

May 5, 2006

To the Graduate School:

This thesis entitled “iComp: A Scanpath Comparison Tool” and written by John Heminghaus is presented to the Graduate School of Clemson University. I recommend that it be accepted in partial fulfillment of the requirements for the degree of Masters of Science with a major in Computer Science.

Dr. Andrew Duchowski

We have reviewed this thesis
and recommend its acceptance:

Dr. Robert Geist

Dr. Brian Dean

Accepted for the Graduate School:

ICOMP: A SCANPATH COMPARISON TOOL

A Thesis

Presented to

the Graduate School of

Clemson University

In Partial Fulfillment

of the Requirements for the Degree

Masters of Science

Computer Science

by

John Heminghous

May 2006

Advisor: Dr. Andrew Duchowski

ABSTRACT

Visual inspection permeates the lives of all humans. Important criteria for evaluating the efficiency of human visual inspection include measurements of performance, as indicated by the inspector's speed and accuracy. These *performance metrics* effectively summarize general outcomes of the process of (visual) inspection. In contrast, eye movements captured during visual inspection provide visualization of the inspector's process, and therefore provide an instance of *process metrics*. Important eye movement related process measures include fixation durations, locations, and orderings. The utility of process metrics relies on their relation to, or explanation of performance. Previous eye tracking studies have relied solely on performance metrics to evaluate the differences in inspection strategies between experts and novices. We developed iComp, a visual tool that implements the unique feature of direct quantitative scanpath comparison (in loci and sequence), to provide a means for computation of an effective process measure for future expert/novice studies.

ACKNOWLEDGEMENTS

I would like to thank Dr. Andrew Duchowski, my adviser, for his guidance throughout the project and the sharing of his experience, William and Brenda Heminghaus, my parents, for encouraging me to pursue higher education and helping me get there, Thomas Grindinger for providing an image used for testing, and all the faculty of Clemson University for their involvement in the completion of my advanced degree. Finally, I would like to especially thank Mandy Krug, my future wife, for her patience, encouragement, and support through this project and many more to come.

TABLE OF CONTENTS

	Page
TITLE PAGE	i
ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	v
1 INTRODUCTION	1
2 BACKGROUND	4
3 OPERATION	8
4 EXPERIMENTAL VALIDATION	13
5 CONCLUSION	18
BIBLIOGRAPHY	19

LIST OF FIGURES

Figure	Page
1.1 Expert (above) and novice (below) scanpaths, captured at Clemson’s Eye Tracking Laboratory.	1
2.1 Example of string editing. From Privitera and Stark [2000].	4
3.1 Example of Levenshtein distance calculation.	10
3.2 Sample data visualization (numerical image)	11
3.3 Sample data visualization (CG image)	12
4.1 Sample (aggregated) numerical and CG stimulus.	13
4.2 Parsing diagram, numerical images, $\sigma_s = 100$	15
4.3 Parsing diagram, CG image, $\sigma_s = 100$	15
4.4 Parsing diagram, numerical images, $\sigma_s = 70$	16
4.5 Parsing diagram, CG image, $\sigma_s = 70$	16
4.6 Parsing diagram, numerical images, $\sigma_s = 40$	17
4.7 Parsing diagram, CG image, $\sigma_s = 40$	17

CHAPTER 1

INTRODUCTION

Eye tracking has shown itself to be a valuable method of providing data describing the visual attention and cognitive state of a subject. Eye movements bring the fovea, the area centered on the gaze point that is seen in high detail, to Regions of Interest (ROIs) within a scene to be further examined. These inspections are fixations interspersed by rapid eye movements, called saccades. A sequence of fixations is defined as a scanpath. It can be assumed that visual attention follows the fovea. Although this is not always the case (one can attend to an object in their periphery), nonfoveal visual attention is immeasurable and unlikely (without overt effort) in most unrehearsed tasks. Consequently scanpath data can be examined to determine what a subject attends to and consequently where the ROIs are located within a scene.

