy of diagnostic systems. *Science 240,* 4857 (June), 1285–1293.

Tobii Technology AB. 2003. Tobii ET-17 Eye-tracker Product Description. (Version 1.1).

Torstling, A. 2007. The Mean Gaze Path: Information Reduction and Non-Intrusive Attention Detection for Eye Tracking. M.S. thesis, The Royal Institute of Technology, Stockholm, Sweden. Techreport XR-EE-SB 2007:008.

Wooding, D. S. 2008. Fixation Maps: Quantifying Eye-movement Traces. *Proceedings of ETRA '02*, 31–36.

Yarbus, A. L. 1967. *Eye Movements and Vision.* Plenum Press, New York, NY.