eyePatterns: Software for Identifying Patterns and Similarities Across Fixation Sequences

Julia M. West, Anne R. Haake, Evelyn P. Rozanski
Rochester Institute of Technology
Information Technology Department
Rochester, NY 14623 USA
{jmw9523, Haake, Rozanski} @it.rit.edu

Keith S. Karn
Xerox Corporation
Industrial Design/Human Interface Department
Rochester, NY 14623 USA
keith.karn@xerox.com

Abstract
Fixation sequence analysis can reveal the cognitive strategies that drive eye movements. Unfortunately this type of analysis is not as common as other popular eye movement measures, such as fixation duration and trace length, because the proper tools for fixation sequence analysis are not incorporated into most popular eye movement software. This paper describes eyePatterns, a new tool for discovering similarities in fixation sequences and identifying the experimental variables that may influence their characteristics.

Keywords: Eye tracking, data analysis, sequence analysis

Categories: G.3 Probability and Statistics---Statistical software; I.5 Pattern Recognition (I.5.2 Design Methodology, I.5.3 Clustering)

1. Introduction
The analysis of eye gaze has been used to determine the location of important and interesting information within a stimulus [e.g., Mackworth and Morandi 1967; Loftus and Mackworth 1978] and to identify cognitive approaches to task completion [e.g., Just and Carpenter 1976; Hornof and Halverson 2003]. Furthermore, fixation sequences, or the succession of fixations on a stimulus, and the cognitive processes that drive them have also been investigated. The analysis of fixation sequences, which are sometimes referred to as scanpaths, has many practical applications, and is becoming increasingly popular in eye movement research.

Early research on fixation sequences focused on the existence of repetition within sequences, and the relationship of that repetition to underlying cognitive strategy. Yarbus [1967] determined that the order of fixations on regions of a stimulus is influenced by the relative importance of the regions to the viewer, and that viewers exhibited repeated cycles, or patterns, of fixations on the most interesting features of a stimulus. Noton and Stark [1971] expanded on this by examining viewers’ sequences of fixations upon first viewing and then reviewing a stimulus. They found that a subject’s gaze tended to follow a similar sequence of fixations, or a scanpath, whenever a scene was viewed repeatedly, indicating that ocular motor control may be part of the memory of the scene.

Most of the past sequence analysis applied to eye tracking data has dealt with subjects passively viewing a visual stimulus. Yarbus [1967] showed that subtle changes to the viewing instructions influenced fixation data during picture viewing. Sequence analysis of eye tracking data may be best suited to manual manipulation tasks in the real world that may lead to more consistency in eye movement patterns [e.g., Ballard, et al. 1997 and Hayhoe 2000]. Additional studies have indicated that cognitive processes, dependent on subject, stimulus, and task, may drive fixation sequences [Stark and Ellis 1981; Brandt and Stark 1997].

Fixation sequence analysis has also been used to evaluate algorithms for automated detection of areas of interest in a visual stimulus. Privitera and Stark [2000] compared sequences of these machine identified AOIs with sequences of fixations made by human subjects.

Figure 1. An example of a fixation sequence on a web page. Each area of interest (AOI) is labeled with a one-character code. The string representation of the fixation sequence is a concatenation of the AOI codes, in the order in which fixations occurred within the AOIs. The string representation of this sequence is ‘TTBNNLBL.’
As gaze sequences may reflect underlying cognitive operations, the differences and similarities between gaze sequences may reflect the differences and similarities between cognitive processes while viewing a scene. Brandt and Stark [1997] compared sequences by using a string-edit algorithm, which can determine the minimum number of operations necessary to convert one string of characters into another and thus their overall similarity. To apply the string-edit algorithm to sequences of fixations, the stimulus is divided into regions, or areas of interest (AOIs), and each region is typically identified by a character. Each fixation in a sequence is represented in a string as the character code for the AOI in which each fixation falls (Figure 1). Josephson and Holmes [2002] also implemented a string-edit algorithm, optimal matching analysis (OMA) [Holmes 1997], to determine the similarity of subjects’ scanpaths while viewing websites. They visualized, with UCINET [Borgatti et al. 1992], an application used to analyze social networks, the relationships between the sequences in two-dimensional graphs.