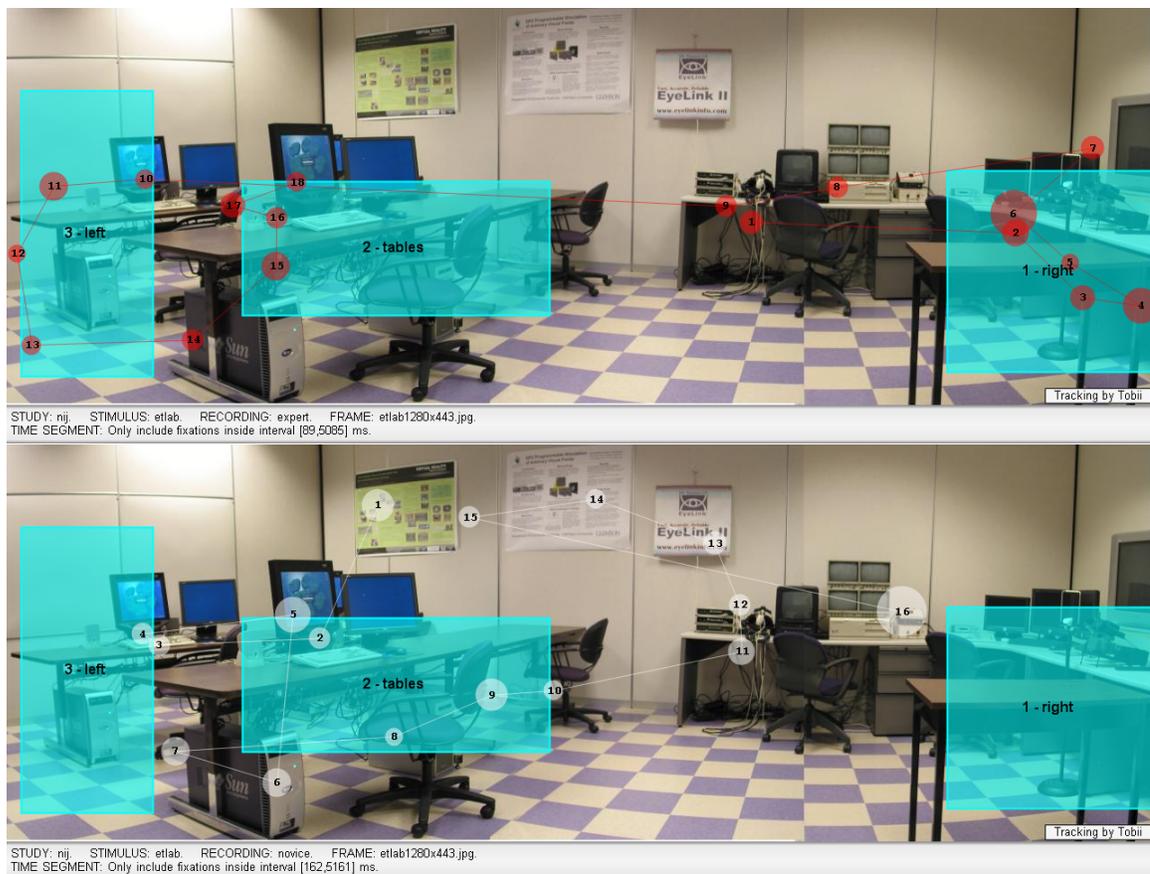


Figure 1.1: Expert (above) and novice (below) scanpaths, captured at Clemson's Eye Tracking Laboratory.

Consider two scanpaths captured during inspection of the eye tracking laboratory at Clemson, shown in Figure 1.1. The top image shows the scanpath of an expert searching a typical room for potential threats. The bottom image shows the scanpath of a novice performing the same task. It is desirable to derive a single objective metric indicating the similarity of the two scanpaths. In this instance, the metric should indicate similarity in one of three viewed ROIs, namely Region 2 (“tables”). Furthermore, a low sequential correlation is expected since in this instance the expert’s scanpath sequence processed the ROIs in sequence {1, 3, 2}, while the novice appears to have processed the ROIs in sequence {2, 3}, although Region 3 was not properly scanned. Note that the data shown in Figure 1.1 is authentic in terms of captured eye movements (on equipment available at Clemson University), however, the inspection task was hypothetical.

This thesis proposes to provide a visual training and assessment methodology based on experts’ eye movements. Experts’ scanpaths will be used to impart to novices a suitable visual search strategy, based on the explicit depiction of the strategy employed by the expert. Assessment of a trainee’s progress will be evaluated off-line via comparison of both search strategies. The basis of the technical software development is the calculation of a quantifiable metric (akin to correlation) indicating the degree of similarity of the visual search strategy adopted by trainees to that of an expert. It is hypothesized that, through training, this metric can provide objective evidence of training effect, particularly for tasks involving visual search.

Schoonard et al. [1973] stated that “visual inspection pervades the lives of all people today. From poultry, meat, and fish inspection, to drug inspection, to medical X-ray inspection, to production line inspection, to photo interpretation, the consequences of inspection directly affect people’s lives through their effects on the quality and performance of goods and services.” Important criteria for evaluating the efficiency of human visual inspection include measurements of performance, as indicated by the inspector’s speed and accuracy. These *performance metrics* effectively summarize general outcomes of the process of (visual) inspection. In contrast, eye movements captured during visual inspection provide visualization of the inspector’s process, and therefore provide an instance of *process metrics*. Important eye movement related process measures include fixation durations, number of fixations, inter-fixation distances, distribution of fixations, fixational sequential indices (order of fixations) and direction of fixations [Megaw and Richardson 1979]. The utility of process metrics relies on their relation to, or explanation of performance. Thus one expects to find a relation between process and performance measures.

Besides being useful for gaging inspection performance, eye movements play a part in training visual search strategy [Wang et al. 1997]. Visual search strategy training can be effective toward the adoption of a desirable (e.g., systematic) search strategy. To train one in a visual search strategy, eye movements can be

used as both a *feedback* mechanism and as confirmation of adoption of the new search strategy. Wang et al. recorded eye movements over visual inspection of artificial printed circuit boards. Eye movements were used to check how well each subject's search pattern followed the strategy trained. Scanpaths were judged by the experimenter after each training trial and feedback was given to the subject about whether they had followed instructions to perform random or systematic search.

Sadasivan et al. [2005] have shown that eye movement scanpaths can also be used as a *feedforward* mechanism for training. They compared the speed and accuracy between experts and novices in finding possible aircraft defects. While these are certainly valid metrics, more specific and informative results could have been obtained from direct scanpath comparison.

Another example of applicable research is presented by Law et al. [2004] concerning surgical training. Again performance metrics were used (speed and accuracy) to evaluate the differences between experts and novices operating a virtual laparoscope where process metrics (scanpath comparison) would have been beneficial.

