

Identifying Usability Issues via Algorithmic Detection of Excessive Visual Search

Corey Holland

Department of Computer Science
Texas State University
San Marcos, TX 78666 USA
ch1570@txstate.edu

Oleg Komogortsev

Department of Computer Science
Texas State University
San Marcos, TX 78666 USA
ok11@txstate.edu

Dan Tamir

Department of Computer Science
Texas State University
San Marcos, TX 78666 USA
dt19@txstate.edu

ABSTRACT

Automated detection of excessive visual search (ES) experienced by a user during software use presents the potential for substantial improvement in the efficiency of supervised usability analysis. This paper presents an objective evaluation of several methods for the automated segmentation and classification of ES intervals from an eye movement recording, a technique that can be utilized to aid in the identification of usability problems during software usability testing. Techniques considered for automated segmentation of the eye movement recording into unique intervals include mouse/keyboard events and eye movement scanpaths. ES is identified by a number of eye movement metrics, including: fixation count, saccade amplitude, convex hull area, scanpath inflections, scanpath length, and scanpath duration. The ES intervals identified by each algorithm are compared to those produced by manual classification to verify the accuracy, precision, and performance of each algorithm. The results indicate that automated classification can be successfully employed to substantially reduce the amount of recorded data reviewed by HCI experts during usability testing, with relatively little loss in accuracy.

Author Keywords

Human-computer interaction; eye tracking; visual search; usability; usability testing.

ACM Classification Keywords

H5.2. User Interfaces: Evaluation/methodology; H5.2. User Interfaces: User-centered design.

General Terms

Algorithms; Experimentation; Human Factors; Theory; Performance.

INTRODUCTION

Usability refers to the ease with which users can make use of a system for its intended purpose, and is arguably the most important quality of any system meant for human

interaction. The ISO 9241 standard defines usability as “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” [1].

Unfortunately, usability testing is often an expensive and time consuming process, requiring careful manual review and analysis of users’ interaction with applications [2]. As a result, despite the integral nature of usability to the success of an application, usability testing is often neglected as part of the development process [3].

The primary shortcoming of usability testing is its qualitative nature, as described by several usability practices and guidelines [4]. By identifying and standardizing usability metrics, usability may be evaluated quantitatively, a process which lends itself to automation. In this way, usability testing itself can be made more usable, by reducing the time and effort spent evaluating an interface.

In this paper, we consider the efficacy of eye movements as an indicator of software usability. Specifically, we explore a number of techniques for the automated segmentation and classification of usability recordings of eye movement data. Through quantitative analysis of basic eye movements (fixations and saccades) and the patterns they produce (scanpaths), we attempt to accurately and precisely locate time intervals in which the user experiences difficulty with a software interface.

There are a number of eye movement types identified by varying characteristics; of these, however, fixations and saccades are of particular importance to the field of human-computer interaction [5]. Fixations occur when the eye is held in a relatively stable position such that the fovea remains centered on an object of interest, providing heightened visual acuity. Saccades occur when the eye globe rotates quickly between points of fixation, with very little visual acuity maintained during rotation [6].

Various sources have described the usability implications of eye movements [7]; however, usability evaluation based on eye movements generally makes use of only scanpath (a sequence of fixations and saccades) and fixation density overlays [8, 9], discarding a wealth of information that may be gained from the complete eye movement record.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI’12, May 5–10, 2012, Austin, Texas, USA.

Copyright 2012 ACM 978-1-4503-1015-4/12/05...\$10.00.

Visual search occurs naturally as a means of obtaining information about our surroundings, and there are two primary types of information processing that occur during visual search, parallel and serial [10]. Parallel search, exemplified in Figure 1, occurs when the target object is distinguishable from distractor objects by a single basic feature (color, motion, orientation, etc.), allowing parallel processing of objects and near constant reaction time. Serial search, exemplified in Figure 2, occurs when the target object is defined by more than one basic feature, requiring attentional shifts between objects until the target is located. Poor interface layout, individual interface component sizes, coloring, and other usability/design issues may lead to prolonged or excessive visual search. For the purposes of this paper and the corresponding research, *excessive visual search* (ES) is defined as *any onscreen search interval not directly related to task completion*.

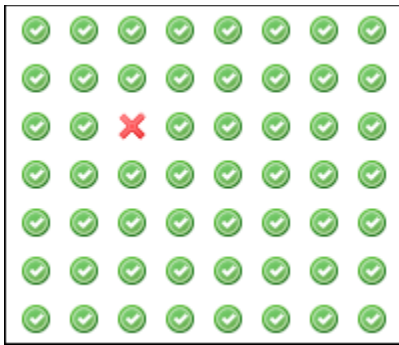



Figure 1. Parallel search example. Find the .



Figure 2. Serial search example. Find the .

While there has been a substantial amount of research on visual search [11, 12] and its implications on usability [13], to the best of our knowledge there has been very little progress in the automated identification of excessive visual search. In previous work, we described and evaluated several techniques for the automated classification of excessive visual search under mouse/keyboard based event segmentation. In this paper we describe several previously considered techniques for the automated classification of excessive visual search [14-16], present a variety of novel eye movement based segmentation methods, and provide an objective evaluation of the various segmentation and classification algorithms as compared to that of manually

classified excessive search intervals across an expanded data set.

We begin by defining visual search and its properties, exploring previous research on its applicability to the field of human-computer interaction, and providing a general description of the way it may be identified by automated analysis. We then present an overview of five segmentation algorithms used to divide the eye movement recordings into distinct intervals, along with seven classification algorithms used to identify intervals of excessive search. Finally, we present the methodology used to verify the accuracy of each algorithm, a description of our manual classification process, and a discussion of the results.

