Scanpath Comparison Revisited

Andrew T. Duchowski,* Jason Driver
School of Computing, Clemson University

Sheriff Jolaoso
Computer Engineering, UMBC

Beverly N. Ramey, Ami Robbins
Computer Science, Winthrop University

William Tan
Computer Engineering, Syracuse University

Abstract

The scanpath comparison framework based on string editing is revisited. The previous method of clustering based on k-means "preevaluation" is replaced by the mean shift algorithm followed by elliptical modeling via Principal Components Analysis. Ellipse intersection determines cluster overlap, with fast nearest-neighbor search provided by the kd-tree. Subsequent construction of Y-matrices and parsing diagrams is fully automated, obviating prior interactive steps. Empirical validation is performed via analysis of eye movements collected during a variant of the Trail Making Test, where participants were asked to visually connect alphanumeric targets (letters and numbers). The observed repetitive position similarity index matches previously published results, providing ongoing support for the scanpath theory (at least in this situation). Task dependence of eye movements may be indicated by the global position index, which differs considerably from past results based on free viewing.

CR Categories: H.1 [Information Systems]: Models and Principles—User/Machine Systems. I.5 [Computing Methodologies]: Pattern Recognition—Clustering. J.4 [Computer Applications]: Social and Behavioral Sciences—Psychology.

Keywords: eye tracking, scanpath comparison

1 Introduction

Sequences of fixations, or scanpaths, have been used for compelling visualizations of captured eye movements since the early 1970s [Noton and Stark 1971], but have as yet not been fully exploited for their quantitative potential. There is a pressing need for quantitative scanpath comparison metrics. An easy to use computational approach is sought that is analogous to statistical packages that quickly and easily generate tables of means and ANOVA statistics from experimental data. We revisit Privitera and Stark's [2000] string editing approach, one that computes similarity (or "distance") between pairs of scanpaths along with statistical levels of significance. The resulting metric is similar to Spearman's rank-order coefficient [Boslaugh and Watters 2008] but yields a value $S \in [0, 1]$ instead of $S \in [-1, 1]$. The novelty of our contribution is two-fold. First, our algorithm improves upon Privitera and Stark's scanpath comparison by substituting k-means clustering with Santella and DeCarlo's [2004] mean shift. The former generally requires *a priori* knowledge of the number of clusters [Duda and Hart 1973] whereas mean shift in comparison is self-organizing, starting with as many cluster means as there are fixations. Second, Principal

*e-mail: duchowski@clemson.edu

	Subj. 1		Subj. 2	
	Pict 1	Pict 2	Pict 1	Pict 2
S1P1	**R**	**I**	**L**	**G**
S1P2		**R**	**G**	**L**
S2P1			**R**	**I**
S2P2				**R**

	Same Subj. (SS)	Diff. Subj. (DS)
Same Img. (SI)→	**R**epetitive	**L**ocal
Diff. Img. (DI)→	**I**diosyncratic	**G**lobal
		Random

Figure 1: *A scanpath comparison Y-matrix and parsing diagram are created for each of S_s and S_p sequence and position similarity indices. Adapted from Privitera and Stark [2000].*

Components Analysis is used to model elliptical cluster boundaries that are in turn used to calculate overlap among clusters belonging to different scanpaths (other approaches are possible, e.g., convex hull, but ellipses lend themselves to straightforward boundary intersection evaluation whereas convex hulls may not). A kd-tree is used to spatially partition each scanpath's clusters for efficient nearest-neighbor queries used to determine cluster overlap, yielding automatic cluster labeling for string-based comparison.

2 Background

Privitera and Stark's [2000] scanpath comparison based on string editing was one of the first methods to quantitatively compare not only the loci of fixations S_p but also their order S_s (see Figure 1). Defined by an optimization algorithm, string editing assigns unit cost to three different character operations: *deletion*, *insertion*, and *substitution*. Characters are then manipulated to transform one string to another, and character manipulation costs are tabulated. For example, with two comparison strings $s_1 = abcfeffgdc$ and $s_2 = afbffdcdf$, the total cost of the combination of deletions, insertions, and substitutions in this case is 6 (see below). The total cost is normalized to the length of the longer string, in this case 9, yielding a sequence similarity index between the two strings of $S_s = (1 - 6/9) = 0.33$. A positional similarity index can be found for the two strings by comparing the characters of the second string to those of the first. Since all the characters of s_2 are present in s_1, the two strings yield a loci similarity index of $S_p = 1$.

For scanpaths from multiple viewers, similarity coefficients are sorted and stored in a table, named the Y-matrix, having as many rows and columns as the number of different sequences to be considered. Scanpath comparison values from the Y-matrix (which typically contains large amounts of data) are condensed (averaged) and reported in two tables, called Parsing Diagrams, one for each of S_p and S_s indices. The S_p statistic gives the correlation between two string sequences in terms of attentional loci (the order of fixations is not considered in this measure). S_p values are generally expected to be higher than the measure reporting on order of fixations, the S_s statistic. Each of the parsing diagrams reports several correlation measures: **R**epetitive, **I**diosyncratic, **L**ocal, and **G**lobal. Repetitive values report an individual's propensity to view a specific

image in the same way. Idiosyncratic values report on the within-subject attentional scanning tendencies of individual subjects, i.e., these values report correlation between scanpaths made over different pictures by the same subject. For example, these values should be large if a person tends to exhibit a similar strategy when viewing a particular stimulus. Local indices report on between-subject correlations of scanning patterns over similar stimuli, i.e., on different subjects' scanpaths over the same picture. For example, in reading studies, English readers would be expected to exhibit high local indices due to the adopted left-to-right text scanning pattern. Global measures report on the correlation between scanpaths made by different subjects over different stimuli. Should these values be highly correlated, this would suggest that stimulus images tend to be viewed similarly by different people.