The room clearing task depicted above is also relevant. Direct scanpath comparison is the natural, completely objective way to measure the similarity between subjects in this case.

To calculate the scanpath comparison metric, the tool *iComp* was developed to directly compare the scanpaths between multiple subjects suitable for expert/novice comparisons, e.g., in training tasks. The tool was designed to be easy to use by eye tracking researchers and as fully automatic as possible. As stated above, several previous eye tracking studies substantiate the inception of such a tool.

CHAPTER 2

BACKGROUND

The creation of a process metric based on eye movements relies on the development of an algorithm capable of measuring differences between two (or more) scanpaths. The goal is to evaluate the correlation between novice and expert inspectors' eye movements. This measurement should be capable of comparing fixated regions as well as the order of fixations.

Recent advances in the research of low-level visual processes have led to the emergence of sophisticated computational models of visual search. These models mimic the low-level portion of the human visual system, automatically detecting Regions of Interest (ROIs) as potential human gaze fixations (e.g., see [Itti et al. \[1998\]](#)). Concomitant with these efforts, techniques have begun to appear which compare human fixations to those generated algorithmically. These techniques are necessary in order to validate the similarity of an artificially generated scanpath to that of a human subject.

[Privitera and Stark \[2000\]](#) recently developed a methodology for comparing their artificially generated scanpaths to that of humans. The comparison algorithm relies on the clustering of gaze data into a limited number of ROIs and a subsequent step of assembling the temporal sequences of ROIs into ordered strings for sequence comparison based on *string editing*. Each clustered ROI is assigned a unique character label.

$s_1 = abcfeffgdc$ $s_2 = afbfddcdf$	start	cost 0
$s_1 = abcfeffgdc$ $s_2 = afeffddcdf$	after substitution of first b by e	cost 1
$s_1 = abcfeffgdc$ $s_2 = abcfeffddcdf$	after insertion of bc after first a	cost 2
$s_1 = abcfeffgdc$ $s_2 = abcfeffdc$	after deletion of last df	cost 2
$s_1 = abcfeffgdc$ $s_2 = abcfeffgdc$	after insertion of g	cost 1

Figure 2.1: Example of string editing. From [Privitera and Stark \[2000\]](#).

Two different indices of similarity are computed between two strings: S_p , the loci similarity index, and S_s , the sequence similarity index. The locational similarity, S_p , between two strings is the number of characters in both strings normalized by the number of *unique* characters in the second. Note that the second string

is assumed to be a “ground truth” against which the first string is compared [Claus et al. 2004] and thus the resultant metric is not symmetrical. To compute the sequential similarity, S_s , the Levenshtein distance algorithm [Levenshtein 1966], or string edit algorithm, (an optimization algorithm) assigns unit costs to three different character operations: *deletion*, *insertion*, and *substitution*. Characters are manipulated to transform one string to another, and character manipulation costs are tabulated. The cost is normalized by the length of the second string and complemented. Complementation is necessary because the normalized Levenshtein distance between two strings is zero based (zero meaning they are equal; one meaning no similarity) while instead a one based measure is desired in order to model percentage of similarity. To illustrate the calculation of both indices, take for instance two strings $s_1 = abcfeffgdc$ and $s_2 = afbffdcdf$. Since all the characters of s_2 are present in s_1 , the two strings yield a loci similarity index of $S_p = (5/5) = 1$. The Levenshtein distance (combined cost of deletions, insertions, and substitutions) between s_1 and s_2 is 6 (as illustrated in Figure 2.1). Given the length of s_2 is 9, the sequence similarity index between the two strings is $S_s = (1 - 6/9) = 0.33$. The results obtained from this example can be interpreted as meaning that the two strings have loci that are 100% within the foveal range and sequence that overlap by 33%.

Once calculated, similarity coefficients are stored in a table, named the *Y*-matrix, having one row and column for each image that a subject viewed (for all subjects). Because a complete *Y*-matrix would be far too large to be displayed, Parsing diagrams are utilized. Parsing diagrams contain averages of similarity coefficients collected from the *Y*-matrices and are a condensed and effective alternate way to display the data. Parsing diagrams consist of four main entries: Repetitive (*R*), the same viewer looking at the same scene at different times; Local (*L*), different viewers looking at the same scene; Idiosyncratic (*I*), the same viewer looking at different scenes; and Global (*G*), different viewers looking at different scenes. In addition, a Random (*Ra*) entry was provided, against which significance of the results could be concluded.

The string-editing methodology has 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 at best. 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.

Recently [West et al. \[2006\]](#) extended the string-editing scanpath comparison approach by augmenting it with the Needleman-Wunsch algorithm commonly used in bioinformatics. Their resultant analysis tool, *eyePatterns*, provides a measurement of sequence similarity that allows flexibility through variable scoring parameters. *eyePatterns* is freely available at <http://www.juliamae.com/eyepatterns/>.

[Privitera and Stark](#)’s approach to the comparison of scanpaths is one of the first methods to appear to quantitatively measure not only the loci of ROIs but also the order of ROIs. However, the technique suffers from a critical limitation in that it relies on k -means clustering of fixation points into ROIs, requiring *a priori* knowledge of the number of clusters, or regions in a scene. [Santella and DeCarlo \[2004\]](#) presented a viable alternative in their *mean shift* clustering technique. In general, clustering starts with a set of n points: $\{\mathbf{x}_i \mid i \in 1 \dots n\}$. The entire process is composed of two steps:

1. Shift the points into denser configurations until they can be easily separated.
2. Administer a distance threshold clustering algorithm.

