

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/222500586>

Computer interface evaluation using eye movements: Methods and constructs

Article in *International Journal of Industrial Ergonomics* · October 1999

DOI: 10.1016/S0169-8141(98)00068-7

CITATIONS

1,018

READS

7,691

2 authors:



Joseph H. Goldberg

Meta

78 PUBLICATIONS 5,028 CITATIONS

[SEE PROFILE](#)



Xerxes P Kotval

Nokia Bell-Labs

13 PUBLICATIONS 1,242 CITATIONS

[SEE PROFILE](#)



ELSEVIER

International Journal of Industrial Ergonomics 24 (1999) 631–645

International Journal of

**Industrial
Ergonomics**

www.elsevier.nl/locate/ergon

Computer interface evaluation using eye movements: methods and constructs

Joseph H. Goldberg*, Xerxes P. Kotval¹

The Pennsylvania State University, Department of Industrial and Manufacturing Engineering, 207 Hammond Building, University Park, PA 16802-1401, USA

Received 1 March 1998; received in revised form 13 March 1998; accepted 7 July 1998

Abstract

Eye movement-based analysis can enhance traditional performance, protocol, and walk-through evaluations of computer interfaces. Despite a substantial history of eye movement data collection in tasks, there is still a great need for an organized definition and evaluation of appropriate measures. Several measures based upon eye movement locations and scanpaths were evaluated here, to assess their validity for assessment of interface quality. Good and poor interfaces for a drawing tool selection program were developed by manipulating the grouping of tool icons. These were subsequently evaluated by a collection of 50 interface designers and typical users. Twelve subjects used the interfaces while their eye movements were collected. Compared with a randomly organized set of component buttons, well-organized functional grouping resulted in shorter scanpaths, covering smaller areas. The poorer interface resulted in more, but similar duration, fixations than the better interface. Whereas the poor interface produced less efficient search behavior, the layout of component representations did not influence their interpretability. Overall, data obtained from eye movements can significantly enhance the observation of users' strategies while using computer interfaces, which can subsequently improve the precision of computer interface evaluations.

Relevance to industry

The software development industry requires improved methods for the objective analysis and design of software interfaces. This study provides a foundation for using eye movement analysis as part of an objective evaluation tool for many phases of interface analysis. The present approach is instructional in its definition of eye movement-based measures, and is evaluative with respect to the utility of these measures. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Eye movements; HCI; Computer interface design; Software evaluation; Fixation algorithms

1. Introduction

1.1. Interface evaluation

The software development cycle requires frequent iterations of user testing and interface

*Corresponding author.

¹Present address: Lucent Technologies, Bell Laboratories, Holmdel, NY.

modification. These interface evaluations, whether at initial design or at later test and evaluation stages, should assess system functionality and the impact of the interface on the user. Earlier, design evaluation methods include cognitive walkthroughs, heuristic, review-based, and model-based evaluations. At more mature product phases, performance-based experiments, protocol/observation, and questionnaires are frequently used as a basis for evaluation (Dix et al., 1998). Performance-based studies assess errors and time to complete specified operations or scenarios (Wickens et al., 1998).

Interface evaluation and usability testing are expensive, time-intensive exercises, often done with poorly documented standards and objectives. They are frequently qualitative, with poor reliability and sensitivity. Provision of an improved tool for rapid and effective evaluation of graphical user interfaces was the motivating goal underlying the present work assessing eye movements as an indicator of interface usability.

1.2. Eye movements on displays

While using a computer interface, one's eye movements usually indicate one's spatial focus of attention on a display. In order to foveate informative areas in a scene, the eyes naturally fixate upon areas that are surprising, salient, or important through experience (Loftus and Mackworth, 1978). Thus, current gaze points on a display can approximate foci of attention over a time period. When considering short time intervals, however, one's attentional focus may lead or lag the gaze point (Just and Carpenter, 1976). By choosing long enough sampling intervals for eye movements, temporal leads/lags should be averaged out.

Applied eye movement analysis has at least a 60 yr history in performance and usability assessments of spatial displays within information acquisition contexts such as aviation, driving, X-ray search, and advertising. Buswell (1935) measured fixation densities and serial scanpaths while individuals freely viewed artwork samples, noting that eyes follow the direction of principal lines in figures, and that more difficult processing produced longer fixation durations. Mackworth (1976) noted that

higher display densities produced 50–100 ms longer fixation durations than lower density displays. Non-productive eye movements more than 20° from the horizontal scanning axis strongly increased as a percentage of all eye movements as the display width and density increased. Kolers et al. (1981) measured eye fixations (number, number per line, rate, duration, words per fixation) as a function of character and line spacing in a reading task. More fixations per line (and fewer fixations per word) were associated with more tightly-grouped, single-spaced material. Fewer, yet longer fixations were made with smaller, more densely packed text characters. Yamamoto and Kuto (1992) found improved Japanese character reading performance associated with series of sequential rather than backtracking eye movements. Eye tracking has aided the assessment of whether the order of product versus filler displays in a television commercial influences one's attention to that product (Janiszewski and Warlop, 1993). Using eye movement analyses while scanning advertisements on telephone yellow pages, quarter-page ad displays were much more noticed than text listings, and color ads were perceived more quickly, more often, and longer than black and white ads (Lohse, 1997).

Prior eye movement-based interface and usability characterizations have relied heavily upon cumulative fixation time and areas of interest approaches, dividing an interface into predefined areas. Transitions into and from these areas, as well as time spent in each area, are tallied. While these approaches can signal areas where more or less attention is spent while using a display, few investigations have considered the complex nature of scanpaths, defined from a series of fixations and saccades on the interface. Scanpath complexity and regularity measures are needed to approach some of the subtler interface usability issues in screen design.

1.3. Objective

Eye tracking systems are now inexpensive, reliable, and precise enough to significantly enhance system evaluations. While the hardware technology is quite mature (Young and Sheena, 1975, for a general review), methods of evaluating data from eye

tracking experiments are still somewhat immature and disorganized. The objective of the present paper is to provide an introduction and framework for eye movement data analysis techniques. These eye movement measures and algorithms are presented in light of results from an experiment presenting users with both “good” and “poor” interfaces.