PREVIOUS WORK

Eye tracking was first employed as a usability metric in the 1950s by Fitts, et al. in their study of aircraft pilot behavior during landing procedures [17], and has since been utilized to analyze the usability of software applications, web pages, questionnaire formats, driving and navigation systems, and handheld devices. Jacob and Karn [18] survey 21 usability studies incorporating the use of eye movement metrics as a primary usability metric.

A common trend among these studies is the noted use of fixation count and duration as primary indicators of software usability. Eye movement metrics are often considered with little attempt to detect problem areas of a recording, instead attempting only to identify that usability problems exist within it [19, 20]. It is this problem in particular that we attempt to address in the current work.

Poole and Ball [7] provide a thorough summary of a variety of quantitative eye movement metrics (including the characteristics of fixations, saccades, scanpaths, blink rate, and pupil size) and their implications towards the usability of a given interface. For instance: fixations indicate areas of interest within an interface, and in the context of visual search short fixations and fixation clustering across a region may indicate difficulty identifying a target; regressive saccades and re-fixations are indicative of difficulty processing a target, often due to poor design or increased complexity; and increased pupil size is often indicative of cognitive effort and fatigue. A scanpath is an aggregate of fixations and saccades directed from one target of interest to the next.

According to [21]: “In a search task, an optimal scan path is viewed as being a straight line to a desired target, with relatively short fixation duration at the target.” This ideal, however, is often not the case, and visual search occurs frequently during the course of human-computer interaction due to the non-uniformity of basic tasks and the complex design patterns of modern user interfaces.

IDENTIFYING EXCESSIVE VISUAL SEARCH

The graphical user interface employed by the vast majority of modern software applications presents interface elements of a diverse range of colors, sizes, and dimensions. As a

Algorithm	Metrics	Region	Threshold
SEG-MK	Mouse/keyboard events	N/A	N/A
SEG-E	Eye position	Square	200 pixels
SEG-EC	Eye position	Circular	275 pixels
SEG-F	Fixation centroid	Square	75 pixels
SEG-FC	Fixation centroid	Circular	175 pixels

Table 1. Segmentation algorithms.

result, eye movements tend to follow a serial visual search pattern, with search time dependent on the number of distractor elements within the interface [10].

Due to the serial nature of visual search within an interface, usability issues in the design and implementation may be identified through analysis of excessive visual search [21]. While automated analysis cannot interpret the placement or quality of individual interface components in the same manner as a human observer, the quantitative properties of eye movements make it possible to identify intervals of excessive visual search within an eye movement recording. Automated identification of excessive search intervals makes it possible to reduce manual inspection of usability recordings, ignoring less relevant sections of the recording and focusing the attentions of a human observer on intervals of poor usability.

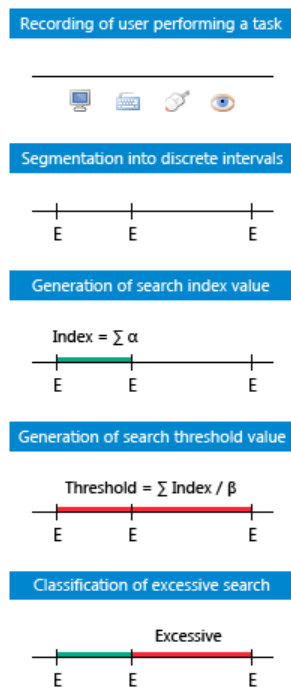


Figure 3. Automated detection of visual search. E represents mouse, keyboard, or eye movement events. α and β are variable metrics involved in classification.

Visual search leading directly to task completion, idle search behavior, and off-screen behavior are not considered excessive. Then, excessive visual search is identified according to the following basic algorithm, as shown in Figure 3, where specific eye movement metrics and threshold values vary:

1. The eye movement recording is parsed and divided into distinct intervals.
2. An index value is generated for each interval based on the characteristics of specific eye movement metrics within the interval.
3. A threshold value is generated, either empirically or as a function of the average index.
4. The intervals with an index above or below the threshold are classified as excessive visual search.

Presented in this work, segmentation algorithms attempt to identify and define distinct intervals of visual search (i.e. the scanpaths that make up a given search task), and classification algorithms attempt to identify the type of visual search being experienced. Threshold values are empirically selected via manual inspection of performance data to provide greater accuracy, though these values may vary with application domain and environment.

SEGMENTATION ALGORITHMS

Each recording is segmented into a set of unique time intervals considered to be candidates for ES according to a number of discrete events. Several segmentation methods are developed using different metrics and summarized in Table 1.

Mouse/Keyboard Segmentation

The mouse/keyboard segmentation method (SEG-MK) uses mouse/keyboard events to segment the eye movement recordings, as these represent conscious decisions by the user and imply that the user’s target has been found. Our first excessive search classification algorithms have been developed using this segmentation method, and as such it is used as a baseline for comparison with subsequent methods. That is, the classification algorithm thresholds were set and fixed using the SEG-MK segmentation method, and during the development of subsequent segmentation methods these thresholds were not changed.

Algorithm	Metrics (per interval)	Threshold
ES-F	Fixation count	> Average
ES-S	Average saccade amplitude	> Average
ES-P	Average pupil dilation	< Average
ES-SL	Total saccade amplitude	> Average
ES-SA	Convex hull area	> Average
ES-SAI	Convex hull area × Inflection count	> Average
ES-SAID	Convex hull area, Inflection count, Duration	> Average, > 5 Inflections, > 4 Seconds

Table 2. Classification algorithms.