The first stage—the mean shift—is crucial, as it ordains the robustness of the entire process. The process continues by repeatedly moving a point \mathbf{x}_i to a new location $\mathbf{s}(\mathbf{x}_i)$, the weighted mean of nearby points 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:

$$\mathbf{C} = \begin{pmatrix} E[x_i, x_i] & E[x_i, y_i] & E[x_i, t_i] \\ E[y_i, x_i] & E[y_i, y_i] & E[y_i, t_i] \\ E[t_i, x_i] & E[t_i, y_i] & E[t_i, t_i] \end{pmatrix} = \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_t^2 \end{pmatrix},$$

assuming independence among x_i , y_i , and t_i , with $E[x_i, y_i] = \frac{1}{n-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$. The following zero-

mean spatiotemporal Gaussian kernel can be used:

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

Generally, it is expected that $\sigma_x^2 > \sigma_y^2$, but for convenience the spatial distribution can be assumed to be symmetric. In this case, denoting $\sigma_s = \sigma_x = \sigma_y$, the spatiotemporal Gaussian kernel simplifies to:

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

where σ_s and σ_t determine local support of the Gaussian kernel in both spatial (dispersion) and temporal extent. For still images, the temporal dimension can simply be excluded by effectively setting σ_t to infinity.

[Santella and DeCarlo](#) show that, unlike with k -means clustering, the mean shift approach eliminates the need for *a priori* estimation of the number of ROIs in a given scene. The only existing user-adjustable parameter is σ_s , which can epistemically be set to match the extent of the foveal dimension of the human retina (about 5° visual angle).

CHAPTER 3

OPERATION

The goal of the project was to create a tool that is easily used by the target audience, maintainable, as fully automatic as possible, and implements the unique feature of direct quantitative scanpath comparison. The first two goals precluded the continuation of the former project (described later), and warranted the porting of the code to C++ to improve the ease of maintenance and modification. In addition, the use of the open source GUI development library Qt allowed for the development of an effective user interface. Automation is highly desired because without it, either the tedious process of defining ROIs by hand for every image must be performed (in the worst case) or the number of expected ROIs must be defined *a priori* (in the case of *k*-means). Either way this limitation is unacceptable when many images or (even worse) video needs to be analyzed. It is precisely because of that reason that [Privitera and Stark](#)'s research had to be extended and [West et al.](#)'s work fell short of our goals.

This implementation sprang out of a previous semester long project. The former project clustered (using [Santella and DeCarlo](#)'s method) input data and produced repetitive values for each subject and local values for each image. The program was written using C and OpenGL, but lent itself more to object oriented design because the complexity and intertwining of the involved data structures. There was insufficient time to implement computation of all the Parsing diagram's values and provide data for statistical analysis. We chose to continue the project because of its considerable significance to the eye tracking community and potential to anyone performing training related to a visual task. It is worth noting that the data from a pilot study conducted during the former project was used (described later on) to validate this project, explaining why the design of the experiment was specifically around the local and repetitive measures. This fact does not present a major problem because local values represent the similarity of multiple viewers performing the same visual task, and therefore are precisely the metric needed for conducting expert/novice studies or training.

Although the clustering algorithm can in principle operate on raw data, it is generally more meaningful to compare fixation points. Fixations can be classified from raw eye movement data by several different approaches, e.g., position-variance, hidden Markov models, or velocity filtering. iComp uses the latter to classify fixations. Input to the tool assumes a sequence of eye gaze data (x, y, t) augmented with state information defining the current viewer and current image the data belongs to. Point-by-point velocity-based saccade detection is performed to filter out (on the order of 10%) saccadic data points. Velocity is defined as

difference in position over difference in time.

$$\mathbf{v}_t = \frac{\sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2}}{dt}$$

Simple thresholding is used to classify points that are related to saccadic eye movement and hence not fixations.

Clustering

Because all data for each frame should be clustered together the spatial version of the mean shift kernel is sufficient, i.e., the spatial Gaussian kernel is used:

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

Kernel support is limited to $2\sigma_s$, in order to eliminate the effects of distant outliers. The implementation keeps two lists: one containing the original data (for each image) and the other containing the mean shifted data. The mean shifted list initially contains a copy of the original data but is repeatedly relocated ($\mathbf{s}(\mathbf{x}_i)$ is calculated) until convergence. Convergence is detected by the condition that no data points move more than ϵ pixels in a single mean shift step. The ϵ value is typically set to a very small integer which ensures sufficient convergence so a simple distance clustering algorithm can easily differentiate clusters. Clusters smaller than a user defined value can be considered outliers and discarded.