The string editing methodology has since been employed in several studies where scanpath comparison was required. Josephson and Holmes [2002] may have been the first to evaluate web page design with Brandt and Stark's [1997] technique. Their results were mixed, however. Some individuals displayed scanpaths that resembled each other over time. However, they also found many instances in which the most similar sequences were from different subjects rather than from the same subject. Their study was descriptive in nature with no tests of significance. More recently, Josephson and Holmes [2006] again used string editing to evaluate on-screen television enhancements such as headline bars and bottom-of-the-screen crawlers. Their study revealed that screen design impacted news story content recall. In both of their studies, the viewing stimulus was partitioned into Region Of Interest *a priori*, thus precluding the need for automatic cluster analysis.

With slightly differing objectives, Hembrooke et al. [2006] used string editing to investigate the amalgamation of numerous scanpaths into a single, representative scanpath. Since string editing essentially defines a multiple sequence alignment algorithm, the final alignment is a pattern constructed from similarities among multiple input patterns. Therefore, one can apply this approach to the construction of something resembling the "ideal observer", or "average expert" over a given visual stimulus.

More recently, West et al.'s [2006] *eyePatterns* substituted Levenshtein similarity with the Needleman-Wunsch distance, increasing flexibility through variable scoring parameters. However, finding similarity and distance alignments are duals of each other where "large distance" is "small similarity" [Waterman 1989].

The only other approaches aimed at scanpath comparison are either limited in capability (Myers and Schoelles's [2005] *ProtoMatch* lacks the ability to perform cluster-type analyses), or take an entirely different trajectory-based approach whose applicability to scanpaths is as yet uncertain [Vlachos et al. 2002; Vlachos et al. 2004; Torstling 2007]. The use of typical distance functions for measuring the similarities of trajectories recorded in Euclidean space was dismissed by Vlachos et al. [2002] due to its sensitivity to outliers and intermediate points, time between regions, and trajectory differences in unrelated areas.

3 Pairwise Scanpath Comparison

The present approach extends prior work [Duchowski et al. 2003; Heminghous and Duchowski 2006] which followed Privitera and Stark's [2000] method closely. Strings for each scanpath are constructed by concatenating the characters from each successive cluster of the scanpath. The Levenshtein similarity is computed by an optimization algorithm that builds an $n \times m$ array (where n and m are the string lengths) and finds the minimum cost to transform one string into the other. The array A is defined as

	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: *Example of Levenshtein distance calculation.*

$$A[i][j] = \min \begin{cases} A[i-1][j\quad] + 1 \\ A[i\quad][j-1] + 1 \\ A[i-1][j-1] + c(i,j). \end{cases}$$

The first two terms in the minimization handle the costs of *deletions* and *insertions*, and the last term handles *substitutions*, with

$$c(i,j) = \begin{cases} 0, \; s_1[i-1] = s_2[j-1] \\ 1, \; \text{otherwise.} \end{cases}$$

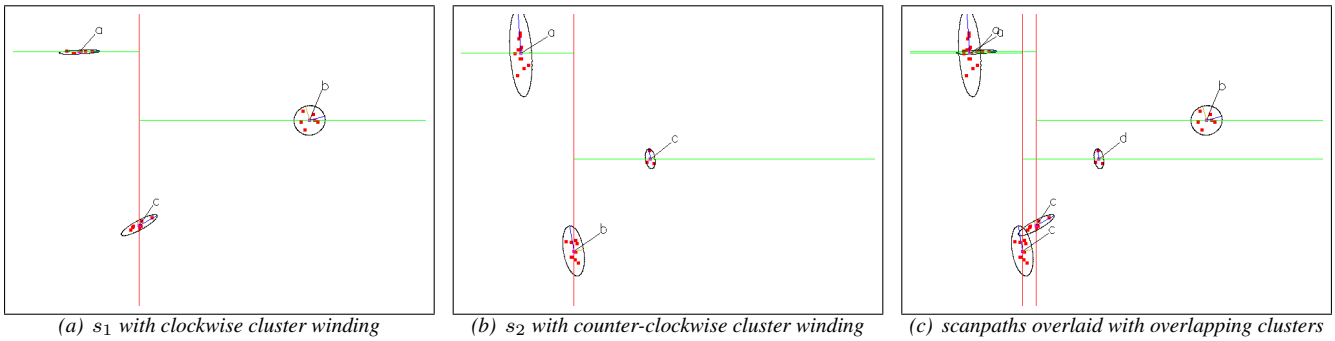
The array's first row and column must be initialized with ascending integers ($[0..m]$ and $[0..n]$) as a pre-processing step. Continuing with the previous example with $s_1 = abcfeffgdc$ and $s_2 = afbffdcdf$, the 10×9 array that would be generated is illustrated in Figure 2. The cost to completely transform one string into another is found at the bottom right most entry of the array. The intermediate values provide the costs of partial transformations.

Following cluster labeling, scanpath comparison is implemented as per Privitera and Stark [2000]. Levenshtein's string similarity (for scanpath strings obtained by concatenating cluster labels) gives the pairwise sequential coefficient S_s. The positional coefficient S_p is related to the number of labels shared between string pairs. Pairwise coefficients are stored in the Y-matrix, consisting of as many rows and columns as the number of sequences being compared.

By clustering (many viewers') aggregate fixations, the automatic labeling scheme employed previously (e.g., by Heminghous and Duchowski [2006]) marred the distinction of individuals' scanpaths and concealed cluster overlap. Instead of clustering fixation points *en masse*, scanpaths are now clustered independently. Subsequently, multiple viewers' scanpaths, each defined as a sequence of clusters (modeled by ellipses), are tested for cluster overlap. Intersecting ellipses are assigned identical character labels. The algorithm's computational efficiency is drawn from fast proximity queries provided by the kd-tree spatial subdivision data structure. The resultant cluster labeling leads to the computation of scanpath similarity (e.g., via string editing comparison).