Eye Movement-based Segmentation

While mouse/keyboard events provide an acceptable segmentation of the recording timelines for automated classification, use of this technique requires access to an additional layer of information. As well, it is our hypothesis that more detailed segmentation methods may improve the accuracy of the presented algorithms.

To further investigate, an eye movement based segmentation method (SEG-E) is developed using only the raw eye movement data to define event intervals as follows:

1. Mark the first/current point in the eye movement recording as the reference point.
2. Continue through the eye movement record until a point that is more than D units horizontally or vertically from the reference point is found.
3. Mark this new point as the current reference point and add it to the event list.
4. Repeat steps 1-3 until all points in the eye movement record have been examined.

Essentially, this defines a rectangular region of interest and allows for a certain amount of overlap as the user’s attention shifts between elements. For our purposes, all off-screen points were ignored and D was empirically set to 200 pixels. Nevertheless, this parameter is application dependent.

An additional segmentation method (SEG-EC) is derived from the SEG-E variant, using the Euclidean distance between points for comparison to D, essentially defining a circular region of interest. For this purpose, all off-screen points are again ignored, and in the current experiments D is empirically set to 275 pixels.

Fixation Based Segmentation

To examine whether additional accuracy could be gained by considering only fixation points, rather than the raw eye movement data, providing the segmentation that is based on strictly defined points of attention. The eye movement based algorithms are modified to use the fixation points filtered by an I-VT algorithm [22].

In comparison to the raw eye movement signal, fixations are often more directly indicative of attention and interest. Fixation based segmentation followed the basic algorithm described previously for SEG-E, with the primary difference being the data set to which the algorithm is applied. In comparison, SEG-E and SEG-EC operate on the raw eye movement signal, while the fixation based algorithms (SEG-F and SEG-FC) operate on the fixations identified within the raw eye movement signal.

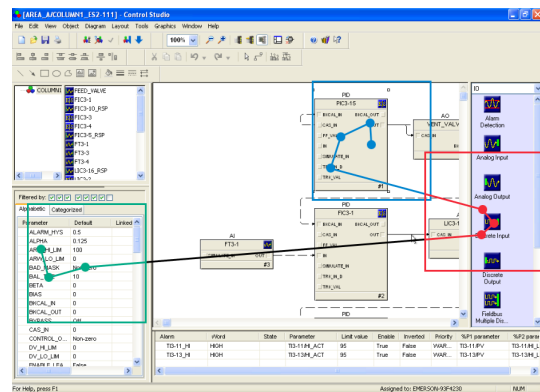


Figure 4. Fixation based segmentation (ES-F). Fixations represented by dots, saccades represented by lines, and logical segments separated by color.

This results in two further segmentation methods: SEG-F, shown in Figure 4, using rectangular regions of interest with a threshold D of 75 pixels; and SEG-FC, using circular regions of interest using a threshold D of 175 pixels. A velocity threshold of 30°/sec is employed for the velocity threshold algorithm (I-VT).

CLASSIFICATION ALGORITHMS

Seven classification algorithms are presented as described previously in [14, 16] and summarized in Table 2. The ES-F, ES-S, ES-P, and ES-SL algorithms rely on basic attributes of the human visual system, while the scanpath based algorithms ES-SA, ES-SAI, and ES-SAID incorporate more advanced aspects of visual search.

Fixation Based Algorithm

The fixation based algorithm (ES-F) uses the fixation count of each interval as its index value, and the average fixation count as the threshold. Search intervals with a fixation count above the threshold are classified as ES.

Saccade Based Algorithm

The saccade based algorithm (ES-S) uses the average saccade amplitude of each interval as its index value, and the average across all intervals as the threshold. Search intervals with an average saccade amplitude above the threshold are classified as ES.

Pupil Based Algorithm

The pupil based algorithm (ES-P) uses the average pupil dilation of each interval as its index value, and the average of average pupil dilations as the threshold. Search intervals with average pupil dilation below the threshold are classified as ES.

Scanpath Length Algorithm

The scanpath length algorithm (ES-SL) uses the total saccade amplitude of each interval as its index value, and the average across all intervals as the threshold. Search intervals with saccade amplitude above the threshold are classified as ES.

Scanpath Area Algorithm

The scanpath area algorithm (ES-SA) uses the area of the convex hull formed by fixation points within each interval as its index value, and the average across all intervals as the threshold. The area of the convex hull is indicative of the total search area, with smaller values indicating efficient search behavior. Search intervals with a convex hull area above the threshold are classified as ES.

Scanpath Area/Inflections Algorithm

The scanpath area/inflections algorithm (ES-SAI) uses the area of the convex hull formed by fixation points multiplied by the number of times the scanpath changes direction (inflections) for each interval as its index value, and the average index value across all intervals as the threshold. Inflections of the scanpath are indicative of attention shifts, with larger inflections counts suggesting increased visual search. Search intervals with an index value above the threshold are classified as ES.

Scanpath Area/Inflections/Duration Algorithm

The scanpath area/inflections/duration algorithm (ES-SAID) uses multiple index values for each interval: the area of the convex hull formed by fixation points, the inflection count, and the total duration of the interval. Longer intervals between subsequent mouse/keyboard events can indicate difficulties in locating the next target, therefore increasing the probability of the ES. Search intervals with a convex hull area above the average, an inflection count greater than 5, or duration of more than 4 seconds are classified as ES. Threshold values were selected empirically.