Comparing

The comparison portion of the implementation followed [Privitera and Stark's](#) method closely. Each cluster is labeled with a unique character. The cluster ordering is defined by the first subject's gaze data and kept consistent for all other subjects. Strings for each subject (for each image) are then constructed by concatenating the character, from each cluster visited by the subject's gaze data, to the end of the string. Duplicate adjacent characters are eliminated if an attempt is made to concatenate them to the string. The Levenshtein distance is computed by an optimization algorithm that builds an $n \times m$ array (where n and m are the lengths of the two involved strings) and finds the minimum cost to transform one string into the other. The array is

constructed by computing:

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

$$\text{array}[i][j] = \min(\text{array}[i-1][j] + 1, \text{array}[i][j-1] + 1, \text{array}[i-1][j-1] + c(i, j))$$

where the first and second terms in the minimization statement handle the costs of *deletions* and *insertions*, and the last term handles *substitutions*. The first row and column must be initialized with ascending integers ($[0..m]$ and $[0..n]$) as a pre-processing step. For example, given two strings $s_1 = abcfeffgdc$ and $s_2 = afbffdcdf$, the 10×9 array that would be generated is illustrated in Figure 3.1. The cost to completely

	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 3.1: Example of Levenshtein distance calculation.

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.

Visual Data Representation

iComp displays a visualization of the eye gaze data of all subjects for each image (see Figure 3.2 and 3.3). The subject's eye movement data is color coded to match the legend at the top right corner of the display. A bright blue ellipse surrounds each cluster, and a bright green character towards the center of the cluster is its label.

The user is able to set the velocity at which to classify points as saccades (and therefore discard them), the σ_s mean shift area of effect, and the minimum number of points that a cluster may contain without being classified as an outlier.

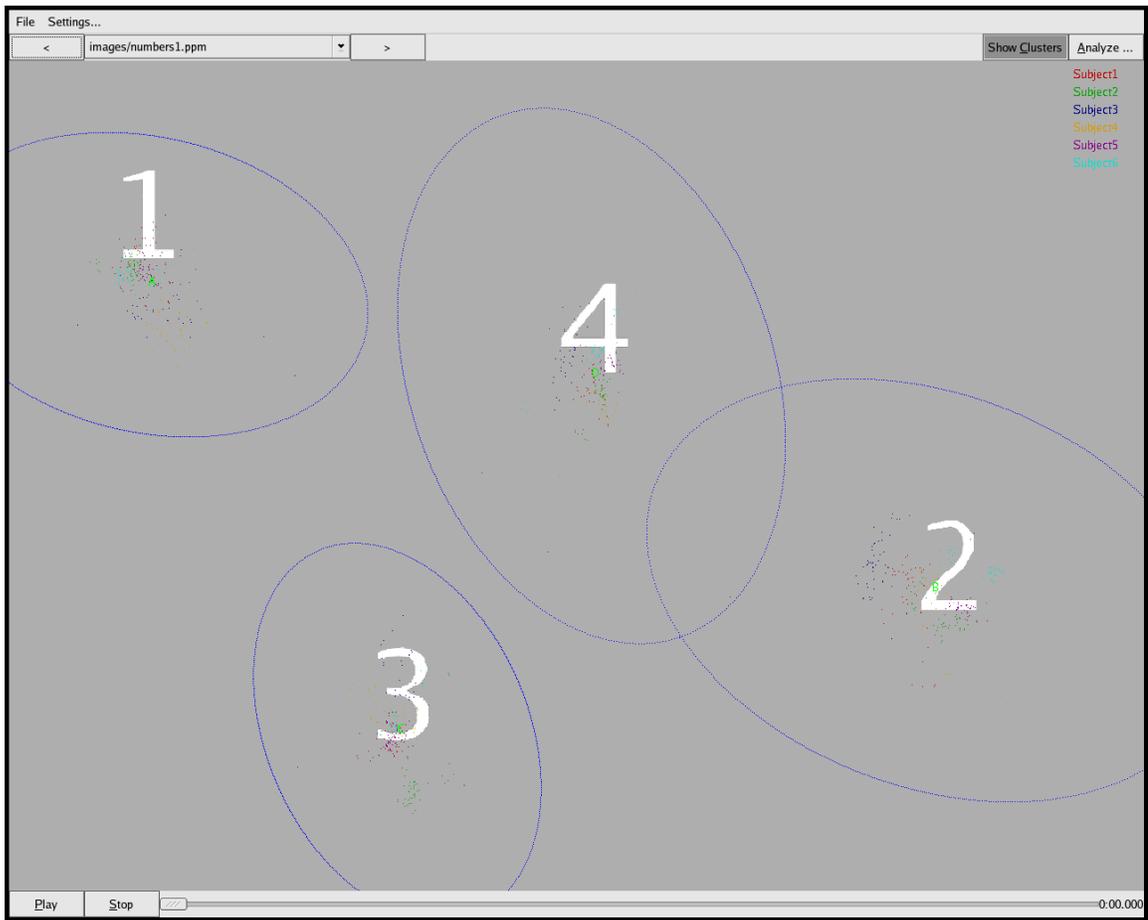


Figure 3.2: Sample data visualization (numerical image)