Clustering depends on detection of fixations within the raw gaze point data stream. Timestamped fixations $\mathbf{x} = (x, y, t)$ are detected via a variant of the *position-variance* approach [Anliker 1976]. This technique defines a fixation by a centroid and variance indicating spatiotemporal distribution. If the variance of a given gaze point is above threshold, it is considered to be part of a saccade, otherwise it is labeled a fixation. In the present implementation, 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.[1]

[1]The position-variance fixation analysis code was originally made



(a) s_1 with clockwise cluster winding *(b) s_2 with counter-clockwise cluster winding* *(c) scanpaths overlaid with overlapping clusters*

Figure 3: *Arbitrary scanpaths with overlapping clusters. The scanpath in (a) is wound in clockwise order. The scanpath in (b) is wound counter-clockwise. Considered independently, the two scanpaths' labels would be $s_1 = s_2 = abc$. When overlapped, as shown in (c), and taking overlapping clusters into into account, the resulting labels are generated as $s_1 = abc$ and $s_2 = acd$, indicating (spatial) cluster overlap at two locations (note that temporal overlap is ignored—if it were not, clusters labeled a and c might not overlap if fixated at different timestamps). Horizontal and vertical lines indicate spatial partitioning provided by the kd-tree.*

4 Automatic Cluster Labeling

The key to string-based scanpath comparison is proper automatic labeling of sequential clusters, ensuring that identical labels are assigned to overlapping clusters, as illustrated by the example given in Figure 3.

Following Santella and DeCarlo [2004], clustering starts with a set of n points: $\{\mathbf{x}_i \mid i \in 1 \ldots n\}$, each with $\mathbf{s}(\mathbf{x}_i)$, a weighted mean of nearby points, initially set to $\mathbf{s}(\mathbf{x}_i) = \mathbf{x}_i,\ \forall i$. The first stage of the clustering algorithm—the mean shift—is crucial, as it ordains the robustness of the entire process. The process iterates by repeatedly shifting each point's $\mathbf{s}(\mathbf{x}_i)$ to a new location based on the kernel function K:

$$\mathbf{s}(\mathbf{x}_i) = \frac{\sum_j K(\mathbf{x}_i - \mathbf{x}_j)\mathbf{x}_j}{\sum_j K(\mathbf{x}_i - \mathbf{x}_j)},$$

where K is typically a multivariate zero-mean Gaussian with covariance $\sigma^2\mathbf{I}$. With fixations expressed as $\mathbf{x}_i = (x_i, y_i, t_i)$, the following zero-mean spatiotemporal Gaussian can be used:

$$K([\mathbf{x}_i, t_i]) = \exp\left(\frac{x_i^2 + y_i^2}{\sigma_s^2} + \frac{t_i^2}{\sigma_t^2}\right) \qquad (1)$$

where σ_s and σ_t determine local support of the kernel in both spatial (dispersion) and temporal extent. If temporal overlap is of no concern (e.g., regression eye movements, or refixations, do not need to be distinguished), the temporal dimension can simply be excluded by setting $\sigma_t = \infty$.

Unlike k-means clustering used by Privitera and Stark [2000], the mean shift eliminates the need for *a priori* estimation of the number of clusters. The only existing user-adjustable parameters are σ_s and σ_t, which can epistemically be set to match the extent of the foveal dimension of the human retina (about 5° visual angle) and typical expected fixation duration. In the present implementation, $\sigma_s = 50$ pixels (at a resolution of 1280×1024, the spatial extent is 1.5° visual angle) and $\sigma_t = 500$ ms.

available by LC Technologies. At the time of this writing, the original `fixfunc.c` was still found on Andrew R. Freed's web pages: `http://freedville.com/`. The C++ interface and implementation ported from C by Mike Ashmore are currently available at: `http://andrewd.ces.clemson.edu/courses/cpsc412/fall08/`.

4.1 Fitting Ellipses to Fixation Clusters

Cluster labeling relies on two initial steps inspired by Hoppe's [1994] surface reconstruction from unorganized points. In the first step, Principal Components Analysis is performed to fit an axis-aligned ellipse centered at the centroid $\mathbf{o}_k = (c, d)$ of the k^{th} cluster. The centroid is then used to compute the covariance matrix $\mathbf{C}$ of the fixations $\mathbf{x}_i$ contained within the cluster. $\mathbf{C}$ is a symmetric, 2×2 positive semi-definite matrix $\mathbf{C} = \sum_i (\mathbf{x}_i - \mathbf{o}_k) \otimes (\mathbf{x}_i - \mathbf{o}_k)$ where $\otimes$ denotes the outer product vector operator.[2] if $\lambda_k^1 \geq \lambda_k^2$ denote the eigenvalues of $\mathbf{C}$, then the associated unit eigenvectors $\hat{\mathbf{v}}_k^1$, $\hat{\mathbf{v}}_k^2$, are chosen as the cluster's major and minor axes $\mathbf{r}$, $\mathbf{s}$, respectively.

An ellipse is fit to the cluster by representing it by its quadratic equation $Ax^2 + By^2 + Cx + Dy + Exy + F = 0$, with center (c, d) and axes $(\mathbf{r}, \mathbf{s})$ of length $r = \|\mathbf{r}\|$ and $s = \|\mathbf{s}\|$ respectively, with the coefficients obtained as:

$$\begin{aligned}
A &= s^2M^2 + r^2N^2 \\
B &= s^2N^2 + r^2M^2 \\
C &= -2c(s^2M^2 + r^2N^2) - 2MNd(s^2 - r^2) \\
D &= -2d(s^2N^2 + r^2M^2) - 2MNc(s^2 - r^2) \\
E &= 2MN(s^2 - r^2) \\
F &= M^2(s^2c^2 + r^2d^2) + N^2(r^2c^2 + s^2d^2) + \\
&\quad 2MNcd(s^2 - r^2) - r^2s^2,
\end{aligned}$$

with $M = \cos(\theta)$, $N = \sin(\theta)$ for an ellipse rotated about it center by angle $\theta = \tan^{-1}(\mathbf{r}_y/\mathbf{r}_x)$.