2. Methods

Scanpaths were collected from 12 subjects while using both “good” and “poor” software interfaces. The resulting scanpaths were characterized using a number of quantitative measures, each designed to characterize different aspects of scanpath behaviors and relate to the cognitive behavior underlying visual search and information processing. A comparison of expected user search behavior using each interface with the results of scanpath measures were used to determine the relative effectiveness of each measure.

2.1. Interface stimuli

Example “good” and “poor” interfaces were programmed to provide a well-controlled and equally familiar environment for all subjects in this study. Their primary purpose was not to evaluate the usability of these particular interfaces per se; rather, they provided a means for validating the various created measures. Fig. 1 shows two of these interfaces. The interface showed a work area with a panel of tool buttons, much like a drawing package.

The good–poor distinction was based upon physical grouping of interface tool buttons. Users expect physically grouped components to be related by some common characteristic, whether physical or conceptual (Wickens and Carswell, 1995). Exploiting this, the “good” interface grouped eleven components into three functionally related groups: editing, drawing, and text manipulation tools (Fig. 1, left panel). These functional groupings were intended to allow relatively efficient tool search, compared with the poorly designed interface (Fig. 1, right panel) intended to cause less efficient visual search. The “poor” interface provided a randomized (i.e., not functional or conceptual) relationship within each tool group.

To verify a substantial difference in perceived quality, fifty typical users and thirty interface design experts rated each interface on a scale from 1 (excellent) to 5 (unacceptable). The functionally grouped interface averaged 1.35, between good and excellent, whereas the randomly grouped interface averaged 4.53, between unacceptable and poor. Thus, the two interfaces were confirmed as substantially different in design quality.

2.2. Apparatus and calibration

The experiment was hosted on a PC with a 13 in (33 cm) VGA monitor with mouse/windows control. A second computer, remotely activated by the host computer, controlled the eye tracking system, a DBA systems Model 626 infrared corneal reflection system (Fig. 2). An infrared sensitive CCD video camera was positioned just below the host

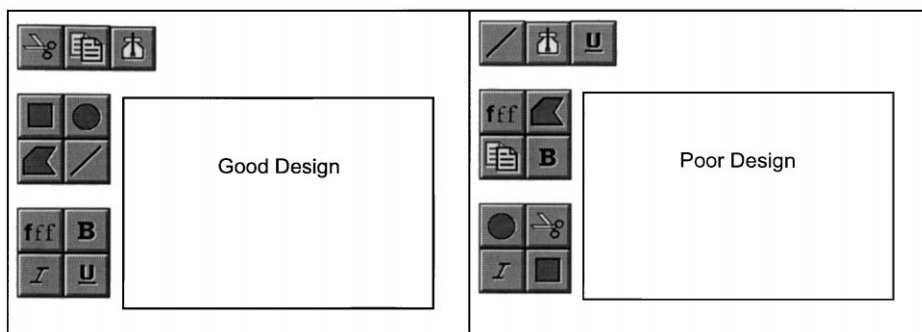


Fig. 1. Interface designs. Left panel: good design; right panel: poor design.

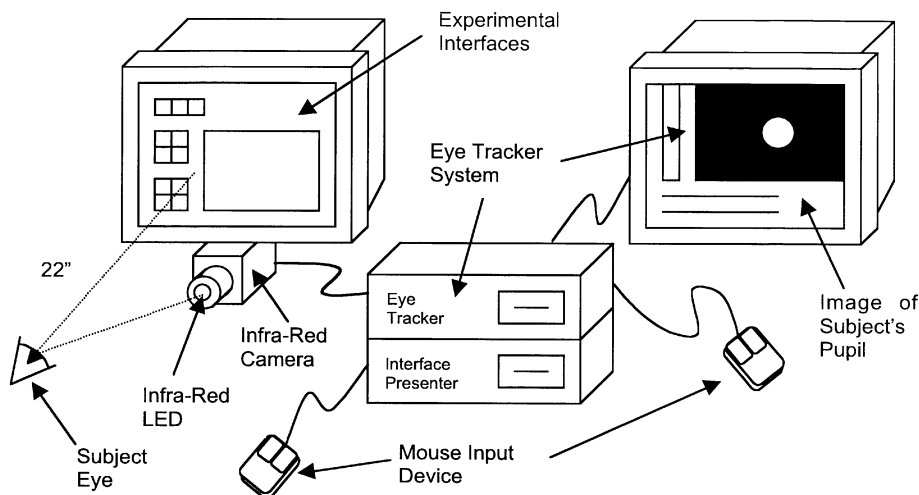


Fig. 2. Experimental apparatus, showing eye tracker with infrared-sensitive camera lens.

computer's monitor. The camera contained an LED inline with its focal axis, generating an illuminated pupil and light glint (first Purkinje reflection) on the subject's cornea. The head posture and eye location were maintained with a head/chin rest, such that the eye was 22 in (56 cm) from the screen, and level with its center. At this distance, the screen subtended 21° and 16° of horizontal and vertical visual angle, respectively. Each 65×65 pixel tool button in the interface subtended 1 in, or 2.2° of horizontal visual angle.

Video images of the pupil and Purkinje reflection were captured at 60 Hz by the eye tracker and assigned light intensity values to each pixel in the digital image. An intensity threshold filtered the video image until the pupil image was isolated. Eye tracker software located the center of the pupil and calculated the vector from it to the corneal light glint. A calibration procedure related this vector with Cartesian coordinates on the interface screen, providing the subject's eyegaze location, or point-of-regard (POR). The POR coordinates were collected and stored in a datafile for later processing.

Calibration used a set of 9 screen locations, and was checked with each block (33 trials) in each subject's session. The criterion for a successful calibration equated to residuals that were less than 0.5 cm (0.5° visual angle) from actual target location. In other words, the eye tracker software

estimate of target location was not more than 10 pixels away from the actual target location.

2.3. Subjects

Twelve subjects (7 female, 5 male) participated in this study. Ages ranged from 20 to 27 yr (mean 23 yr). Participants averaged 4.8 yr of experience using typical windowing software, spending an average of 15.3 h a week using software interfaces. Because corrective lenses produce additional surface reflections which interfere with the eye tracker's identification and processing of the Purkinje image, subjects performed the experiment without corrective lenses. All subjects had an uncorrected Snellen visual acuity of 20/35 or better, as determined by a Bausch and Lomb Vision Tester (Cat. 71-22-41).

