



**AIIM**

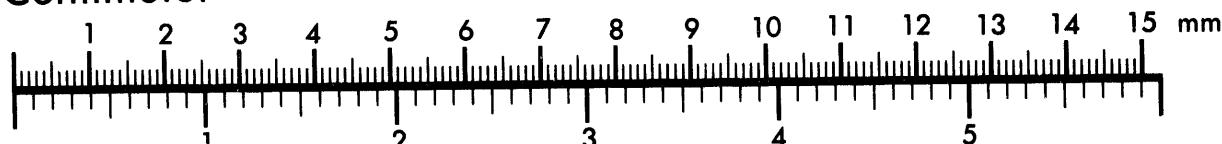
**Association for Information and Image Management**

1100 Wayne Avenue, Suite 1100

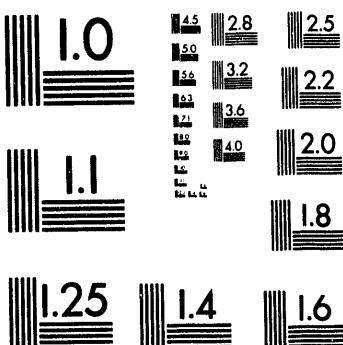
Silver Spring, Maryland 20910

301/587-8202

**Centimeter**



**Inches**



MANUFACTURED TO AIIM STANDARDS  
BY APPLIED IMAGE, INC.

1 of 1

Conf-9308228--1

## Engineering Physics and Mathematics Division

### EYE-GAZE DETERMINATION OF USER INTENT AT THE COMPUTER INTERFACE\*

#### DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

J. H. Goldberg  
Department of Industrial Engineering  
The Pennsylvania State University  
207 Hammond Bldg.  
University Park, PA 16802

J. C. Schryver  
Cognitive Systems and Human Factors Group  
Oak Ridge National Laboratory  
P. O. Box 2008  
Oak Ridge, TN 37831-6360

"The submitted manuscript has been authorized by a contractor of the U.S. Government under contract No. DE-AC05-84OR21400. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so for U.S. Government purposes."

To be published in the Proceedings of the Seventh European Eye Movement Conference

---

\* Research sponsored by the Office of Technology Support Programs, U.S. Department of Energy, under Contract No. DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc., and U. S. DOE and Oak Ridge Associated Universities, under Contract No. DE-AC05-76OR00033.

MASTER *do*  
DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

# EYE-GAZE DETERMINATION OF USER INTENT AT THE COMPUTER INTERFACE

*Joseph H. Goldberg<sup>a</sup> and Jack C. Schryver<sup>b</sup>*

<sup>a</sup>Department of Industrial Engineering, The Pennsylvania State University,  
207 Hammond Building, University Park, PA 16802 USA

<sup>b</sup>Cognitive Systems & Human Factors Group, Engineering Physics & Mathematics  
Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6360 USA

Determination of user intent at the computer interface through eye-gaze monitoring can significantly aid applications for the disabled, as well as telerobotics and process control interfaces. Whereas current eye-gaze control applications are limited to object selection and x/y gazepoint tracking, a methodology was developed here to discriminate a more abstract interface operation: zooming-in or out. This methodology first collects samples of eye-gaze location looking at controlled stimuli, at 30 Hz, just prior to a user's decision to zoom. The sample is broken into data frames, or temporal snapshots. Within a data frame, all spatial samples are connected into a minimum spanning tree, then clustered, according to user defined parameters. Each cluster is mapped to one in the prior data frame, and statistics are computed from each cluster. These characteristics include cluster size, position, and pupil size. A multiple discriminant analysis uses these statistics both within and between data frames to formulate optimal rules for assigning the observations into zoom-in, zoom-out, or no zoom conditions. The statistical procedure effectively generates heuristics for future assignments, based upon these variables. Future work will enhance the accuracy and precision of the modeling technique, and will empirically test users in controlled experiments.

**Keywords:** Eye-Gaze, Eye Movements, Computer Control, Cluster Analysis,  
Discriminant Analysis, User Intent, Gaze-Contingent Control

## User Intent and Eye-Gaze

User intent discrimination is an essential feature of truly adaptive user interfaces. When implemented correctly, intent discrimination generates interface transparency, i.e., the user does not notice the workings of the interface and is not encumbered with the need to input control actions in correct syntax. Application software that infers user intent does not rely on traditional input "devices." Rather, it uses all available information to discover what the user is trying to accomplish at the interface. Eye-gaze is a rich source of information regarding user intent, as evidenced by almost 20 years of research demonstrating the relationship between cognitive processing and eye-gaze position (Just and Carpenter, 1976).