4.2 Constructing the kd-tree

In the second step, a kd-tree is constructed for each clustered scanpath. The kd-tree is constructed by first sorting the clusters on their centroid in each dimension ($O(kn \log n)$ and $O(kn)$ storage). At each tree level, the median division of the cluster set is accomplished in $O(kn)$ time. The kd-tree construction algorithm given in Algorithm 1 is widely available, e.g., see Weiss [2006].

[2] If $\mathbf{a}$ and $\mathbf{b}$ have components a_i and b_j respectively, then the matrix $\mathbf{a} \otimes \mathbf{b}$ has a_ib_j as its ij-th entry.

```
tree_node kdtree(vector<cluster *> els, int depth)
{
  if(els.empty()) return NULL;
  else {
    // axis depends on depth, cycling through all valid values
    int axis = depth % k;

    // sort point list and choose median as pivot element
    sort(els.begin(), els.end(), ClusterAxisCompare(axis));
    select median from els;

    // create node and construct subtrees
    tree_node node(location = median);
    node.left = kdtree(points in els before median, depth+1);
    node.right = kdtree(points in els after median, depth+1);
    return node;
  }
}
```

Algorithm. 1: *Construction of a balanced kd-tree of n clusters.*

4.3 Character Labeling

Once each scanpath is represented by a kd-tree of clusters (i.e., its clusters are arranged spatially for efficient nearest-neighbor queries), the algorithm then labels all scanpaths iteratively, by selecting labels for cluster pairs whose means are within σ_s and whose ellipses intersect at 2^+ points. For each pair of scanpaths:

1. For each cluster of the first scanpath, compare with every cluster of the second. Iteratively find the k^{th} nearest neighbor (starting with $k = 1$) until no cluster intersections are found.[3] The nearest-neighbor search has been shown to run in $O(\log n)$ average time per search.

2. Assign matching labels to overlapping clusters, only if either or both of the clusters are as yet unlabeled.

Note that temporal overlap (σ_t) between clusters of different scanpaths is ignored. While it makes sense to consider time when clustering a particular scanpath's fixations (intra-scanpath clustering), it does not necessarily make sense to do so when labeling multiple scanpaths' clusters (inter-scanpath clustering). Fixation timing is relative to the start of a given recording yet there is no guaranteed temporal synchronization of multiple scanpaths to any external marker. As an example, consider two identical scanpaths, but one shifted a fraction of a second in time (as if the viewer hesitated slightly before performing an exact duplicate of a prior action). The lack of temporal overlap would preclude high sequential correlation between scanpaths. Reducing the dimensionality of inter-scanpath cluster comparison by ignoring σ_t effectively eliminates clustering mismatches due to a potential temporal shift between scanpaths.

For position similarity (S_p) computation, duplicate characters are removed from the string representing a scanpath sequence.

4.4 Scanpath Visualization

Current scanpath visualization (see Figure 4) relies on transparency to facilitate viewing multiple scanpaths (each rendered in a randomly drawn color). Each scanpath is made up of the raw gaze point sequence, detected fixation sequence, and sequence of clustered fixations. Following cluster labeling, each cluster center is annotated with its character label. In practice each scanpath cluster label is indexed by an integer, but to facilitate visualization, the integer is mapped sequentially to its ASCII representation, restricted to the range of capitalized letters (65–90).

[3] Back-of-the-envelope proof of correctness: if the nearest neighbor does not intersect the given reference cluster, no others can since they are all farther away.

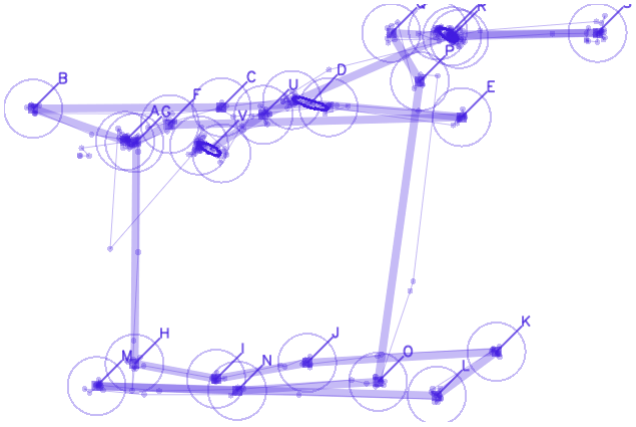


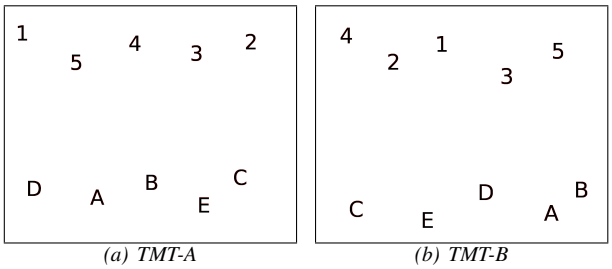
Figure 4: *Example scanpath visualization. Small circles connected by thin lines denote raw gaze points. Large circles connected by thick lines denote fixations. Fixation clusters are identified by large squares (fixation centroids) and thick ellipses (fixation clusters). Following cluster labeling, cluster centers are annotated with capitalized characters in alphabetic order starting with 'A'.*

4.5 Random Scanpath Generation

To facilitate testing for statistical significance, random scanpaths are generated by a simulator for comparison with actual scanpaths. The simulator emulates an eye tracker operating at 50 Hz. During each iteration, a new gaze point is either created within close proximity to the current fixation or a saccade is initiated. A new saccade's coordinates are determined according to a normal distribution $\mathcal{N}(\mu, \sigma')$ with μ set to the screen center and σ' set to a sixth of the display width and height. Fixations are modeled by a Poisson distribution with a mean of 1300 ms. New gaze points that are part of a fixation are modeled by a normally distributed offset characterized by $\mathcal{N}(0, \sigma'')$ with σ'' set to 0.91 visual angle from the previous location, or about 30 pixels. The choice of 30 pixels is not unusual; it is actually more conservative than the typical choice of 50 pixels in common dispersion-based (position-variance) fixation detection algorithms.