2.4. Procedure and design

After adjusting the chinrest and workstation, each subject was carefully calibrated. Calibration was also repeated prior to each block. Practice, consisting of a block of 33 trials, was provided for each of the tested interfaces. Each trial on a block was initiated by the subject selecting a "Continue" button at the center of the work area with the mouse. The "Continue" button was then

Table 1
Classification of eye movement and scanpath measures

	Temporal		Spatial
60 Hz gazeport samples	Duration		Length Convex hull area Spatial density
	Transition density		
Fixations	Number of fixations		
	Fixation/saccade ratio	Fixation duration	
Saccades			Number of saccades Saccade duration

immediately replaced by the name of one of the eleven randomized tool buttons (e.g., CUT) in the middle of the workspace. The eye tracker initiated its POR data collection at this time. The subject, as quickly as possible, then located the tool button from the tool menu at the left of the display, and clicked the left button on the mouse, stopping POR collection. Feedback, consisting of a statement of “correct” or “incorrect” at the position of the initial instruction location, was provided after each trial. A 1 min break was provided between each block; the total subject testing time was 40 min.

Within each of 12 subjects, the experiment presented 6 replicates of each of the 11 tool button components for each of the two interfaces presented here. The trial order was counterbalanced between subjects. A fully-crossed ANOVA for each dependent measure included Subjects (12 levels, random effect) $\times$ Interface (2 levels, fixed effect) $\times$ Tool Component (11 levels, fixed effect) $\times$ 6 replicates).

3. Scanpath generation

3.1. Classification of measures

Scanpaths are defined by a saccade–fixate–saccade sequence on a display. For information search tasks, the optimal scanpath is a straight line to

a desired target, with relatively short fixation duration at the target. The derived scanpath measures discussed below attempt to quantitatively measure the divergence from this optimal scanpath in several ways. The measures each provide a single quantitative value, with some requiring no knowledge of the content of the computer interface. Table 1 provides a summary of these measures, categorizing them on two dimensions. Temporal measures describe the sequential, time-based nature of a scanpath, whereas spatial emphasizes the spread and coverage of a scanpath. Furthermore, the measures may rely upon unprocessed, 60 Hz raw gazeport samples, or may be more oriented to processed fixations and/or saccades within a scanpath. Typically, reported eye movement data has been pre-processed to form fixations and saccades, by one of many different algorithms (Goldberg and Schryver, 1993). The resulting set of fixations and saccades are further processed to characterize scanpaths and their dynamic change (Goldberg and Schryver, 1995). However, some of the measures presented here can be applied to the gazeport samples, which is computationally easier, but with less behavioral meaning.

3.2. Fixations

The eyes dart from fixation to fixation in a typical search on a display. At least three processes take

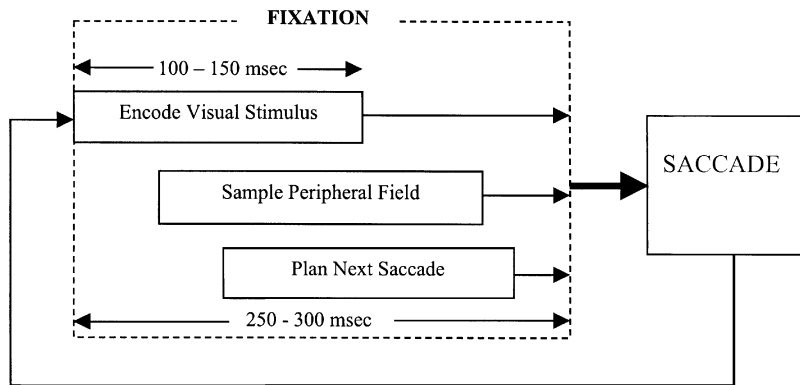


Fig. 3. Events occurring within typical fixations.

place within the 250–300 ms of a typical fixation (Viviani, 1990), as shown in Fig. 3. First, visual information is encoded, presumably to label the general scene (Loftus and Mackworth, 1978). Next the peripheral visual field of the current gaze is sampled, to determine subsequent informative areas. Finally, the next saccade is planned and prepared. These processes overlap, and may occur in parallel.

Gazepoints sampled at 60 Hz represent the line-of-sight at the time of sampling, and may or may not be at a location within a fixation, as sampling may have occurred during a saccade or perhaps during a blink or other artifact. Most commercial eye tracking systems include software removal of these artifacts, and some also include fixation construction algorithms. Fixation algorithms may be based on cluster and other statistical analyses, and may be locally adaptive to the amplitude of ocular jumps (Goldberg and Schryver, 1995; Ramakrishna et al., 1993; Belofsky and Lyon, 1988; Scinto and Barnette, 1986). Most algorithms develop fixation clusters by using a constrained spatial proximity determination, but temporal constraints can also be used. Latimer (1988) used temporal information related to each sample gazepoint but only to determine the cumulative fixation time after the cluster had been defined by spatial criteria. A fixation algorithm must produce fixations that meet certain minimum characteristics. The center of a typical fixation is within 2–3° from the observed target object (Robinson, 1979) and the minimum process-

ing duration during a fixation is 100–150 ms (Viviani, 1990).

The present study used a data position variance method (Anliker, 1976), after removing blinks and other eye movement artifacts. Fixations were initially constrained to a $3^\circ \pm 0.5^\circ$ spatial area, and had to be of at least 100 ms in duration. This corresponded to a minimum of 6 sample gazepoints per fixation (at 60 Hz), following Karsh and Breitenbach (1983), and agrees with descriptions of saccades lasting 20–100 ms (Hallet, 1986). Once a fixation was initially defined, its spatial diameter was computed. Subsequent gazepoint data samples falling within this diameter threshold were interactively added to the fixation. The spatial diameter threshold was then raised or lowered within a subject, following the method of Krose and Burbeck (1989), with only one fixation diameter allowed per subject. Maximum fixation diameters were varied from 2° to 4°, in 0.5° increments, until defined fixations sufficiently fit the gazepoint data. Allowed fixation diameters were increased when too few fixations (of very short duration) were evident. Conversely, fixation durations longer than 900 ms indicated that fixation diameter should be decreased.