## Computer Interface Control

Control of the computer interface from a user's eye-gaze location is an application of user intent discrimination. Prior reports of eye-gaze control have

been limited to using eye-gaze location as an analog to concrete mouse operations such as cursor control, object selection, or drag and drop. Applications have been developed for word processing (Gips, et al., 1993; Frey, et al., 1990), selecting menu items (Hutchinson, et al., 1989), information disclosure in fictional worlds (Starker and Bolt, 1990) and object selection and movement in a tactical decision-making simulation (Jacob, 1991).

Mouse substitution may not be effective for eye-gaze control of more abstract operations such as interface zooming. Such operations are likely driven by more covert visual attention or cognitive processes. Uncovering overt patterns among otherwise covert operations represents a challenge to the eye-gaze investigator. An interface control heuristic formed from these overt patterns is essentially a marker of user intent, if gathered prior to the execution of a desired operation.

Automated discrimination of user intent for some mundane tasks can potentially free up human operator resources for more important decision making. While user intent analysis can occur at many levels (e.g., intent determination from physical positions or motions, or from speech or language patterns), analysis of eye-gaze patterns were chosen here for several reasons:

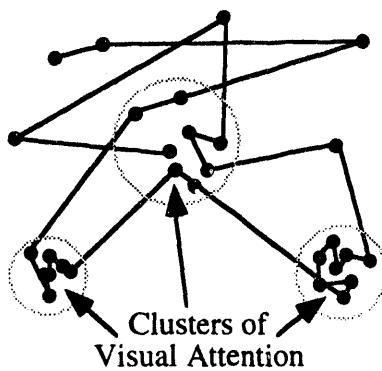
- Eye-gaze can be non-invasively monitored while using a computer.
- Based upon clinical psychology techniques, eye-gaze may possibly provide significant information on user intent.
- A small literature on using eye-gaze for computer interface control exists.
- Eye-gaze location, a spatial mechanism, should have a natural compatibility with control of spatial devices, such as cameras or robots.
- If natural eye-gaze tendencies are used, eye-gaze based control should prove relatively automatic and attention-free.

### **Spatial Clusters of Attention**

Spatial clusters of eye-gaze locations signal important loci of visual attention. Fig. 1 shows a scanpath with spatial clusters circled. The clusters may be formed by a serial connected set of fixations (lower right-hand cluster in Fig. 1), or may be formed from a series of refixations within a spatial area (upper cluster in Fig. 1). The formation of spatial clusters from time-separated fixations implies that cluster formation methods must inherently neglect scanpath analysis methodologies. Cluster formation methods from eye-gaze have been presented in prior papers (Pillalamarri, et al., 1993; Latimer, 1988; Belofsky and Lyon, 1988; Scinto and Barnette, 1986). Thus, independent consideration of eye-gaze location on a display is insufficient to understand visual attention loci, without an accompanying cluster analysis. Interpretation of user intent, if based upon visual attention patterns, should primarily consider the location and characteristics (e.g., size, shape) of eye movement clusters.

Analysis of clusters of eye movement or visual attention locations may be static or quite dynamic. In the latter case, clusters may expand or contract, and possibly spawn additional clusters as visual attention becomes more distributed over an area. If clusters are composed of at least 3-4 fixations, then the time represented by a cluster should be at least one second. Analysis of several clusters should thus require several seconds. A clustering scheme based solely on fixation locations will be quite slow and nonresponsive. The present approach sidesteps this difficulty by

neglecting the concept of fixations altogether. Instead, locations of eye-gaze point-of-regard are sampled at the maximum rate of the tracking system; successive samples may or may not be within the same fixation. Eye movements that occur within a fixation (e.g., microsaccades, ticks) could form multiple clusters, given input clustering criteria described below, if an adaptive clustering technique is used. Such clusters are treated and characterized just as clusters formed between fixations. At a 30 Hz eye movement sampling rate, up to 30 observations per second are provided by this methodology, greatly aiding the power of the cluster approach.



*Fig. 1. Spatial clusters of eye-gaze location, formed independently of serial scanpath.*

This paper describes an empirical methodology for construction of on-line algorithms which perform intent discrimination from eye-gaze characteristics. Artificial results are provided to illustrate the use of the methodology to construct decision rules for zooming in and out, or not zooming, at the computer interface.

## Methodology

The general procedure for user intent discrimination consisted of 6 sequential steps. While currently conducted off-line, the procedure has large potential for real-time analysis. The steps, described further below, were: (1) Eye-gaze sampling, (2) Data reduction of eye-gaze samples, (3) Spatial clustering of eye-gaze samples, (4) Characterization of spatial clusters, (5) Multiple discriminant analysis across spatial and temporal cluster characteristics, and (6) Visualization of significant variable relationships.