5 Empirical Validation

An experimental paradigm was sought whose design would elicit similar scanpaths from participants. A gaze-directed variant of the Trail Making Test protocol [Bowie and Harvey 2006] was chosen for this purpose as it asks participants to visually connect specific and easily identifiable targets (numbers and letters). The most widely used version of the TMT comprises parts A and B. In part A, the participant connects a series of numbers in numerical order, followed by a series of letters in alphabetical order. In part B, the participant connects an interleaved sequence of alternating numbers and letters, still in the same sequential order (see below). The TMT is sensitive to a variety of neurological impairments and processes, and is believed to measure the cognitive domains of processing speed, sequencing, mental flexibility, and visual-motor skills. Part A is generally presumed to be a test of visual search and motor speed skills, whereas part B is considered to also be a test of higher level cognitive skills.



(a) TMT-A *(b) TMT-B*

Figure 5: *Stimulus images used for both training and trial of TMT parts A and B, inverted for print reproduction; actual images used red letters and numbers atop a black background.*

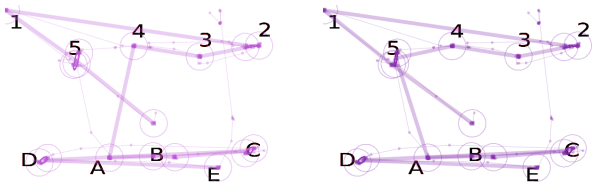
In its normal invocation, the TMT's main dependent variable of interest is the total time to completion of both parts A and B. In its present instantiation, the primary measure of interest is the scanpath. Although the scanpath inherently encodes processing time in its total duration, the main concern here is its spatial distribution and ordering. The two parts of the test readily provide two images over which idiosyncratic similarity indices can be computed. Repetitive scores are obtained by recording two scanpaths over a single image. Local and global indices are gathered by having multiple participants perform the test.

Subjects. Six college students participated in the study (4 M, 2 F; ages 18-27, median age 21). Results from the TMT protocol should generally be stratified by age and education [Tombaugh 2004]; the present sample represents one such age and education strata.

Stimulus. Two images were used as stimulus, with numbers and letters distributed pseudo-randomly (numbers on top, see Figure 5).

Procedure. Each session started with a short, 5-point calibration sequence. During the first half of the session (TMT-A portion), participants were asked to fixate the sequence of numerals followed by the sequence letters, i.e., 1-2-3-4-5-A-B-C-D-E. The first image was viewed twice, once during training, then again during the trial. During the second half of the session, participants were asked to fixate the sequence of numerals interleaved with the sequence letters, i.e., 1-A-2-B-3-C-4-D-5-E. The second image was again viewed twice, once during training, then again during the trial. Participants were asked to view the sequences as quickly as possible but while doing so to dwell over each number or letter for a fraction of a second (they were aware of the underlying fixation algorithm).

Apparatus. A Tobii ET-1750 video-based corneal reflection (binocular) eye tracker was used for real-time gaze coordinate measurement (and recording). The eye tracker operates at a sampling rate of 50 Hz with an accuracy typically better than 0.3° over a $\pm 20^\circ$ 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 of the screen). 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 (connected via a switch on the same subnet). Participants sat at a viewing distance of about 50 cm from the monitor, the tracker video camera's focal length.



(a) spatial clustering *(b) spatiotemporal clustering*

Figure 6: *The importance of spatiotemporal clustering. The scanpath in (a) is clustered spatially—notice that the viewer's second fixation over the numeral 5 (a waypoint fixation on the path to numeral 1) is clustered with fixations made later in time, following fixation of numeral 4. The resulting scanpath incorrectly suggests a saccade from the numeral 4 to the letter A. The scanpath in (b) is clustered spatiotemporally, correctly distinguishing fixations atop numeral 5 as two distinct clusters (in time).*

Experimental Design. Since each of two images was viewed twice in succession, the study follows a basic AABB stimulus presentation order. Note that this differs from the traditional TMT sequence wherein different images are presented for training and trial, e.g., ABCD. In our case, the same image was used for both training and trial because part of the evaluation criteria required repetitive viewing, i.e., same subject viewing the same image more than once. All participants performed the TMT-A portion before performing the TMT-B portion of the test. No counterbalancing was imposed because unlike traditional application of the TMT, we were not concerned with traditional performance metrics (time to completion).

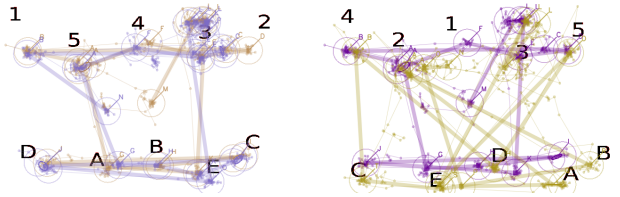
5.1 Pilot Testing

Pilot testing revealed the importance of both spatial and temporal support during mean-shift fixation clustering. By setting $\sigma_t = \infty$ in Equation (1), local support of the kernel K in time is lost. This implies that a cluster created at an initial fixation will cluster any fixations made in the vicinity at any later time. This situation is illustrated in Figure 6 where a saccade to the numeral 5 is missed because of an earlier fixation made in its proximity.