While these methods are useful in identifying critical areas of attentional focus on a display, information based on the temporal order of fixations is lost. When and how often a target is fixated during a scanpath provides valuable information for the evaluation of an interface. Fig. 4 illustrates

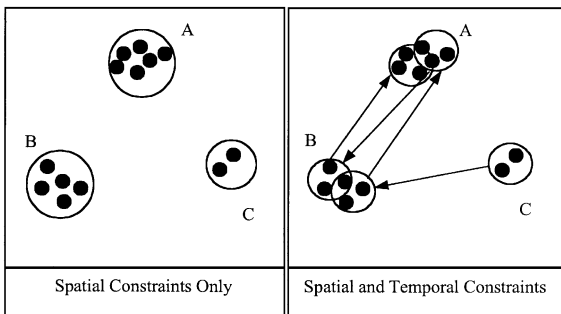


Fig. 4. Comparison of fixation algorithms. Spatial constraints (left panel); spatial plus temporal constraints (right panel).

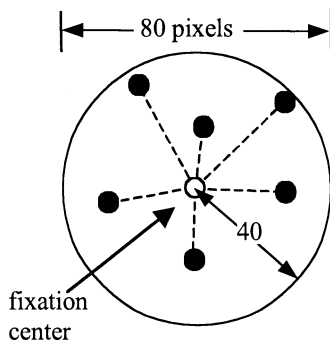


Fig. 5. Fixation cluster definition, showing 80 pixel diameter.

the difference between spatially derived fixations and fixations derived on the basis of both spatial and temporal criteria. Areas *A* and *B* are areas of high interest due to the number of gazepoint samples at each location. The left panel shows a temporal independent clustering (spatial constraint only), whereas the right panel shows a temporal-sensitive clustering. Areas *A* and *B* are still shown as areas of high interest, but by keeping track of the temporal order of samples, better information about the relationship between *A* and *B* are obtained.

The present study supplemented the preceding fixation method by testing sampled gazepoints in temporal order (Latimer, 1988; Tullis, 1983). Each of the 6 or more (defining at least 100 ms, at 60 Hz) temporally sequential gazepoint samples had to be within 40 pixels (0.6 in or 1.3°) from the centroid of the gazepoint sample, as shown in Fig. 5. This

Table 2
Algorithm for fixation clustering

<i>Fixation cluster algorithm</i>	
Step 1	Place first node in current cluster
Step 2	Compute common mean location of all samples in current cluster, and the next temporally sequential sample
Step 3	If the new point is within 40 pixels from the common mean location include new point in the current cluster If the new point is not within 40 pixels of the common mean then the current cluster becomes an old cluster and the new point becomes the current cluster If the number of points (<i>n</i>) in the old cluster ≥ 6 then the cluster is classified as a FIX-ATION of $n \times 16.67$ ms duration If $n < 6$ then the cluster is classified as SACCADE of $n \times 16.67$ ms duration GOTO Step 2 until done

defined fixations that were within the 2–3° range described by Robinson (1979). If the total number of samples within a cluster was less than 6, then the cluster was categorized as part of a saccade. The general fixation algorithm applied to the present data is in Table 2.

4. Measures of search

Illustrated descriptions of each of the eye movement measures and algorithms are provided below. Results from the good versus poor interfaces and other factors are also presented here. The same hypothetical scanpath is used in all examples below, for easy comparison. All of these measures may be used for a given scanpath, with each offering a slightly different interpretation of the data. Scanpaths may also be viewed as directed or undirected graphs, allowing additional characterizations of complexity and size from graph theory. The organized functional grouping of components in the good design was expected induce subjects to find components quickly, producing rapid direct search patterns. In contrast, the randomized groups of the poor design were intended to mislead subjects, causing them to stay in an incorrect grouping

or leave a correct grouping under the incorrect expectation that grouped components were related functionally. As a result the poor design was expected to produce more extensive search behavior.

4.1. Scanpath length and duration

Scanpath length is a productivity measure that can be used for baseline comparisons or defining an optimal visual search based on minimizing saccadic amplitudes. This may be computed independently from the actual screen layout, and may be applied to gaze point samples or to processed fixation data. The length (in pixels) is the summation of the distances between the gaze point samples. An example scanpath is illustrated in Fig. 6. Lengthy scanpaths indicate less efficient scanning behavior but do not distinguish between search and information processing times. Unless scanpaths are formed from computed fixations and saccades, the scanpaths should not be used to make detailed inferences about one's attentional allocation on a display.

Scanpath duration is more related to processing complexity than to visual search efficiency, as much more relative time is spent in fixations than in saccades. Using 60 Hz gaze point samples, the number of samples is directly proportional to the temporal duration of each scanpath, or Scanpath Duration = $n \times 16.67$ ms, where n = number of samples in the scanpath. However, using fixations, the scanpath duration must sum fixation durations with saccade durations.

Using gaze point samples, there was no significant difference in the overall duration of scanpaths produced by the good and poor interfaces ($F_{1,1320} = 1.90$, $p > 0.05$). The average duration from the good interface was 1439 ms ($sd = 368.4$), while the poor interface produced average durations of 1543 ms ($sd = 566.5$). The non-significant duration here, possibly due to (non-significant) variance differences, should not be interpreted along as a sign of similar interface quality. Further measure comparisons should be conducted.

Extensive search behavior produces spatially lengthy scanpaths. Two fixation-saccade scanpaths may have the same temporal duration but considerably different lengths due to the differences in the extent of search required. Using the summated

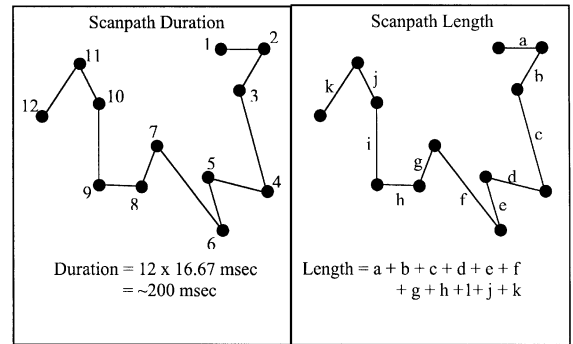


Fig. 6. Example computations for scanpath duration (left panel) and length (right panel).

lengths (by the pythagorean theorem), scanpath lengths were computed, as in Fig. 6. The poor design did indeed produce longer scanpaths ($F_{1,1320} = 5.16$, $p < 0.05$), averaging 228 pixels ($sd = 860$), 14% longer than the better interface (1978 pixels, $sd = 491$). Note that 65 pixels, here, was equivalent to a screen distance of 2.5 cm.