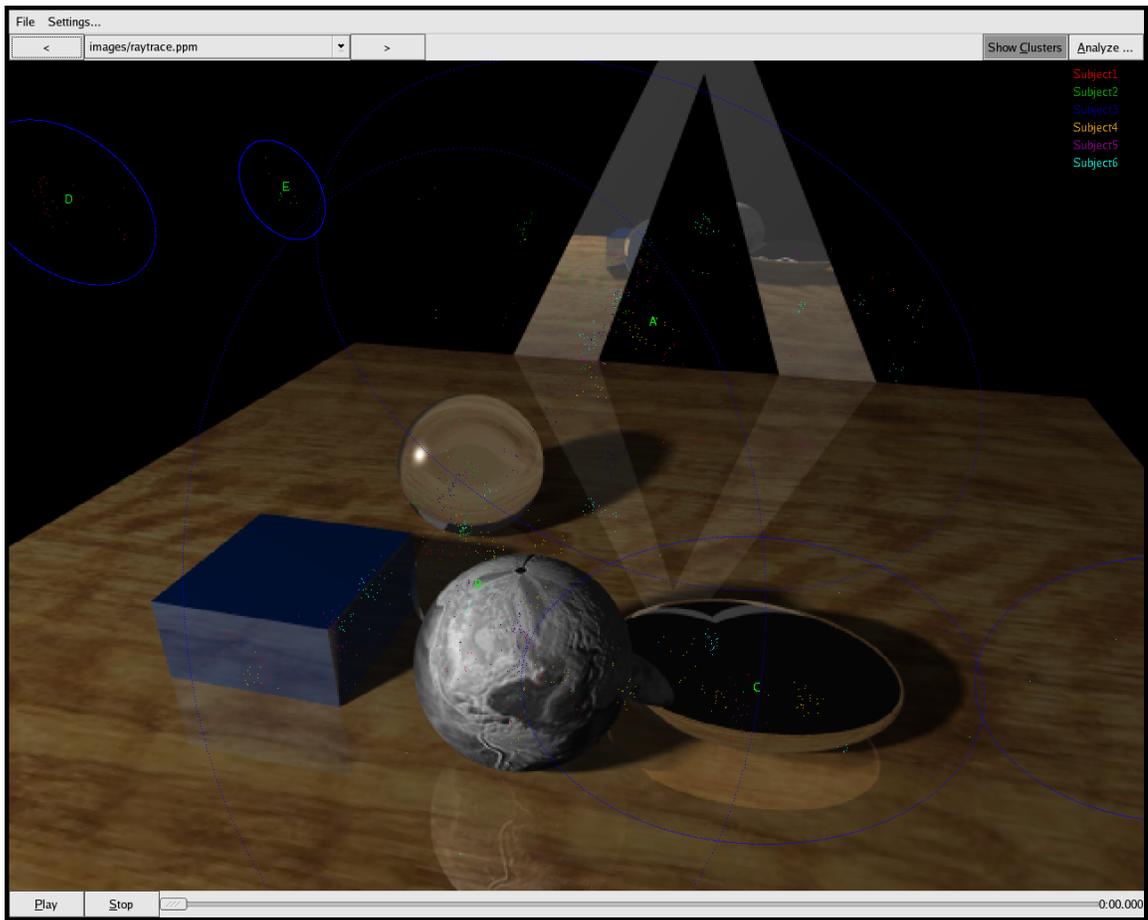


Figure 3.3: Sample data visualization (CG image)

CHAPTER 4

EXPERIMENTAL VALIDATION

The goal of the experiment was to verify the correctness of the comparison algorithm. It was hypothesized that the local, L , value between subjects instructed to view an intuitive scene and given a task should be significantly higher than the random, R_a , value for both loci, S_p , and sequence, S_s . On the contrary, subjects instructed to view a counter-intuitive scene and not given any specific task should produce L values much lower and closer to the R_a value.

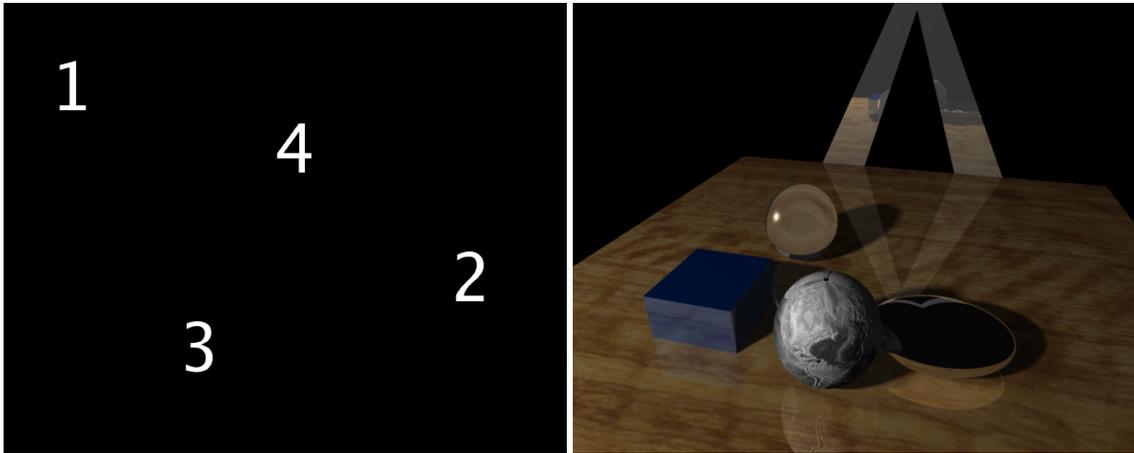


Figure 4.1: Sample (aggregated) numerical and CG stimulus.