METHODOLOGY

Software

Usability testing is performed on the DeltaV process control software, utilizing an interface similar to common diagram editing applications (such as Microsoft Visio), as part of a related but separate study for developing methods of objective usability evaluation. Screen recordings and their corresponding scanpath/input event overlays are viewed for manual classification of visual search with Tobii Studio. All algorithms and data analysis are implemented and performed in MATLAB.

Apparatus

Usability testing is conducted with the Tobii X120 eye tracker running at 120Hz. DeltaV is run on a Dell Optiplex 745 with 4 GB of RAM, and displayed on a 19 inch flat panel monitor with a resolution of 1280×1024. A velocity threshold (I-VT) of 30°/sec is used to reduce the eye movement data into the fixations and saccades [22].

Participants

A total of 14 student volunteers and 12 experienced users have participated in the usability testing. Due to the substantial amount of time required for manual classification, randomly selected recordings from 21 of these subjects are used.

Manual Classification

Four basic user behaviors are considered during the manual classification: task completion (TC), in which the user is performing the operations necessary to complete a given task; excessive visual search (ES), in which the user is experiencing prolonged difficulties finding the interface components necessary for task completion; idle (IL), in which the user is waiting for the interface to respond after a specific action is performed; and off-screen (OS), in which the user is reading task-related instructions presented outside the boundaries of the computer monitor.

Manual classification of the task recordings was performed by a trained research assistant using superimposed eye movement traces to build a classification baseline. A full and thorough description of the manual classification process is described in [15].

Procedure

Participants are given a series of 15 tasks to complete in the process control application, during which screen recordings, eye movement records, and input logs are generated and synchronized for each task. All tasks are performed within the same user interface and follow a uniform procedure: 1) delete component; 2) add component; 3) make connections between components; 4) transfer changes to controller simulator; 5) change component value; 6) change interface view; 7) save changes and exit. Tasks are similar to each other with specific interface components varied to reduce learning effects.

ES intervals are then manually classified for the recordings (chosen arbitrarily from unique subjects with a uniform

distribution of trials) according to the previous description of manual search classification. The automated ES classification algorithms are subsequently run using the various segmentation methods with the eye movement recordings and input logs of the manually identified recordings. ES intervals generated by the automated analysis are compared to those provided by manual classification to determine the percentage of automatically identified search intervals that were correctly identified as excessive and the percentage of ES intervals missed or erroneously identified.

To determine the relative performance of the different segmentation/classification algorithms, each algorithm was then run separately across all eye movement records and computation times were measured in seconds.

RESULTS

Note that in all the figures, an asterisk (*) indicates statistical significance of $p < 0.05$ and a dagger (†) indicates statistical significance of $p < 0.001$, as determined by a one-way ANOVA test between the algorithms of a particular group. For the segmentation algorithms, the comparison is performed across all classification algorithms, and vice versa, where $F(6, 140)$ for segmentation algorithms and $F(4, 100)$ for classification algorithms. For example, the label “ES-F *” indicates that there is a significant main effect in the values produced by the ES-F algorithm when compared across segmentation algorithms.

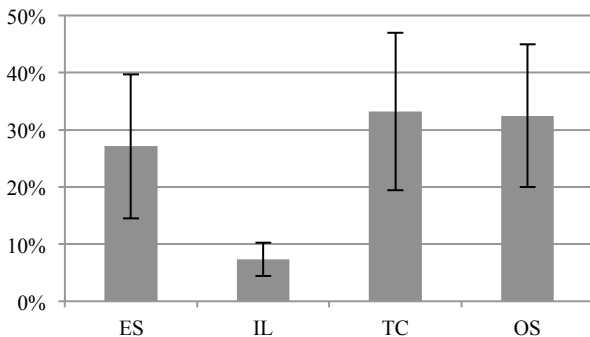


Figure 5. Manual Search Behavior.

Manual Classification

Figure 5 presents a summary of the relative distribution of search behavior identified during manual classification. Across the 21 eye movement recordings, task completion and offscreen behavior comprise roughly one third of the recording time each, while excessive visual search and idle search constitute the remaining duration. The amount of idle search is relatively low, occurring only when the interface is unresponsive, and often resembling excessive search behavior. The overall amount of non-task completion behavior is substantial, on average leaving 67% of the recording time as irrelevant. The difference in time between different behaviors was statistically significant, $F(3, 80) = 24.01, p < 0.001$.

Automated Classification of Excessive Visual Search

Average Percent of Total Time Classified

$$\text{Percent of Total} = \frac{\text{total classified time}}{\text{total recording time}} \quad (1)$$

Figure 6 presents a summary of the average percent of total time classified by each algorithm. Assuming correct identification of ES intervals, a lower percent of total time classified as ES indicates more precise identification. Averaged across all classification algorithms, segmentation by mouse and keyboard events (SEG-MK) on average marks the smallest amount of the total recording time as ES ($M = 47\%$, $SD = 8\%$), while segmentation based on fixations (SEG-FC) marks the largest ($M = 54\%$, $SD = 12\%$). Across all segmentation algorithms, pupil based classification (ES-P) on average marks the smallest amount of the total recording time as ES ($M = 31\%$, $SD = 14\%$), while classification based on scanpath length (ES-SL) marks the largest ($M = 64\%$, $SD = 6\%$).