4.2. Convex hull area

Circumscribing the entire scanpath extends the length measures to consider the area covered by a scanpath. If a circle circumscribed the scanpath, small deviations in gaze point samples would lead to dramatic changes in the area of the circumscribed circle, exaggerating actual differences in the scanpath area. As shown in Fig. 7, left panel, scanpaths *A* and *B* are similar, differing by only one excursion, but the area of the circle circumscribing scanpath *B* is 4 times the area of the circle circumscribing scanpath *A*. In contrast, scanpaths *B* and *C* are dramatically different in shape and range but produce the same circumscribed circle area.

Using the area of the convex hull circumscribing the scanpath, illustrated in Fig. 7, right panel, the exaggeration can be reduced. Note that scanpath areas *A* and *B* are now more similar, and *B* and *C* are less similar than with circumscribed circles. Table 3 provides an algorithm to construct convex hulls and hull area, of which steps 1–4 are illustrated in Fig. 8. Fig. 9 provides a simple example of a scanpath convex hull area. Alternative algorithms

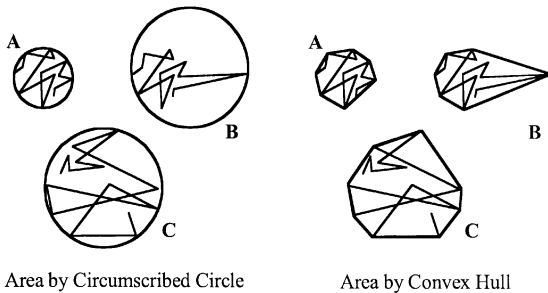


Fig. 7. Relative comparison of areas defined by circumscribed circles (left panel) and convex hulls (right panel).

for generating convex hulls are provided by Sedgewick (1990). Triangle areas (e.g., ΔABC) were computed from:

$$\Delta ABD = \sqrt{P(P - \overline{AB})(P - \overline{BC})(P - \overline{CA})},$$

where the perimeter,

$$P = (\overline{AB} + \overline{BC} + \overline{CA})/2$$

and

$$IJ = \sqrt{|X_I - X_J|^2 + |Y_I - Y_J|^2}.$$

While the convex hull area may seem to be a more comprehensive measure of search than the scanpath length, note that long scanpaths may still reside within a small spatial area. Used in conjunction, the two measures can determine if lengthy search covered a large or a localized area on a

Table 3

Algorithm for convex hull area

Step 1	Search all samples to identify and label the four samples with the Min x, Max y, Max x and Min y
Step 2	Set Min x as Vertex(1)
Step 3	Compute the slope of Vertex(n) with every sample in the scanpath
Step 4	IF {Min x < Vertex(n) Max y} OR {Min y < Vertex(n) < Max x} THEN set Vertex(n) to the sample with the largest positive slope IF {Max y < Vertex(n) < Max x} OR {Min y < Vertex(n) < Min x} THEN set Vertex(n) to the sample with the least negative slope Store Vertex(n) in a list IF Vertex(n) = Min x THEN GOTO Step 5 Increment n and GOTO Step 3
Step 5	Set n = 2
Step 6	Compute and store the area of the triangle created by the points Vertex(1), Vertex(n) and Vertex(n + 1). Increment n by 1. Repeat Step 6 until done. Sum of stored areas equals convex hull area

display. In the present experiment, the poor design produced 11% larger (31 339 pixel², sd = 14 952) search areas than the better design (28 168 pixel², sd = 12 009, $F_{1,1320} = 6.70$, $p < 0.05$). The larger search area coupled with the longer scanpath produced by the poorer interface indicated that the disorganized interface produced a widely distributed search pattern.

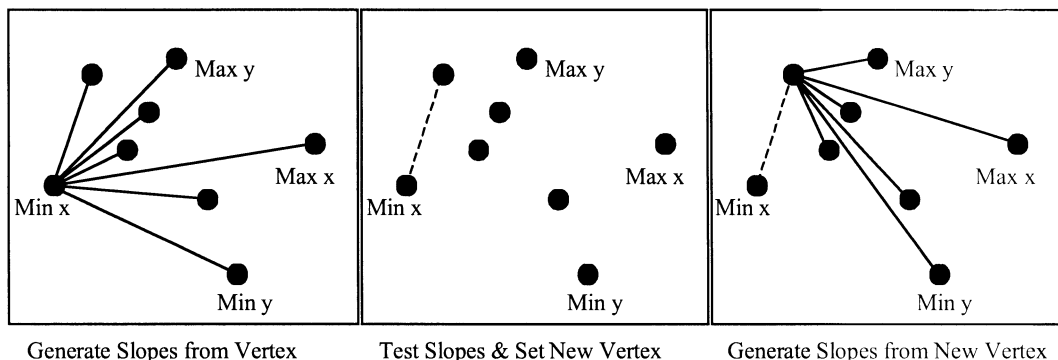


Fig. 8. Iterative example of convex hull generation algorithm.

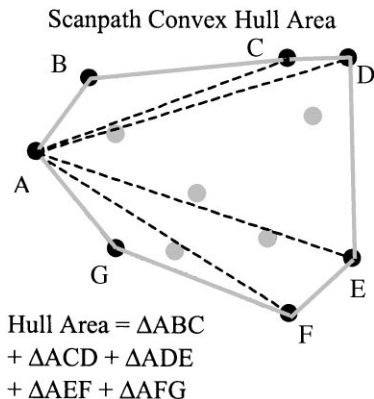


Fig. 9. Example area generated by convex hull algorithm.