Apparatus

The experiment was performed with a Tobii 1750 eye tracker. The Tobii 1750 is a 17 inch flat screen monitor with an incorporated eye tracker. The resolution of the monitor is 1280×1024 . The eye tracker is capable of binocular tracking at a 50Hz sampling rate within 0.5 degrees of accuracy. A PC with dual AMD Opteron 64 processors running Microsoft Windows XP and software provided with the Tobii gathered the eye gaze data and exported it via TCP/IP. The display and data collection program was run on a PC with an AMD Opteron processor running Fedora Core Linux. The Linux box used TCP/IP to collect the data from the Windows box. The display and data collection application was developed with C++ and OpenGL.

Experimental Design

Subjects consisted of six college students (all male). Ages of the participants ranged from 21 to 42 years old. The number of subjects was selected relative to past studies and availability. Subjects were screened based on their ability to calibrate well with the eye tracker.

The visual stimuli consisted of three blank, black screens with randomly placed numbers 1 through 4 (Figure 4.1) and a more complex computer generated (CG) image produced by a ray tracer (Figure 4.1). While displaying the first three numerical images each number was flashed onto the screen one at a time. The individual numbers were displayed for 500ms to allow for an initial orientation or seeking time and one long fixation. Fixation durations generally range from 150-500ms [Duchowski 2003]. The first stimulus (all four numbers) was repeated in order to obtain repetitive results. The last (CG) stimulus was displayed for 5 seconds to give subjects additional time to view the feature-rich image.

Procedures

The subjects were placed directly in front of the Tobii at approximately 60cm distance. Calibration was performed by displaying nine blue circles evenly spaced throughout the display. The circles were displayed independently and shrank down from a diameter of 30 pixels to a diameter of 2 pixels. The eye tracker collected 22 calibration samples at each circle location. The calibration accuracy was stored after each collection. Average precision variance was computed to ensure each calibration was within acceptable limits.

Before each trial began the test subjects were instructed to fixate on each number as it appeared and to freely inspect the last (CG) image. They were informed of how much time they would have to look at each number and the final image. For each run all the eye gaze data (x , y , and t) was collected. The data was exported along with headers indicating the current viewer and image to a log file to be analyzed at a later time. After the test another calibration was performed and the average precision variance was again collected. The variances from the two calibrations were compared in order to factor out any slippage of the eye tracker.

Results

Any velocity above $130^\circ/\text{sec}$ [Duchowski 2003] classifies data as saccadic. Given a viewer 60cm away from a 17 inch (diagonal) display, and a resolution of 1280×1024 pixels, the velocity at which to consider data saccadic translates to approximately 12,527 pixels/sec and was set accordingly. The parafoveal range

that can be seen by humans in high detail is approximately 5° away from the center of vision in any direction. Given the above dimensions of the display, the parafoveal radius translates to approximately 100 pixels. Therefore data was analyzed with $\sigma_s = 100$ to cluster in respect to what a viewer can foveate on. Clusters with less than ten points were treated as outliers and discarded.

	Same Subj.	Diff. Subj.	Same Subj.	Diff. Subj.
Same Image (SI)	Repetitive 1.0	Local 1.0 $F(1,6) = 176.0, p < 0.01$ $\chi^2 = 6.14, p = 0.01$	Repetitive 0.59	Local 0.91 $F(1,6) = 325.0, p < 0.01$ $\chi^2 = 5.40, p = 0.02$
Diff. Image (DI)	Idiosyncratic 0.94	Global 0.95	Idiosyncratic 0.63	Global 0.63
	S_p	Random 0.51	S_s	Random 0.03

Figure 4.2: Parsing diagram, numerical images, $\sigma_s = 100$

	Same Subj.	Diff. Subj.	Same Subj.	Diff. Subj.
Same Image (SI)	Repetitive 0	Local 0.81	Repetitive 0	Local 0.45
Diff. Image (DI)	Idiosyncratic 0	Global 0	Idiosyncratic 0	Global 0
	S_p	Random 0.47	S_s	Random 0.04

Figure 4.3: Parsing diagram, CG image, $\sigma_s = 100$

The results for the numerical images presented in Figure 4.2 demonstrate that all subjects looked at all clusters. As anticipated, the local value was significantly high expressing the strong similarity of scanpaths of different viewers on the same images. The results for the CG image is presented in Figure 4.3. The repetitive, idiosyncratic, and global values are zero because only one image was viewed by multiple subjects. The local values are lower for both loci and sequence, as expected.

Since σ_s was selected based on human physiology it is possible that the data could have been skewed (too large/not enough clusters) relative to the image features and size. To examine further, the data was run with $\sigma_s = 70$ and $\sigma_s = 40$, with results exhibited in the Figures 4.4, 4.5, 4.6, and 4.7.

The first value below the L value for each of the numerical images contains results from ANOVA. Statistically we have two treatments: the similarities between different subjects viewing the same image, and the similarities between subjects and the random algorithm viewing the same image. ANOVA is applied to determine whether the means L, Ra, vary significantly, indicating that they belong to different populations.