Another method that has been used to compare gaze sequences is first-order statistical dependency. Senders [1964] investigated subjects’ distribution of attention among many information sources and used, among other data, the probability of subject’s gaze transitioning from one AOI to another to predict the manner in which attention switches between information sources. Ellis and Smith [1985] discussed the degree of randomness that is found in fixation sequences, and introduced the idea that fixation sequences may be generated by a random, stratified random, or statistically dependent Markov process. Ellis and Stark [1986] further investigated the statistical dependency of fixation sequences by looking at the frequencies of transitions between each AOI in a stimulus. Ellis and Stark performed a chi-square goodness-of-fit test of transition frequencies found in subjects’ fixation sequences against those that would result from a stratified random sampling of the data, and determined that the frequency of a transition to a given AOI is statistically dependent upon the previous AOI. Ellis’ application of Chi-square analysis can be expanded to determine whether a statistically significant difference between the transitions in two sequences or groups of sequences exists.

The effectiveness of string-edit algorithms and transition frequency analysis for comparing gaze sequences should be cause for eye movement researchers to investigate the order of fixations more often than is presently done. Unfortunately, these methods are too computationally complex to be readily implemented, and the available tools, which were not created with eye movement researchers in mind, are difficult to use.

This paper introduces eyePatterns, new software that uses well
-established sequence analysis algorithms [Levenshtein 1966; Needleman and Wunsch 1970; Smith and Waterman 1985], designed specifically to aid eye movement researchers in comparing sequence patterns within and across experimental groups of subjects. Comparison of sequence patterns affords insight into questions regarding how subjects examine a scene while performing a task of interest to the researcher.

Before designing this software, we looked at the current applications and analysis tools that eye movement researchers are using to analyze fixation sequences. Our goal was to integrate many of the analyses that researchers are currently using (e.g., string-editing, transition frequency analysis, clustering) with analyses that are not usually applied to eye movement research (e.g., sequence alignment, pattern discovery) and to provide an intuitive interface and meaningful visualizations to create a highly usable, yet powerful sequence analysis tool.

2. The eyePatterns Interface

eyePatterns' graphical user interface (GUI) is designed so that users can see all information important to their tasks on one screen (Figure 2). There are several views displayed in the interface. The Sequences panel displays information relevant to all sequences, including any experimental variables. Displayed sequences can be sorted and selected, and then analyzed. The Source panel shows labels that have been assigned to sequences, and provides one-click access to display the sequences under each category. The Output panel displays the text output for all analyses. In addition, when a visualization of sequence relationships is displayed, a separate window appears, showing the visualization and controls for that visualization.

3. Gaze Sequences

eyePatterns performs sequence analysis on strings that represent fixation sequences. Each fixation in the string is defined by the area of interest (AOI) in which it is located. A fixation sequence that contains multiple successive fixations within an AOI is called an “expanded” sequence in eyePatterns. Successive fixations within a single AOI can be collapsed into one representation within the sequence (i.e., a “gaze” or “dwell” on the AOI). When collapsed in this way, we refer to the resulting sequence as a “collapsed” or “gaze” sequence. For example, the fixation sequence illustrated in Figure 1, “TTBNLNBL”, collapses into “TBNLBL..” Collapsed sequences highlight only the order in which a subject looks at AOsIs. Expanded sequences, though, convey both the sequential characteristics of a sequence and some temporal characteristics. The duration of individual fixations, however, is not encoded, but can be approximated by the number of successive fixations within an AOI.

Investigations of eye movement behavior generally involve two or more experimental conditions. eyePatterns allows sequences to be described with any number of experimental variables, which are customizable to the user's needs. For example, it may be convenient to label a sequence by subject ID and task or trial number, as well as by experiment-specific variables. A researcher studying human-computer interfaces might include computer experience, task success, or even preference data. Allowing experimental variables to describe the sequences makes it easy for the user to look at a subset of sequences and identify distinguishing patterns across experimental variables. It also establishes a meaningful method for identifying sequences other than an identification number.