4.3. Spatial density

Coverage of an interface due to search and processing may be captured by the spatial distribution of gaze point samples. Evenly spread samples throughout the display indicate extensive search with an inefficient path, whereas targeted samples in a small area reflect direct and efficient search. The interface can be divided into grid areas either representing specific objects or physical screen area. In the present experiment, the display was divided into an evenly spaced 10×10 grid, with each cell covering 64×48 pixels. The spatial density index was equal to the number of cells containing at least one sample, divided by the total number of grid cells (100); an example is shown in Fig. 10. A smaller spatial density indicated more directed search, regardless of the temporal gaze point sampling order. The poor interface produced 7% larger spatial density indices (Mean index = 10.2, sd = 3.1%) compared with the better interface (Mean index = 9.5%, sd = 2.3%, $F_{1,1320} = 6.31$, $p < 0.05$).

4.4. Transition matrix

A transition matrix expresses the frequency of eye movement transitions between defined Areas of Interest (AOI's) (Ponsoda et al., 1995). This metric considers both search area and movement over time. While the scanpath spatial density provides useful information about the physical range of

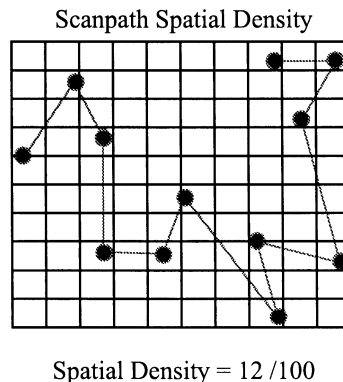


Fig. 10. Example spatial density computation.

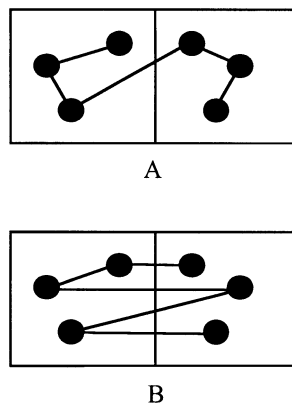


Fig. 11. Relative scanpath differences between efficient (A) and inefficient (B) search.

search, a transition matrix adds the temporal component of search. Also known as link analysis (Jones et al., 1949), frequent transitions from one region of a display to another indicates inefficient scanning with extensive search. Consider the two simple spatial distributions presented in Fig. 11. Both distributions produce the same index of spatial density and convex hull area. However, the search behaviors are dramatically different. Scanpath A has a more efficient search pattern with a shorter scanpath length than scanpath B.

The transition matrix is a tabular representation of the number of transitions to and from each defined area. As shown in Fig. 12, a directed scanpath from region 3 to region 5 forms a unique cell

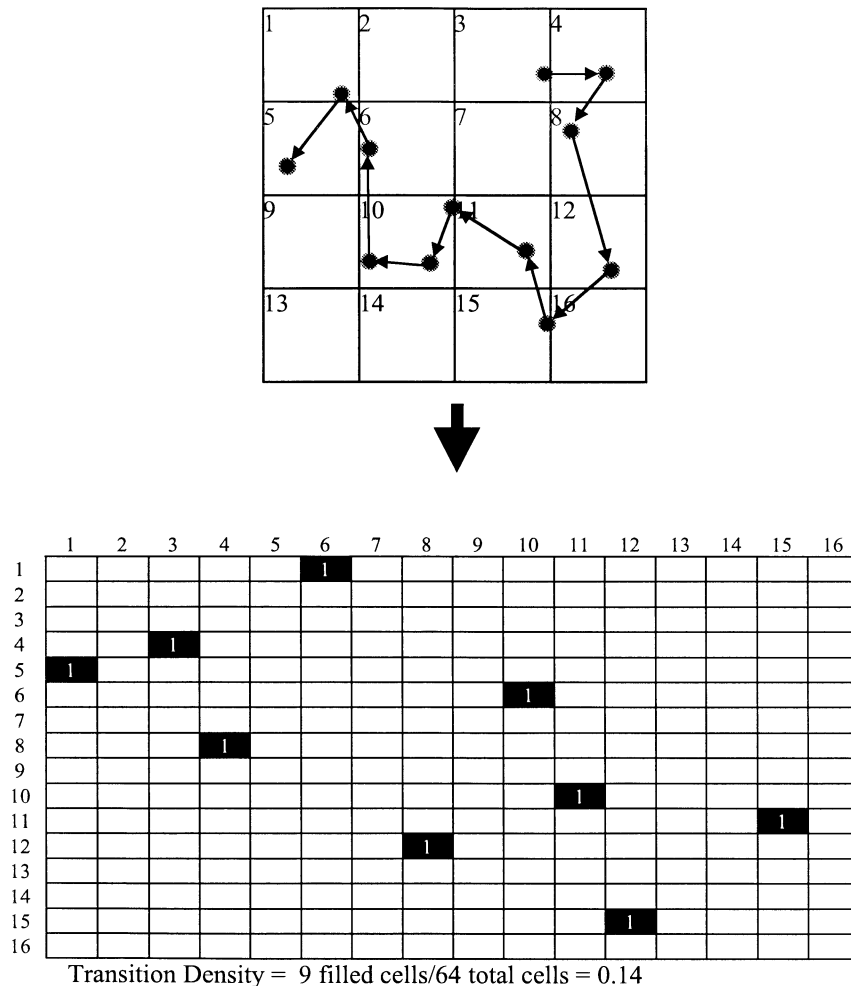


Fig. 12. Example development of transition matrix from areas of interest on display.

pattern in the transition matrix. An unusually dense transition matrix, with most cells filled with at least one transition, indicates extensive search on a display, suggesting poor design. A sparse matrix indicates more efficient and directed search. The matrix may be characterized with a single quantitative value by dividing the number of active transition cells (i.e., those containing at least one transition) by the total number of cells. A large index value indicates a dispersed, lengthy, and wandering scanpath, whereas smaller values point to more directed and efficient search.

The defined AOI's may be of equal or unequal size. A content-dependent analysis would assign

each AOI to a screen window or object, with a unique AOI expressing all non-interesting areas. A content-independent analysis would simply divide the display into a grid, assigning an AOI to each grid cell. The present experiment divided the display interface into 25 regions; 24 were of equal size, whereas the 25th was the larger workspace area. In order to better capture dynamic search activity within the scanpath, intra-cell transitions need not be included (these were not shown in Fig. 12). The transition matrix density is the number of non-zero matrix cells divided by total number of cells (25×25 cells here). The poor interface had denser (1.69%, $sd = 1.01$) transition

matrices than the better interface (1.37%, $sd = 0.65$, $F_{1,1320} = 6.91$, $p < 0.05$), consistent with the undirected and more extensive search behavior expected due to the grouping of unrelated components in the poorer interface design.