Average Percent Correctly Classified

$$\text{Percent Correct} = \frac{\text{correctly classified time}}{\text{total manual time}} \quad (2)$$

Figure 7 presents a summary of the average percent of time correctly identified as excessive search by each algorithm. Averaged across all classification algorithms, segmentation by mouse and keyboard events (SEG-MK) has the highest average percent of correctly identified intervals ($M = 55\%$, $SD = 8\%$), while segmentation based on fixations (SEG-FC) has the lowest ($M = 45\%$, $SD = 13\%$). Across segmentation algorithms, scanpath based classification (ES-SL) has the highest average percent of correctly identified intervals ($M = 64\%$, $SD = 5\%$), while pupil based classification (ES-P) has the lowest ($M = 25\%$, $SD = 16\%$).

Average Percent Error

$$\text{Percent Error} = \frac{\text{misclassified} + \text{unclassified time}}{\text{total recording time}} \quad (3)$$

Figure 8 presents a summary of the average percent of time erroneously classified by each algorithm. Averaged across all classification algorithms, eye movement based segmentation (SEG-EC) has the lowest average percent error ($M = 31\%$, $SD = 4\%$), while fixation based segmentation (ES-FC) has the highest ($M = 35\%$, $SD = 2\%$). Across all segmentation algorithms, fixation based classification (ES-F) has the lowest average percent error ($M = 26\%$, $SD = 3\%$), while pupil based classification (ES-P) has the highest ($M = 36\%$, $SD = 2\%$).

Average Computational Performance

Recording times did not exceed 4.8 minutes ($M = 2.6$ minutes, $SD = 1.0$ minutes). Segmentation algorithms did not exceed 3.5 seconds per recording ($M = 1.3$ seconds, $SD = 0.7$ seconds), and classification algorithms did not exceed 1.0 seconds per recording ($M = 0.3$ seconds, $SD = 0.2$ seconds).

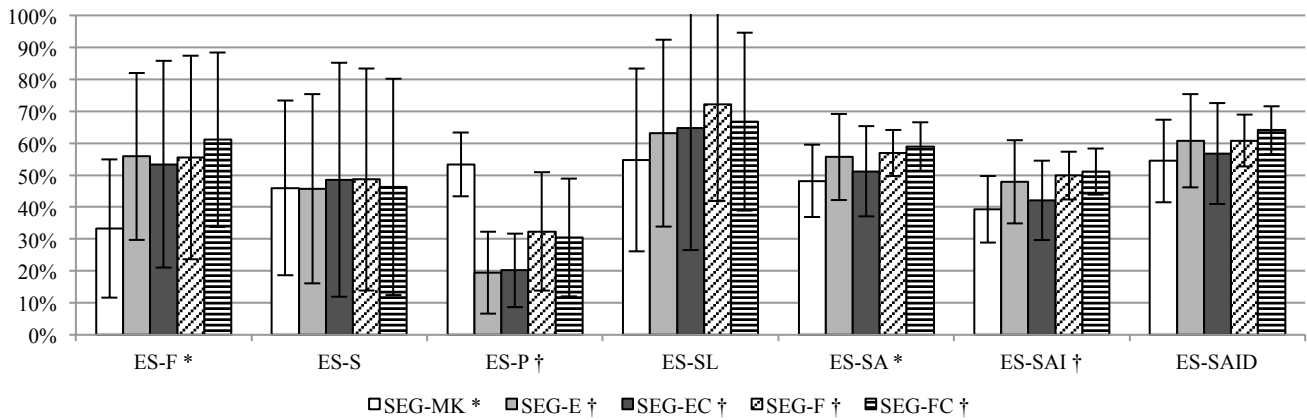


Figure 6. Average Percent of Total Time Classified.

DISCUSSION

Segmentation Algorithms

Of the various segmentation methods, segmentation by mouse and keyboard events (SEG-MK) provides the most stability, with less fluctuation in the results obtained between algorithms. Classification algorithm thresholds are set using the SEG-MK algorithm, which may account for its relative stability; it is likely that the accuracy of the various segmentation methods could be improved by modifying these thresholds. Mouse/keyboard events represent a conscious action by the user, often directly related to task completion. This leads to some amount of overlap in behavioral intervals, and increases the likelihood that any given interval contains at least some amount of task completion behavior.

Eye movement based segmentation (SEG-E and SEG-EC) obtains the best results overall, often having a higher percent of correctly identified intervals and a lower error than the other segmentation methods. Eye movement based segmentation defines intervals according to approximated regions of interest, allowing the users' attention to determine the logical segments of the recording. In general, SEG-E outperforms SEG-EC; this may be due to the fact that user interface elements (i.e. buttons, menus, controls, etc.) are often rectangular.

Fixation based segmentation (SEG-F and SEG-FC) shows the least accuracy of the considered segmentation methods, generally having a higher percent of total time classified, a lower percent correct, and a higher percent error than the opposing segmentation methods. Inaccuracy in the fixation based segmentation methods may be due, in part, to inaccuracies inherent in the I-VT algorithm used to identify fixations and the reduced specificity caused by the merging of individual data points into discrete fixations/saccades.

Eye movement based segmentation (SEG-E) may be the most useful algorithm, relying solely on the eye movement record without the need for extraneous information such as input events or application state. In general, the SEG-E algorithm surpasses all segmentation algorithms with the

exception of SEG-EC in classification accuracy, and provides the highest accuracy of all considered algorithms when paired with the ES-SAID classification algorithm.