	Same Subj.	Diff. Subj.	Same Subj.	Diff. Subj.
Same Image (SI)	Repetitive 1.0	Local 0.99 $F(1,6) = 579.9, p < 0.01$ $\chi^2 = 5.6, p = 0.02$	Repetitive 0.59	Local 0.88 $F(1,6) = 258.5, p < 0.01$ $\chi^2 = 5.33, p = 0.02$
Diff. Image (DI)	Idiosyncratic 0.85	Global 0.87	Idiosyncratic 0.52	Global 0.54
	S_p	Random 0.29	S_s	Random 0.03

Figure 4.4: Parsing diagram, numerical images, $\sigma_s = 70$

	Same Subj.	Diff. Subj.	Same Subj.	Diff. Subj.
Same Image (SI)	Repetitive 0	Local 0.78	Repetitive 0	Local 0.23
Diff. Image (DI)	Idiosyncratic 0	Global 0	Idiosyncratic 0	Global 0
	S_p	Random 0.35	S_s	Random 0.07

Figure 4.5: Parsing diagram, CG image, $\sigma_s = 70$

The second statistical value is the output produced by the Kruskal-Wallis rank sum test. Because the sample population was limited, it may be presumptuous to assume a normal distribution. The Kruskal-Wallis test is used to gauge statistical significance without the assumption of normality of their distributions.

Discussion

The results were generally as expected. For the numerical images, both the S_p and S_s L values were significantly higher than the random value. The I and G values were high and conjectured so because the numerical visual task was strictly directed. Even without a provided visual task, it was expected and confirmed that the S_p values for the CG image should be somewhat high (approximately 0.50 or above [Privitera and Stark 2000]). Even so the L values for the CG image were much lower than those for the numerical images. Unfortunately, insufficient amounts of data were collected to perform significance tests on the CG image measures. As anticipated, decreasing the σ_s value had the effect of lowering all of the measures. The numerical image values still retained significance and remained considerably higher than the CG image values.

The L values for all but the first (numerical) image suffered because data was collected uninterruptedly throughout each entire trial. Because of this fact, subjects did not have sufficient time to reorient their eyes, and data from the previous image's last fixation was carried over to the next image.

	Same Subj.	Diff. Subj.	Same Subj.	Diff. Subj.
Same Image (SI)	Repetitive 0.96	Local 0.95 $F(1,6) = 397.0, p < 0.01$ $\chi^2 = 5.33, p = 0.02$	Repetitive 0.77	Local 0.80 $F(1,6) = 71.1, p < 0.01$ $\chi^2 = 5.33, p = 0.02$
Diff. Image (DI)	Idiosyncratic 0.83	Global 0.86	Idiosyncratic 0.49	Global 0.52
	S_p	Random 0.30	S_s	Random 0.06

Figure 4.6: Parsing diagram, numerical images, $\sigma_s = 40$

	Same Subj.	Diff. Subj.	Same Subj.	Diff. Subj.
Same Image (SI)	Repetitive 0	Local 0.67	Repetitive 0	Local 0.20
Diff. Image (DI)	Idiosyncratic 0	Global 0	Idiosyncratic 0	Global 0
	S_p	Random 0.32	S_s	Random 0.08

Figure 4.7: Parsing diagram, CG image, $\sigma_s = 40$

CHAPTER 5

CONCLUSION

The experiment performed validated the results as expected, verifying the correctness of the combination of previous methods. The results suffered somewhat because of the limited amount of data used for validation. Even so the test data used was catered around the expectation that iComp will be used for expert/novice studies and, therefore, verified the relevant portion of the tool. In general, iComp can be used in the future for eye tracking studies where direct scanpath comparison is desired to provide a solid process metric.

Future work will include the implementation of controls to limit data temporally to counteract the problem of fixations carrying over between images (as experienced in the experimental validation of iComp). Controls to replay the data in real time will be employed. Various visual reports (tool-tips displaying important information about viewers, images, and clusters) will be added. The source distribution is available at <http://www.e-t-r-a.org/iComp/>.

BIBLIOGRAPHY

- 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.
- CLAUSS, M., BAYERL, P., AND NEUMANN, H. 2004. Evaluation of regions-of-interest based attention algorithms using a probabilistic measure. In *5th Workshop Dynamische Perzeption*. Tübingen, 227–232. URL: <http://neuro.informatik.uni-ulm.de/basilic/Publications/2004/CBN04> .
- DUCHOWSKI, A. T. 2003. *Eye tracking methodology: theory and practice*. Springer-Verlag, London, England.
- 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.
- 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.
- 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.
- LAW, B., ATKINS, M. S., KIRKPATRICK, A. E., LOMAX, A. J., AND MACKENZIE, C. L. 2004. Eye Gaze Patterns Differentiate Novice and Experts in a Virtual Laparoscopic Surgery Training Environment. In *Eye Tracking Research & Applications Symposium*. ACM, 41–47.
- LEVENSHTEIN, V. I. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Doklady Physics* 10, 707–710.
- MEGAW, E. D. AND RICHARDSON, J. 1979. Eye movements and industrial inspection. *Applied Ergonomics* 10, 3, 145–154.
- 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.
- SADASIVAN, S., GREENSTEIN, J. S., GRAMOPADHYE, A. K., AND DUCHOWSKI, A. T. 2005. Use of Eye Movements as Feedforward Training for a Synthetic Aircraft Inspection Task. In *Proceedings of ACM CHI 2005 Conference on Human Factors in Computing Systems*. ACM Press, Portland, OR, 141–149.
- 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.
- SCHOONARD, J. W., GOULD, J. D., AND MILLER, L. A. 1973. Studies of Visual Inspection. *Ergonomics* 16, 4, 365–379.
- WANG, M.-J. J., LIN, S.-C., AND DRURY, C. G. 1997. Training for strategy in visual search. *Industrial Ergonomics* 20, 101–108.
- 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.