4.5. Number of saccades

The number of saccades in a scanpath indicates the relative organization and amount of visual search on a display, with more saccades implying greater amount of search. For applied purposes, the distance between each successive fixation can generally be defined as a saccade, with both an amplitude and duration. That is, the number of saccades may be defined from the number of fixations minus one. A minimum amplitude of 160 pixels (5.3° visual angle) was required here to filter out small micro-saccades resulting from moving to the periphery of the prior fixation (only 80 pixels). Saccades may be quite large (e.g., 20°), so no upper bound was placed on the saccadic amplitude. The overall algorithm for counting saccades thus first tested the distance between fixation centers, incrementing the saccade count if greater than 160 pixels. Prior to acquiring the tool button target, there were 17% more saccades (averaging 2.53 saccades, $sd = 1.47$) produced from the poor interface than from the better interface (2.17 saccades, $sd = 1.16$, $F_{1,1320} = 8.74$, $p < 0.05$).

4.6. Saccadic amplitude

A well designed interface should provide sufficient cues to direct the user's scanning to desired targets very rapidly, with few interim fixations. This will result in an expectation of larger saccadic amplitudes. If the provided cues are not meaningful or misleading, the resultant saccades should be smaller, negotiating the interface until a meaningful cue appears. The average saccadic amplitude is computed from the sum of the distances between consecutive fixations, dividing this by the number of fixations minus one. Note that all saccades were used to generate this sum, with no minimum length criterion. There was no significant difference in average saccadic amplitude between the two interface designs ($F_{1,1320} = 0.22$, $p > 0.05$). This

indicated that even with more extensive search in poorer interfaces, the size of the saccades was similar between good (303 pixels, $sd = 109$) and poor (299 pixels, $sd = 104$) interfaces. While the local search step size was the same between the two interfaces, the overall extent of search was greater in the poor design. The functional grouping layout thus aids visual search planning for proper tool selection, but does not impact individual saccadic motions.

In both interfaces, subjects typically moved to one group and sampled a component. In the good interface, a rapid determination could be made to determine if the desired component was in the same group (common function) or not (different function). If the group functionality matched the desired component, small saccades could be made to each component from within the tool grouping until the target was acquired. If the group does not match, a saccade is rapidly made to another group. The functional grouping layout, however, did not provide cues about the next group to sample, thus a small saccade was still made to an adjacent group, continuing the search. The poorer interface provided little or no information about the other components within a grouping, once a tool button was acquired. As a result, subjects again made small local saccades, exhaustively searching within the group before executing a saccade to an adjacent group.

Using these search measures, it was clear that the functional component grouping reduces the extent of required visual search by allowing one to rapidly "zoom in" on the desired component, while maintaining a relatively small amplitude for saccades.

5. Measures of processing

Visual search is conducted to obtain information from an interface, where more extensive search allows more interface objects to be processed. This does not consider the depth of required processing, however. In the present study, as the same representations were used in both interfaces, the depth of processing required to distinguish and interpret a component was not expected to differ.

5.1. Number of fixations

The number of fixations is related to the number of components that the user is required to process, but not the depth of required processing. When searching for a single target, a large number of fixations indicates the user sampled many other objects prior to selecting the target, as if distracted or hindered from isolating the target. The poor interface, intentionally designed to mislead the subject, produced significantly more fixations than the good design ($F_{1,1320} = 8.36$, $p < 0.05$). The poor interface produced on average 2.53 fixations ($sd = 1.16$) for each component search, 17% more fixations than the good interface required (average = 2.17, $sd = 1.47$). The functionally grouped design allowed one to “zoom in” on the correct component more efficiently, requiring fewer components to be processed.

5.2. Fixation duration

Longer fixations imply the user is spending more time interpreting or relating the component representations in the interface to internalized representations. Representations that require long fixations are not as meaningful to the user as those with shorter fixation durations. Maximum and average fixation times are context-independent measures, but the duration of single fixations on targets is dependent on the interface layout. Average fixation duration was calculated by summing the number of gaze point samples in all the fixations and dividing by the number of fixations.

The level of processing for graphic representations was expected to be the same between interfaces, as the icons were the same. Confirming this, the good interface average fixation durations (411 ms, $sd = 144$) and the poor interface fixations (391 ms, $sd = 144$) did not significantly differ ($F_{1,1320} = 1.92$, $p > 0.05$).

5.3. Fixation/saccade ratio

This content-independent ratio compares the time spent processing (fixations) component representations to the time spent searching (saccades) for the components. Interfaces resulting in higher

ratios indicate that there was either more processing or less search activity than interfaces with lower ratios. Other measures can determine which of these was the case. As the fixation/saccade ratio did not significantly differ between good (mean = 14.8, $sd = 5.9$) and poor (mean = 13.9, $sd = 6.2$) interfaces ($F_{1,1320} = 1.26$, $p > 0.05$), if more search was required, a proportionate amount of processing was also required.

5.4. Other measures

The preceding measures only describe a portion of the potential universe of eye movement and scanpath characterization tools. Other measures may further aid interface analysis in certain circumstances. Several examples may be considered: First, a *backtrack* can be described by any saccadic motion that deviates more than 90° in angle from its immediately preceding saccade. These acute angles indicate rapid changes in direction, due to changes in goals and mismatch between users' expectation and the observed interface layout. Second, the ratio of *on-target: all-target* fixations can be defined by counting the number of fixations falling within a designated AOI or target, then dividing by the all fixations. This is a content-dependent efficiency measure of search, with smaller ratios indicating lower efficiency. Third, the number of *post-target fixations*, or fixations on other areas, following target capture, can indicate the target's meaningfulness to a user. High values of non-target checking, following initial target capture indicate target representations with poor meaningfulness or visibility. Fourth, measures of *scanpath regularity*, considering integrated error or deviation from a regular cycle, can indicate variance in search due to a poor interface or users' state of training. Many potential measures of scanpath complexity are possible, once cyclic scanning behavior is identified.