Classification Algorithms

Of the classification algorithms, fixation based classification (ES-F) obtains the best results using eye movement based segmentation (SEG-EC), with an average of 53% of total time classified, 60% correctly classified, and 25% erroneously classified. Saccade based classification (ES-S) obtains the best results using eye movement based segmentation (SEG-EC), with an average of 49% of total time classified, 52% correctly classified, and 29% erroneously classified. Pupil based classification (ES-P) obtains the best results using segmentation by mouse and keyboard events (SEG-MK), with an average of 53% of total time classified, 52% correctly classified, and 39% erroneously classified. Classification based on scanpath length (ES-SL) obtains the best results using eye movement based segmentation (SEG-EC), with an average of 65% of total time classified, 69% correctly classified, and 28% erroneously classified. Classification based on scanpath area (ES-SA) obtains the best results using eye movement based segmentation (SEG-E), with an average of 56% of total time classified, 65% correctly classified, and 32% erroneously classified. Classification based on a combination of scanpath area and inflections (ES-SAI) obtains the best results using eye movement based segmentation (SEG-EC), with an average of 42% of total time classified, 53% correctly classified, and 31% erroneously classified. Classification based on a combination of scanpath area, inflections, and duration (ES-SAID) obtains the best results using eye movement based segmentation (SEG-E), with an average of 61% of total time classified, 73% correctly classified, and 31% erroneously classified.

Classification based on a combination of scanpath area, inflections, and duration (ES-SAID) may be the most useful algorithm, averaging the highest percent correctly identified and an error rate somewhere between the other algorithms. The ES-SAID algorithm, in its current state, could be used

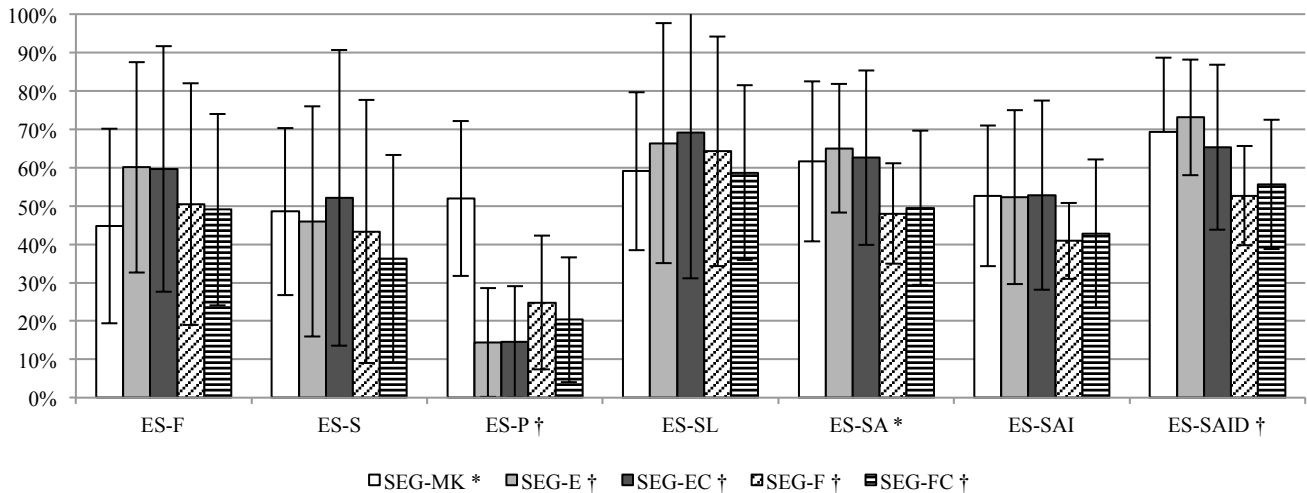


Figure 7. Average Percent Correct.

to discard irrelevant sections of recording with little loss of relevant data, substantially reducing the time required to review recorded data.

Pupil based classification (ES-P) is clearly the least effective of the considered algorithms, averaging the lowest percent correctly identified and the highest error rate. While pupil diameter is indicative of cognitive effort, it is also susceptible to the influence of light, fatigue, and emotion [7]. As such, it seems to be a poor indicator from which to draw conclusions about usability in the context of this work.

Of the basic eye movement metrics, fixation count seems to be the most reliable indicator of excessive search behavior, with a greater number of fixations indicating more extensive visual search and processing among distractor elements in the interface. Average saccade amplitude is less indicative of excessive search, as this reflects similarly the extent of visual search and processing, but is also largely affected by the size and location of interface elements. Scanpath length and area indicate the overall span of attention within the interface, and are more accurate indicators of excessive search than the basic metrics. Scanpath inflections indicate shifts in user attention and scanpath duration indicates the extent of processing; applying these in conjunction with scanpath area provides greater accuracy than any individual metric for the identification of excessive visual search.

Peculiarities

Throughout the course of manual classification, we have noticed several eye movement patterns indicative of excessive visual search. Difficulty selecting an interface element (often due to size) generally results in a scanpath concentrated within a small region, with long fixation durations and small saccade amplitudes. Difficulty finding interface elements (often text within a list/menu) generally results in vertical scanning of a localized screen region, with short fixation durations, slightly larger saccade

amplitudes, and an increased number of vertical inflections. Difficulty understanding the interface layout (due to non-intuitive design or unclear instructions) often results in extensive scanning across multiple regions of the screen, with large saccade amplitudes, slightly longer fixation durations, and erratic inflections. In addition, while its classification as excessive search is debatable, there are a number of occurrences of what could be deemed habitual search, these are immediately preceded and followed by off-screen behavior and generally consist of 3-7 short fixations not necessarily related to task completion. It should be noted that while these are the most prevalent indicators of excessive visual search, they should not be considered comprehensive.