### Eye-Gaze Tracking Apparatus

An LC Technologies eye tracking system collected serial records at 30 Hz of eye-gaze screen location, pupil diameter and cumulative time. The system camera was mounted beneath a Sun Sparcstation 2 monitor, and pointed toward the user's right eye, which was at a distance of 50 cm. The eyetracking output data was transferred, via a host computer, to the Sun workstation for subsequent analysis. The system provided accurate records of serial eye-gaze location, based upon prior reference calibration. Using a chin rest to stabilize the head of the user, the average angular

bias error was less than  $0.5^\circ$ , with about 4 cm of head motion tolerance in the horizontal and vertical frontal planes. The system worked equally well with eyeglasses or contact lenses.

### Eye-Gaze Sampling

The example data reported here were collected during an experiment requiring users to view a display and make a decision whether to zoom-in or out to gather further information. The user performed the zoom, when necessary, by pressing one of two buttons on a mouse. The user's task was a same/different judgement between an initially memorized test stimulus, and a presented comparison stimulus probe. On each trial, eye-gaze samples were collected following the probe stimulus object presentation, and ended with the user's zoom response. Thus, the serial sample of eye-gaze locations preceded the motor zoom response. An example trial is provided in Fig. 2, showing relevant events. Fig. 2A shows a sample trial on which a zoom-in was required, while Fig. 2B shows a zoom-out trial. Two responses were actually required for each trial: an initial zoom response on the mouse, then a keyboard response of "s" (same) or "d" (different). Though this paper is a report on methodology, note that several different stimuli, randomly permuted, were actually used.

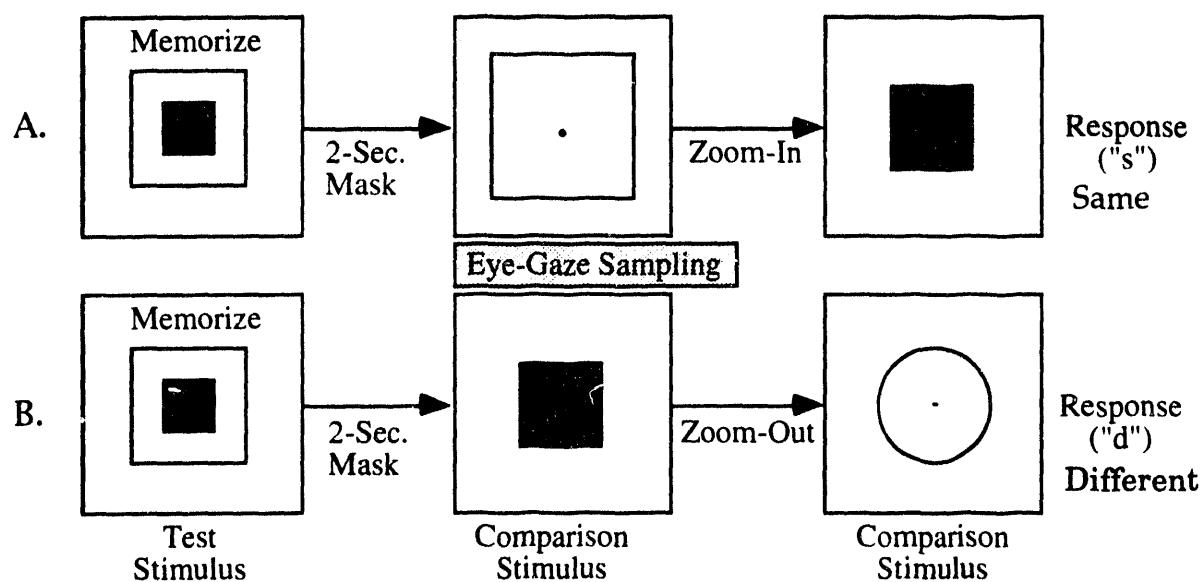


Fig. 2. Example trial events, shown within screen borders. Approximate eye-gaze sampling interval is also indicated; all sampling preceded actual zoom responses by the user. A. Zoom-in trial, showing stimulus border as the comparison stimulus. B. Zoom-out trial, showing stimulus interior as the comparison stimulus.

On a typical trial, the decision to zoom-in, zoom-out, or not to zoom required just under one second; about 20-30 30 Hz serial eye-gaze locations were collected during

this interval. An experiment of 100 trials thus recorded about 3000 records for subsequent, off-line analysis.

### Initial Data Reduction

Data reduction and analysis was conducted off-line for this project, although real-time methods are feasible extensions. Each of the serial eye-gaze locations (hereafter referred to as data samples) was represented by an X/Y location pair and a time stamp. The eye-gaze analyst controlled several aspects of data reduction via user inputs to the analysis software.