6. Discussion

Successful interaction with a computer clearly requires many elements, including good visibility, meaningfulness, transparency, and the requirement

of simple motor skills. Eye movement-based evaluation of the interface, as espoused here, can only address a subset of critical interface issues that revolve around software object visibility, meaningfulness, and placement. Though not a panacea tool for design evaluation, characterization of eye movements can help by providing easily comparable quantitative metrics for objective design iteration.

One particular strength of eye movement-based evaluation is in the assessment of users' strategies at the interface. Usually unaware of their search processes, eye movements can provide a temporal/spatial record of search and flow while using a computer. Strategy differences are most evident during lengthy (e.g., 10–15 s) tasks, where a sufficient scan-path exists for characterization by the above methods. Task that are too rapid (e.g., under 1 s) do not allow sufficient data for many of the measures.

In the case of component grouping, the present study demonstrated very good validity between users' and designers' ratings of good versus poor, and many of the eye movement-based measures. The framework and measures proposed here allow improved objective estimation of users' strategies, and the influence of interface design on those strategies. Compared with a randomly organized set of component buttons, well-organized functional grouping resulted in shorter scanpaths, covering smaller areas. The poorer interface resulted in less directed search with more (though equal in amplitude) saccades. Though similar in duration, the poor interface resulted in more fixations than the better interface. Whereas the poor interface produces less efficient search behavior, the layout of component representations did not influence their interpretability.

Ongoing investigations will further consider the sensitivity, reliability, and validity of eye movement-based measures for interface evaluation. By presenting several designed interfaces that more continuously vary in rated quality, assessment of which measures reflect these quality differences can be made. Other factors are also being introduced, such as component meaningfulness and visibility. Ultimately, eye movements may lend some degree of diagnosticity to interface evaluations, and may possibly lead to design recommendations, similar to Tullis's (1983) methodology for text-based material.

References

- Anliker, J., 1976. On line measurements, analysis and control. In: Monty, R.A., Senders, J.W. (Eds.), *Eye Movements and Psychological Processes*. Erlbaum Press, Hillsdale, NJ.
- Belofsky, M.S., Lyon, D.R., 1988. Modeling eye movement sequences using conceptual clustering techniques. Air Force Human Resources Laboratory, Doc. # AFHRL-TR-88-16, Air Force Systems, Brooks Air Force Base, TX.
- Buswell, G.T., 1935. *How People Look at Pictures. A Study of the Psychology of Perception in Art*. The University of Chicago Press, Chicago, IL.
- Dix, A., Finlay, J., Abowd, G., Beale, R., 1988. *Human-Computer Interaction*. 2nd ed., Prentice-Hall, London.
- Goldberg, J.H., Schryver, J.C., 1993. Eye-gaze determination of user intent at the computer interface. In: Findlay, J.M., Walker, R., Kentridge, R.W. (Eds.), *Eye Movement Research: Mechanisms, Processes and Applications*. North-Holland Press, Amsterdam, pp. 491–502.
- Goldberg, J.H., Schryver, J.C., 1995. Eye-gaze contingent control of the computer interface: Methodology and example for zoom detection. *Behavior Research Methods, Instruments and Computers* 27 (3), 338–350.
- Jones, R.E., Milton, J.L., Fitts, P.M., 1949. Eye fixations of aircraft pilots; IV: Frequency, duration and sequence of fixations during routine instrument flight, US Air Force Technical Report 5975.
- Just, M.A., Carpenter, P.A., 1976. Eye fixations and cognitive processes. *Cognitive Psychology* 8, 441–480.
- Kolers, P.A., Duchnick, R.L., Ferguson, D.C., 1981. Eye movement measurement of readability of CRT displays. *Human Factors* 23 (5), 517–527.
- Krose, B.J.A., Burbeck, C.A., 1989. Spatial interactions in rapid pattern discrimination. *Spatial Vision* 4, 211–222.
- Latimer, C.R., 1988. Eye-movement data: Cumulative fixation time and cluster analysis. *Behavior Research Methods, Instruments, and Computers* 20 (5), 437–470.
- Loftus, G.R., Mackworth, N.H., 1978. Cognitive determinants of fixation location during picture viewing. *Journal of Experimental Psychology: Human Perception and Performance* 4 (4), 565–572.
- Mackworth, N.H., 1976. Stimulus density limits the useful field of view. In: Monty, R.A., Senders, J.W. (Eds.), *Eye Movements and Psychological Processes*. Erlbaum, Hillsdale, NJ.
- Ponsoda, V., Scott, D., Findlay, J.M., 1995. A probability vector and transition matrix analysis of eye movements during visual search. *Acta Psychologica* 88, 167–185.
- Ramakrishna, S., Pillalamarri, B., Barnette, D., Birkmire, D., Karsh, R., 1993. Cluster: A program for the identification of eye-fixation-cluster characteristics. *Behavior Research Methods, Instruments, and Computers* 25 (1), 9–15.
- Robinson, G.H., 1979. Dynamics of the eye and head during movement between displays: A qualitative and quantitative guide for designers. *Human Factors* 21 (3), 343–352.
- Scinto, L., Barnette, B.D., 1986. An algorithm for determining clusters, pairs and singletons in eye-movement scan-path

- records. *Behavior Research Methods, Instruments, and Computers* 18 (1), 41–44.
- Sedgewick, R., 1990. *Algorithms in C*. Addison-Wesley, Reading, MA.
- Tullis, T.S., 1983. The formatting of alphanumeric displays: Review and analysis. *Human Factors* 25 (6), 657–682.
- Viviani, P., 1990. In: Kowler, E. (Ed.), *Eye Movements and Their Role in Visual and Cognitive Processes*, Ch. 8. Elsevier Science, Amsterdam.
- Wickens, C.D., Carswell, C.M., 1995. The proximity compatibility principle: Its psychological foundation and relevance to display design. *Human Factors* 37 (3), 473–494.
- Wickens, C.D., Gordon, S.E., Liu, Y., 1998. *An Introduction to Human Factors Engineering*. Addison-Wesley and Longman, New York.
- Yamamoto, S., Kuto, Y., 1992. A method of evaluating VDT screen layout by eye movement analysis. *Ergonomics* 35 (5/6), 591–606.