Although these sequence-describing variables provide a means to select a group of sequences to view or analyze, users can establish a shortcut by assigning labels to sequences so as to provide quicker access to groups of sequences. For example, a user can label all sequences whose variable “task number” is 1 as “Task 1”. A new label group appears in the source panel, enabling the user to show all the sequences that have that label or to perform an analysis on those sequences

4. Sequence Similarity

When working with eye tracking data, it is common to determine some measurement of similarity between many subjects' fixation or gaze sequences. Often, similarity is determined by using simple visual analysis. This analysis is prone to error and bias, whereas computational analysis of sequence similarity generates concrete and reproducible results. Unfortunately, available tools that perform sequence similarity analysis, such as string-edit and transition frequency analysis, are challenging and time consuming to learn. They also do not offer data mining tools that can produce results that are easy to interpret. All of these factors were taken into account when building sequence similarity analysis into eyePatterns.

Currently, eyePatterns allows users to choose between two algorithms to determine mathematical similarity between sequences. The first is the Levenshtein distance algorithm [Levenshtein 1966] also known as the string-edit algorithm, which was included because of its simplicity and growing use within eye movement research. The second is the Needleman-Wunsch [Needleman and Wunsch 1970] algorithm that, unlike the basic string-edit algorithm, allows the user to specify scoring parameters that are optimal for a given experiment.

While string-edit distance is an efficient measurement of dissimilarity between two sequences, its most basic implementation provides no flexibility in how distance scores are calculated. Needleman-Wunsch is a common algorithm used in bioinformatics to align genetic sequences along their entire lengths. This global alignment provides a measurement of sequence similarity in a manner that allows flexibility through variable scoring parameters. In some implementations of this algorithm, user-defined parameters such as gap penalties and similarity matrices, determine which sequence units will optimally be paired in the alignment. A similarity matrix determines the score for each mismatch between a pair of characters in the alignment. Deriving a similarity measurement from flexible parameters would be advantageous in situations in which two different AOsIs have a close spatial proximity or serve a similar function. In these situations, we might expect the two AOsIs to be interchangeable in some fixation sequences.

eyePatterns uses the NeoBio library [Anibal 2003] implementation of the Needleman-Wunsch algorithm to determine a similarity score.

4.1 Visualizing Sequence Similarity

Typically, eye movement studies generate many individual fixation sequences. If all sequences are compared in a pair-wise manner, as with the Needleman-Wunsch algorithm, the size of the resulting dataset presents a difficult situation in terms of data interpretation. Graphing the relationships between each pair of sequences can highlight groups, or clusters, of sequences that are mathematically similar. Coupling similarity visualizations with contextual information about the sequences allows researchers to extract prominent trends in how experimental variables might affect eye movement sequences. eyePatterns integrates graphing
into its similarity analyses so that researchers can produce expository visualizations of the relationships between sequence similarity and experimental variables. The tool uses the prefuse library [Heer et al. 2005] for displaying visualizations.

By default, eyePatterns will output a matrix depicting the degree of similarity between each pair of sequences that has been compared. To help users determine which sequences are most alike, eyePatterns can perform unsupervised learning, or form natural groupings of sequences. Users have the option to visualize the output of these algorithms that will display the sequences, placing the most similar sequences closest together.

One method included in eyePatterns is hierarchical clustering, a popular data analysis method. Clustering partitions data into subsets (i.e., clusters) of items that share similar traits. For example, the common trait among gaze sequences that are clustered together is their proximity to one another i.e., a small distance score. Agglomerative hierarchical clustering builds a hierarchy of clusters, beginning with the two most closely related sequences, and ending with the most distant sequence or cluster. The hierarchy tree can be visualized (Figure 3), exposing outlying and the most similar sequences closest together.