An individual's scanpaths over the TMT-A images (e.g., Figure 7) allow evaluation of **R** and **I** indices and facilitate visualization (multiple scanpaths are difficult to decipher as their number increases). As expected, higher similarity indices are gathered from repetitive scanpaths than from idiosyncratic ones. Assuming that the S_s and S_p indices are analogous to Spearman's rank-order coefficient, $0.9 \leq S \leq 1$ would suggest very strong correlation (probably unlikely for S_p due to various factors, e.g., eye tracker error), $0.7 \leq S \leq 0.9$ suggests strong correlation (probably more likely for S_p than S_s), and $0.5 \leq S \leq 0.7$ suggests moderate correlation (probably more likely for **R** and **L** indices than for **I** and **G**).

5.2 Aggregate Results

Figures 8(a) and 8(b) summarize the aggregate analysis. To tabulate the parsing diagrams from the Y-matrix created from the 24 scanpaths of all six participants (four scanpaths per individual over the AABB stimulus sequence), only the lower diagonal entries of the 24×24 matrix were used (due to matrix symmetry and exclusion of diagonal entries which hold trivial self-similarity values of 1.0). The 276 lower diagonal entries of the matrix consisted of 12 repetitive, 120 local, 120 global, and 24 idiosyncratic pairings. The degrees of freedom in the F statistics are double these amounts,



(a) repetitive evaluation *(b) idiosyncratic evaluation*

Figure 7: *An individual's scanpaths over the TMT stimuli. Two scanpaths in (a) recorded over the same image generate repetitive indices $S_s = 0.61$ and $S_p = 0.79$. The scanpath labels are $s_1 = abcdefaghijkllm$ and $s_2 = nbecefaghkijolllllp$, which are pruned of duplicate entries, yielding $s_1 = abcdefghijklm$ and $s_2 = nbecfaghkijolp$ that are used to compute S_p. Comparing s_1 captured over the image used in TMT-A and the scanpath s_3 captured over the image used in TMT-B and shown in (b), generates idiosyncratic indices $S_s = 0.12$ and $S_p = 0.36$. The scanpath labels are unchanged for s_1 and $s_3 = nopqaarstbhdulvl$, with its pruned version $s_3 = nopqarstbhdulv$ used for computation of S_p.*

reflecting comparison of similarly indexed entries in the random Y-matrix.

To gauge statistical significance of the parsing diagram aggregates, random scanpaths were generated, one for each of the actual scanpaths and of the same duration. A dual random Y-matrix was generated, with as many rows as actual scanpaths and as many columns as random scanpaths (in this instance the number of actual and random scanpaths equaled, although as few as a single random scanpath can be used for this comparison). Each entry of the random Y-matrix thus contained a pairwise similarity value between an actual scanpath and a random scanpath, mirroring the organization of the actual Y-matrix entries. This organization of the data allowed repeated-measures one-way ANOVA between actual and random similarity measures, with the Y-matrix entry serving as fixed factor (and pairwise comparison used as the random factor [Baron and Li 2007]). The F-statistic reported by ANOVA is therefore an indicator of variance between actual-actual scanpath similarity and actual-random similarity. The null hypothesis inherent in ANOVA assumes no difference between the similarity means.

For example, given the 12 actual repetitive Y-matrix entries (i.e., from the two viewings of each of the two images by each of the six participants), 12 random repetitive Y-matrix entries were also generated, comparing each of the repetitive scanpaths to a random scanpath. One-way ANOVA of repetitive similarity suggests a highly significant main effect of the type of comparison (actual-actual versus actual-random) for both position (F(1,22) = 98.2, $p < 0.01$) and sequence (F(1,22) = 34.6, $p < 0.01$) similarities.[4]

5.3 Segregate Results

Aggregate statistics tend to obscure processes that may be related to individual behaviors or stimuli. To probe further, visualization and generation of similarity statistics of selected scanpaths is performed. For example, to examine the claim that part B of the Trail Making Test protocol is a test of higher level cognitive skills, repetitive scanpaths can be compared over each of the TMT-A and TMT-B pairs of images. Because part A of the test relies mainly on visual search and should therefore be easier to execute (fewer errant saccades), a reasonable expectation would be that repetitive (and local) scores should be higher for this portion of the test. Parsing diagrams

[4] Assuming sphericity as computed by the statistical package R.

	SS	DS
SI →	**R**epetitive 0.65 F(1,22) = 98.2, $p < 0.01$	**L**ocal 0.47 F(1,238) = 848.2, $p < 0.01$
DI →	**I**diosyncratic 0.44 F(1,46) = 165.4, $p < 0.01$	**G**lobal 0.44 F(1,238) = 884.0, $p < 0.01$
	S_p	Random 0.06

(a) position similarity

	SS	DS
SI →	**R**epetitive 0.35 F(1,22) = 34.6, $p < 0.01$	**L**ocal 0.23 F(1,238) = 148.5, $p < 0.01$
DI →	**I**diosyncratic 0.18 F(1,46) = 52.1, $p < 0.01$	**G**lobal 0.17 F(1,238) = 221.0, $p < 0.01$
	S_s	Random 0.08