Classification error as defined in the previous section is a combination of the unclassified excessive search intervals and the intervals misclassified as excessive search. Several factors contribute to this error. The most substantial of these may be the accuracy of segmentation; manual classification is a relatively fine-grained approach, defining intervals of excessive search at the millisecond level, while automated segmentation provides a much coarser separation. As such, the intervals defined by manual and automated classification are not exactly equivalent. In this sense, a certain amount of error is unavoidable, but can be reduced by improving the accuracy of segmentation. Additionally, excessive search is not clearly defined by a single eye movement pattern, and as a result it is difficult to pinpoint excessive search intervals using only a single metric (as is illustrated by the relatively high accuracy of the ES-SAID algorithm, which employs a combination of metrics).

Limitations

The primary limitation of this technique is the inherent cost of suitable eye tracking equipment. Unfortunately, automated classification of excessive visual search does not yet provide the level of accuracy and precision necessary to eliminate the need for a trained human observer. As such,

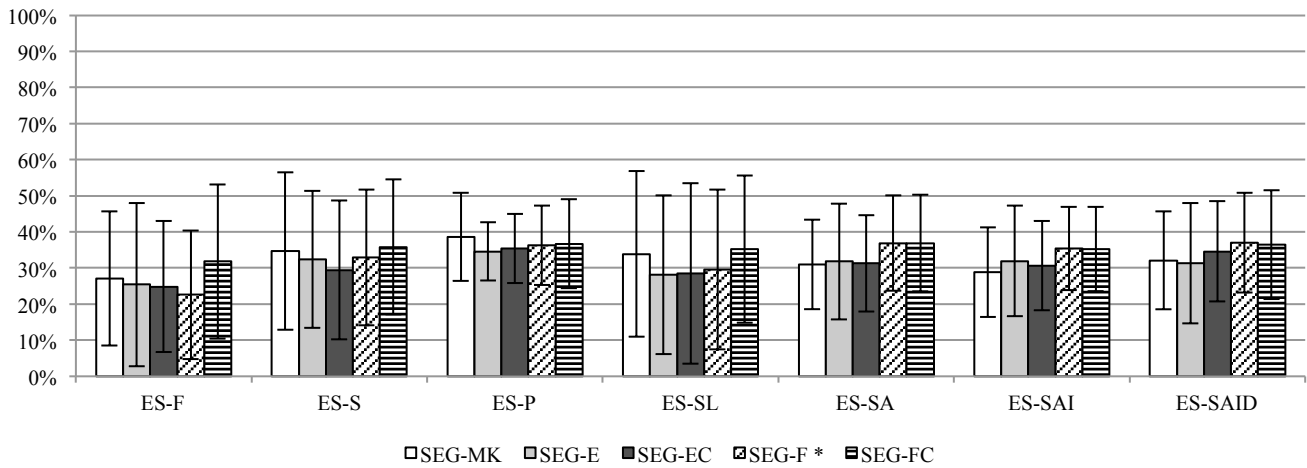


Figure 8. Average Percent Error.

the cost of eye tracking equipment must be compared to the possible time savings that may be obtained by its use. However, as eye tracking becomes more ubiquitous, and novel methods of producing cheap eye tracking equipment receive more attention [23, 24], this limitation will inevitably be resolved.

Another issue is the applicability of the usability problems identified in the process control interface to the usability of software interfaces in a more general context. For instance, an excessive amount of time spent reading a control list within a process control interface would not necessarily be considered an excessive amount of time while reading the contents of a web browser. In general, however, these metrics are valid for many applications in which direct user manipulation is the primary context of use (i.e. diagram editing, photo editing, etc.), though this assertion is currently untested.

Unfortunately, the time, effort, and training required for manual classification is extremely prohibitive, and, as such, it was only possible to obtain manual classification data from a single source in the current work. This issue may have introduced some amount of bias or inaccuracy into the performance comparison, as manual inspection is at least partially subjective and may vary from person to person. Despite this, the strictest adherence to formulaic inspection of usability recordings was employed to ensure similarity in the manual intervals produced.

Applications

The current paper focuses on the identification of intervals of excessive visual search within the recording, allowing software developers to more easily inspect areas of concern; however, the range and scope of possible applications is much wider. Based on classification statistics and assuming accurate classification it may be possible to provide further diagnostics. For example, the percentage of intervals classified as excessive visual, or a related measure, may be applied to determine the overall usability of the interface according to pre-defined thresholds; or similarly, it may be

possible to determine the type and degree of usability issues within an interval based on the scanpath characteristics displayed. These are areas of active research, however, and are beyond the scope of this paper.

CONCLUSION

The primary shortcoming of usability testing is its qualitative nature, requiring detailed and time consuming analysis by a trained observer. In our experience, each minute of recorded data required approximately one hour of manual review. With average recording times of 3 minutes per subject, this equates to roughly 63 hours spent on manual classification of the 21 usability recordings considered in this paper. In comparison, automated classification with the ES-SAID algorithm is able to correctly identify an average of 73% of the excessive search intervals while discarding roughly 40% of the total recording time across the 21 recordings in less than 5 minutes. This is a substantial improvement, providing a potential savings of roughly 25 hours spent on manual classification.

In this paper we have described several previously considered techniques for the automated classification of excessive visual search, presented a variety of novel eye movement based segmentation methods, and provided an objective evaluation of the various segmentation and classification algorithms across and expanded data set. The results indicate that automated classification of excessive visual search may be employed to substantially reduce the amount of recorded data reviewed during usability testing.