A second method that helps the user visualize relationships between sequences is multidimensional scaling (MDS) [Kruskal and Wish 1978], a statistical algorithm that assigns low-dimensional coordinates to items so that the distance between each item is preserved and can be more easily visualized. Sequences are first given random coordinates in two-dimensional space and the distances between the sequences’ coordinates are calculated. These values are then compared to the actual distance values and adjustments are made to the coordinates. A cycle of comparing the distances and adjusting the coordinate assignments continues until no improvement can be made. The result is a pairing of the sequences to coordinates in a two-dimensional space that can be visualized. When graphed, the visual distance between items should be proportional to the calculated distance measurement between them (Figure 4).

In order to clarify relationships between sequences, eyePatterns employs a customizable color-coding system. Sequences, appearing as nodes on a graph, are colored according to their values for a particular variable. This ideally can highlight the experimental variables that may be affecting visual attention and the resulting eye movement sequences. The user can easily toggle the variable that determines the color of the sequence nodes so that the relationship between experimental variables and sequence similarity may be examined.

The sequence relationship graphs are also highly interactive. Users can re-center a visualization by clicking on a sequence node, drag the mouse to select multiple sequences, and right click on a sequence node to perform operations on that node.

5. Pattern Finding

It is possible that two gaze sequences may be considered by a global similarity metric to be dissimilar and yet actually have important undetected local similarities. This may occur because the sequences are of different lengths or may contain the same subsequences, or cycles, which appear at different locations for each sequence. For example, if a dozen participants have been eye tracked while using a webpage, their gaze sequences during a given task might have widely differing lengths and overall composition. It is possible, though, that the majority of the participants similarly may have looked back and forth between two elements of the page. Measuring overall similarity between all of the sequences would not have revealed this. In situations like this, identifying eye movement patterns may be the best method for finding similarities between sequences. In eyePatterns, patterns are defined as subsequences that are found more than once in one or more sequences. Patterns might reveal elements in

Figure 3. A visualization of a hierarchical tree displaying clusters of similar sequences. Each colored node represents one sequence. The nodes are color-coded by the value of a sequence variable, chosen by the user.

Figure 4. The calculated string-edit distances between twenty-two sequences, shown in a two-dimensional visualization.
eyePatterns provides users with several methods for discovering patterns in gaze sequences. One method is the calculation of transition frequencies, which can be determined for one or more sequences. Transition frequencies are output as a matrix, in which each entry is color-coded on a scale that shows the highest frequencies in red and the lowest frequencies in black. In addition, eyePatterns can be used in eyePatterns to verify that the transitions within one or many sequences are statistically different from those observed in another group of sequences, or from the transitions found in a randomly generated sequence.

Locally aligning sequences [Smith and Waterman 1985], is another method, widely used in bioinformatics for finding patterns in data. A local alignment between two sequences highlights closely related subsequences that are found in both sequences by only aligning the regions that will produce the largest similarity score. The subsequences are not necessarily located in the same position in both sequences, but nonetheless are identified as being highly similar (Figure 5). This method, new to the eye movement field, enables valuable pattern discoveries to be made. For example, performing a local alignment on two eye movement sequences could identify visual search strategies shared by two individuals although they occur at different times within an experiment.

With eyePatterns, unknown and specified patterns can be found through discovery and pattern matching, respectively. The discovery method allows users to input length and content criteria, and returns all patterns matching those criteria. For example, a user can elect to discover only patterns that are at least five characters long, which contain the AOI labeled “B,” and are found in at least four sequences. Any instance of a substring that meets those criteria will be returned. Pattern matching is included for users who already have an idea of the patterns for which they are looking. Users can search for an exact pattern, or provide a regular expression—a string that uses certain syntax rules to describe a set of strings. For users that are not familiar with regular expression syntax, eyePatterns lets users match patterns using criteria that can be entered through a form.

6. Conclusion

eyePatterns makes analyzing fixation sequences easier than previously possible. This software tool provides trusted sequence analysis techniques and an intuitive interface that guides the user from identifying which experimental variables may influence fixation sequences to discovering patterns shared by similar sequences. The software currently can import data in several formats, but more software integration is planned for future development, including the possibility of capturing fixation sequence data online during eye movement experiments. Additional sequence analysis techniques, such as compression and filtering of sequences are also planned.

Acknowledgements

We wish to thank Chad Lapa and Christina Sekella for their contributions and tireless support.

References