(b) sequence similarity

Figure 8: *Parsing diagrams for all data.*

SS	DS		SS	DS
R 0.70	**L** 0.48	← SI →	**R** 0.42	**L** 0.27
I —	**G** —	← DI →	**I** —	**G** —
S_p	Ra —		S_s	Ra —

(a) scanpaths over TMT-A images

SS	DS		SS	DS
R 0.60	**L** 0.44	← SI →	**R** 0.36	**L** 0.22
I —	**G** —	← DI →	**I** —	**G** —
S_p	Ra —		S_s	Ra —

(b) scanpaths over TMT-B images

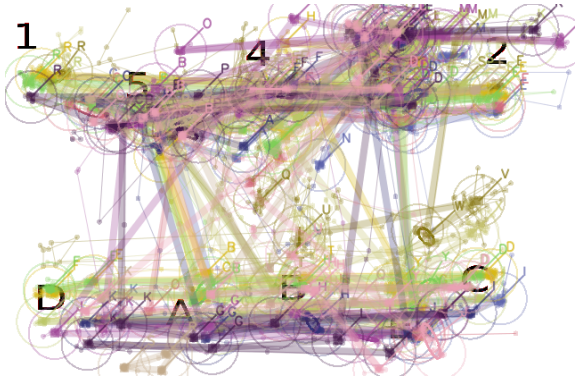
Figure 9: *Parsing diagrams from 12 scanpaths made by all six participants (two scanpaths per individual over each of the AA and BB portions of the AABB stimulus sequence). Comparison with random scanpaths are omitted for brevity.*

for TMT-A and TMT-B are given in Figures 9(a) and 9(b). Visualizations of the scanpaths are given in Figures 10(a) and 10(b).

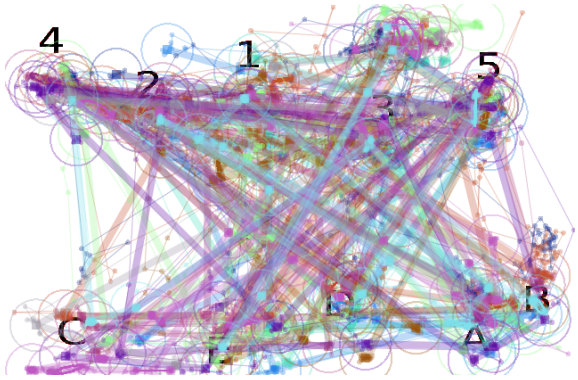
6 Discussion

Aggregate analysis of the six participants' scanpaths shows position indices are generally higher than sequence indices, as expected. Repetitive indices show the highest correlations. This is not surprising given the task stipulated by the Trail Making Test protocol.

Lower S_p and S_s statistics for the TMT-B portion of the protocol (particularly the repetitive values) seem to support the notion of increased cognitive difficulty presented by this task. This may be the first gaze-based evidence supporting this characterization of



(a) Repetitive scanpaths (labeled) over TMT-A



(b) Repetitive scanpaths (labels omitted) over TMT-B

Figure 10: *Repetitive scanpaths over TMT-A (a) and TMT-B (b) images (two scanpaths per individual over the each of the AA and BB portions of the AABB stimulus sequence).*

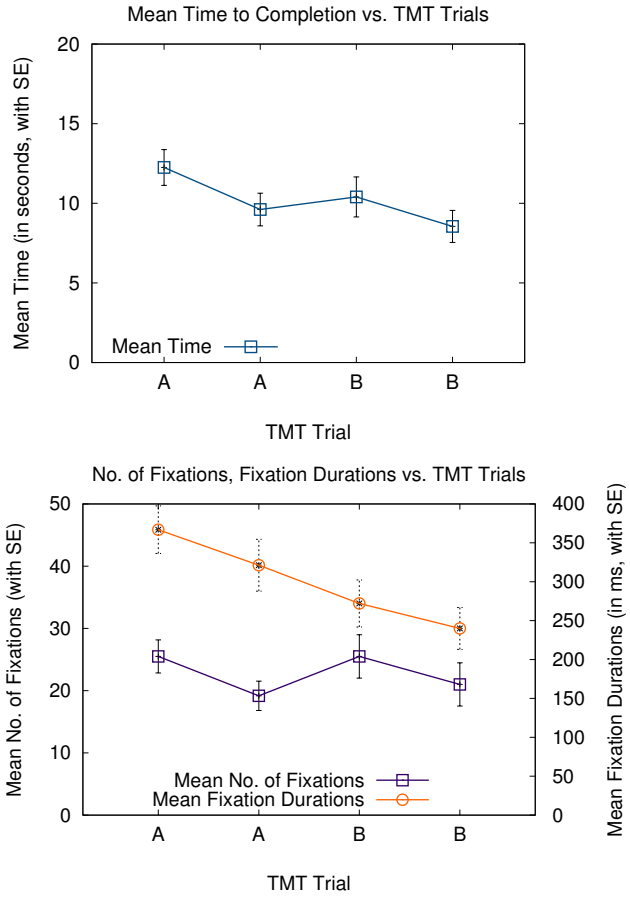


Figure 11: *Performance and process measures across trials.*

the TMT, but it is tenuous since comparison of performance and process measures distilled from scanpaths do not agree with this observation. Specifically, repeated measures ANOVA only shows a marginally significant main effect of trial replicate on time to completion (F(3,15) = 3.38, $p < 0.05$), as measured by scanpath duration length, with mean time to completion tending to decrease from TMT-A to TMT-B (see Figure 11). Pairwise t-tests (with Bonferroni correction) show no significant difference in time to completion between trials. Decreasing time to completion does not support increased cognitive difficulty, rather, it may be indicative of a learning effect, which is likely as task order was not counterbalanced. Indeed, process measures suggest a learning effect as fixation durations decrease significantly across trials (F(3,15) = 6.93, $p < 0.01$), whereas the number of fixations do not (F(3,15) = 2.19, $p = 0.13$, n.s.). Pairwise t-tests show a significant difference in fixation durations during the first A and last B trials (no other significantly different pairings were found).