Of the considered algorithms, the segmentation algorithm SEG-E and classification algorithm ES-SAID provided the most accurate detection of excessive visual search, confirming that eye movement based segmentation is able to provide more accurate search intervals than segmentation based purely on mouse/keyboard events. This was accomplished by defining search intervals according to variable regions of interest across the screen using the recorded eye gaze signal, and identifying excessive visual

search through a combined analysis of attentional span (scanpath area), attentional shifts (inflection count), and processing complexity (scanpath duration).

Future research in this area will likely involve improvements and innovations in both segmentation and classification, the eventual goal being to detect not only when usability problems occur, but to provide further analysis of the location and reason for these problems within the interface. In addition to this, we hope to develop more complex algorithms for the automated detection of additional search behaviors, such as task completion and idle search.

REFERENCES

- [1] A. A. Witold, *et al.*, "Consolidating the ISO Usability Models," presented at the 11th International Software Quality Management Conference and 8th Annual INSPIRE Conference, 2003.
- [2] J. Rubin and D. Chisnell, *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*, 2 ed. New York, NY: Wiley, 2008.
- [3] L. Vukelja, *et al.*, "Are engineers condemned to design? a survey on software engineering and UI design in Switzerland," presented at the 11th IFIP TC 13 international conference on Human-computer interaction, Rio de Janeiro, Brazil, 2007.
- [4] J. S. Dumas and J. C. Redish, *A Practical Guide to Usability Testing*: Intellect Books, 1999.
- [5] R. J. Leigh and D. S. Zee, *The Neurology of Eye Movements*, 4 ed.: Oxford University Press, USA, 2006.
- [6] A. Duchowski, *Eye Tracking Methodology: Theory and Practice*, 2nd ed.: Springer, 2007.
- [7] A. Poole and L. J. Ball, "Eye tracking in human-computer interaction and usability research: current status and future prospects," in *Encyclopedia of Human-Computer Interaction*, C. Ghaoui, Ed., ed: Idea Group, 2005, pp. 211-219.
- [8] L. J. Ball, *et al.*, "Applying the Post-Experience Eye-Tracked Protocol (PEEP) Method in Usability Testing," *Interfaces*, vol. 67, pp. 15-19, 2006.
- [9] M. C. Russell, "Hotspots and hyperlinks: using eye-tracking to supplement usability testing," *Usability News*, vol. 7, 2005.
- [10] J. M. Wolfe, "What Can 1 Million Trials Tell Us About Visual Search?," *Psychological Science*, vol. 9, pp. 33-39, 1998.
- [11] J. Shen, *et al.*, "Distractor ratio influences patterns of eye movements during visual search," *Perception*, vol. 29, pp. 241-250, 2000.
- [12] I. D. Gilchrist and M. Harvey, "Refixation frequency and memory mechanisms in visual search," *Current Biology*, vol. 10, pp. 1209-1212, 2000.
- [13] A. J. Hornof and T. Halverson, "Cognitive strategies and eye movements for searching hierarchical computer displays," in *SIGCHI conference on Human factors in computing systems*, Ft. Lauderdale, Florida, USA, 2003, pp. 249-256.
- [14] O. V. Komogortsev, *et al.*, "Eye movement driven usability evaluation via excessive search identification," in *14th International Conference on Human-Computer Interaction*, 2011.
- [15] O. Komogortsev, *et al.*, "EMA: Automated eye-movement-driven approach for identification of usability issues," in *Design, user experience, and usability. Theory, methods, tools and practice*. vol. 6770, A. Marcus, Ed., ed: Springer Berlin / Heidelberg, 2011, pp. 459-468.
- [16] O. Komogortsev, *et al.*, "Aiding usability evaluation via detection of excessive visual search," presented at the 2011 ACM SIGCHI Conference on Human Factors in Computing Systems (CHI), Vancouver, BC, Canada, 2011.
- [17] P. M. Fitts, *et al.*, "Eye movements of aircraft pilots during instrument-landing approaches," *Aeronautical Engineering Review*, vol. 9, pp. 24-29, 1950.
- [18] R. Jacob and K. Karn, "Commentary on Section 4. Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises," in *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*, ed: Elsevier, 2003, pp. 573-607.
- [19] F. T. W. Au, *et al.*, "Automated usability testing framework," presented at the Proceedings of the ninth conference on Australasian user interface - Volume 76, Wollongong, Australia, 2008.
- [20] M. Ivory and A. Chevalier, "A Study of Automated Web Site Evaluation Tools," University of Washington, Department of Computer Science 2002.
- [21] J. H. Goldberg and X. P. Kotval, "Computer interface evaluation using eye movements: methods and constructs," *International Journal of Industrial Ergonomics*, vol. 24, pp. 631-645, 1999.
- [22] O. V. Komogortsev, *et al.*, "Standardization of Automated Analyses of Oculomotor Fixation and Saccadic Behaviors," *IEEE Transactions on Biomedical Engineering*, vol. 57, pp. 2635-2645, 2010.
- [23] W. Sewell and O. Komogortsev, "Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network," in *28th of the international conference extended abstracts on Human factors in computing systems*, Atlanta, Georgia, USA, 2010, pp. 3739-3744.
- [24] J. S. Agustin, *et al.*, "Low-cost gaze interaction: ready to deliver the promises," presented at the Proceedings of the 27th international conference extended abstracts on Human factors in computing systems, Boston, MA, USA, 2009.