It is interesting to note how the aggregate similarity indices given in Figure 8 compare with those previously published. The **R** value for our six viewers (0.65) is remarkably close to Privitera and Stark's [2000] seven viewers (0.64). This finding, as in the previous work, suggests that the strings for repetitive viewing have loci that are about 65 percent within fixational or foveal range. Our results therefore provide continuing support for the scanpath theory (the substance of which states that a top-down internal cognitive model drives eye movements). The key difference between the present and prior results is that the task in the present case was quite well defined whereas in prior work it was not (no particular task was assigned). Task dependence of eye movements may be indicated by the **G**lobal position index. In the present case, all different subjects looking at all different stimuli had an S_p value of 0.44 whereas the same index in the previous study was only 0.28.

6.1 Limitations of the Approach

A shortcoming of the given framework is the lack of significance testing between different groups of scanpaths, e.g., testing for significance between similarity of scanpaths made during the TMT-A and TMT-B portions of the experiment (Figure 9). Feusner and Lukoff [2008] suggest computation of the $d^* = d_{between} - d_{within}$ statistic, where $d_{between}$ is the average distance between scanpaths in different groups and d_{within} is the average distance between scanpaths in the same group. To include this computation within the present framework would likely require construction of additional between-group and within-group Y-matrices.

7 Conclusion

Mean shift fixation clustering and subsequent elliptical modeling via Principal Components Analysis enables automation of the string editing approach to scanpath comparison. Construction of a kd-tree facilitates efficient lookup of k nearest cluster neighbors. The combination of these algorithms removes prior reliance on preevaluation and human intervention (interaction). The resulting analysis of multiple scanpaths is computationally efficient, providing output

in the form of parsing diagrams, yielding quantitative measures of scanpath position and sequence similarity. These similarity indices can be used to gain insight into visual processes supporting traditional performance metrics of speed and accuracy.

Scanpath comparison metrics validated empirically by a variant of the Trail Making Test show that position indices should generally be more highly correlated than sequential indices. In particular, given a well-defined visual task, scanpath comparison can be expected to yield moderately correlated repetitive and global position indices.

Acknowledgments

This work was supported in part by CNS grant #0850695 from the National Science Foundation (REU Site: Undergraduate Research in Human-Centered Computing).

References

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.

BARON, J. AND LI, Y. 2007. Notes on the use of R for psychology experiments and questionnaires. Online Notes. URL: http://www.psych.upenn.edu/∼baron/rpsych/rpsych.html (last accessed December 2007).

BOSLAUGH, S. AND WATTERS, P. A. 2008. *Statistics in a Nutshell*. O'Reilly Media, Inc., Sebastopol, CA.

BOWIE, C. R. AND HARVEY, P. D. 2006. Administration and interpretation of the Trail Making Test. *Nature Protocols 1,* 5, 2277–2281.

BRANDT, S. AND STARK, L. 1997. Spontaneous Eye Movements During Visual Imagery Reflect the Content of the Visual Scene. *Journal of Cognitive Neuroscience 9,* 1, 27–38.

DUCHOWSKI, A. T., MARMITT, G., DESAI, R., GRAMOPADHYE, A. K., AND GREENSTEIN, J. S. 2003. Algorithm for Comparison of 3D Scanpaths in Virtual Reality. In *Vision Sciences Society (Posters)*. Sarasota, FL. URL: `http://journalofvision.org/3/9/311/` (last accessed July 2009).

DUDA, R. O. AND HART, P. E. 1973. *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc., New York, NY.

FEUSNER, M. AND LUKOFF, B. 2008. Testing for statistically significant differences between groups of scan patterns. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, New York, NY, 43–46.

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.

HEMINGHOUS, J. AND DUCHOWSKI, A. T. 2006. iComp: A Tool for Scanpath Visualization and Comparison. In *Applied Perception in Graphics & Visualization (APGV)*. ACM, Boston, MA. (Poster).

HOPPE, H. 1994. Surface Reconstruction From Unorganized Points. Ph.D. thesis, University of Washington, Seattle, WA.

JOSEPHSON, S. AND HOLMES, M. E. 2002. Visual Attention to Repeated Internet Images: Testing the Scanpath Theory on the World Wide Web. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, New Orleans, LA, 43–49.

JOSEPHSON, S. AND HOLMES, M. E. 2006. Clutter or Content? How On-Screen Enhancements Affect How TV Viewers Scan and What They Learn. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, San Diego, CA, 155–162.

MYERS, C. W. AND SCHOELLES, M. J. 2005. ProtoMatch: A tool for analyzing high-density, sequential eye gaze and cursor protocols. *Behavior Research Methods, Instruments, Computers (BRMIC) 37,* 2, 256–270.

NOTON, D. AND STARK, L. 1971. Scanpaths in Saccadic Eye Movements While Viewing and Recognizing Patterns. *Vision Research 11*, 929–942.

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.

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.

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

TOMBAUGH, T. N. 2004. Trail Making Test A and B: Normative data stratified by age and education. *Archives of Clinical Neuropsychology 19*, 203–214.

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.

VLACHOS, M., GUNOPULOS, D., AND DAS, G. 2004. Rotation invariant distance measures for trajectories. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 707–712.

VLACHOS, M., KOLLIOS, G., AND GUNOPULOS, D. 2002. Discovering Similar Multidimensional Trajectories. In *ICDE '02: Proceedings of the 18th International Conference on Data Engineering*. IEEE, Washington, DC, 673–685.

WATERMAN, M. S. 1989. Sequence Alignments. In *Mathematical Methods for DNA Sequences*, M. S. Waterman, Ed. CRC Press, Inc., Boca Raton, FL, 53–92.

WEISS, M. A. 2006. *Data Structures and Algorithm Analysis in C++*, 3rd ed. Pearson Education (Addison Wesley), Boston, MA.

WEST, J. M., HAAKE, A. R., ROZANSKI, E. P., AND KARN, K. S. 2006. eyePatterns: Software for Identifying Patterns and Similarities Across Fixation Sequences. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, San Diego, CA, 149–154.