**Data Frames.** The data framing process was initially defined via user inputs. The user split up each trial into successive data frames, with defined overlap, by inputting the number of samples within each frame. All samples within each frame were, in essence, considered simultaneous. A frame size of one sample simply led to analysis of individual eye-gaze samples. A frame size as large as the number of collected samples (or larger) within a trial created one data frame for that trial. For analyses below, 2 to 4 data frames with 15 samples were formed, with overlaps of 5 samples per frame. Improved resolution was obtained by smaller frames, and improved stability or accuracy in spatial locations was obtained by larger frames.

**Minimum Spanning Trees.** All samples within a data frame were connected into a shared minimum spanning tree (MST) data representation. The data treatment here differs substantially from the fixation and scanpath analyses by others, which attempt to group successive samples into fixation locations via temporal and location-based heuristics. As these heuristics are rather arbitrary and not necessary for the present work, the actual sample locations were used.

An additional advantage of the MST representation over scanpath analysis concerns the identification of attentional foci. Scanpaths often loop back on themselves, visiting former locations. Identification of attentional foci must consider these refixations before dividing the screen into separate areas. The MST representation inherently considers refixations as additional samples, given a large enough sampling frame. By iteratively varying the size of data frames, optimal, converging evidence for number of attentional foci may be found. Note that all data within a data frame is considered to be simultaneous; the MST thus loses all sequential information within a data frame. Smaller data frames, with fewer observations, regain this temporal information, however, for subsequent analyses.

Minimum spanning trees were formed by Prim's algorithm (Camerini, et al., 1988), due to its rapid and efficient operation. Given a spatial array of eye-gaze locations, only one such MST can be formed. The MST has the beneficial property of no closed circuits, so it can be rapidly searched. In addition, it defines the minimum distance network that interconnects all locations or nodes. Starting from an arbitrary eye-gaze location, the minimum distance location was connected, creating the first edge of the MST. The next shortest distance to either of the two nodes of the tree was next connected. These connections continued until no more unconnected eye-gaze samples were present. Fig. 3 presents a set of eye-gaze samples interpreted in two different ways. Fig. 3A shows a classic temporal scanpath interpretation (assuming each sample is a fixation location, for present purposes of illustration). Fig. 3B shows the same data, interpreted as an MST. Note the lack of completed circuits in the latter representation, which enables cluster definition.

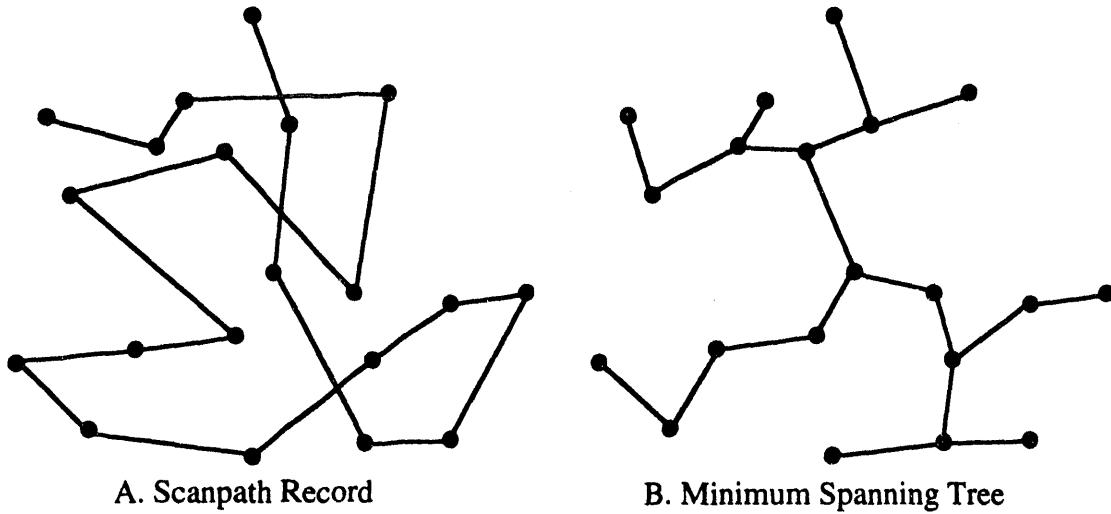


Fig. 3. Comparison of data representations for identical eye-gaze samples. A. Scanpath interpretation, temporally linking adjacent samples. B. Minimum spanning tree representation, easing cluster separation.

## Spatial Cluster Formation

Spatial MSTs were clustered to define the user's actual attentional foci on the computer interface. If a relatively long time is spent observing a particular screen area, more samples are taken within this area, and the resulting MST will be fairly tightly clustered. The clusters were computed following input of several controlling parameters. As opposed to eye-gaze clustering techniques requiring significant user interaction (Pillalamarri, et al., 1993), the MST representation allowed rapid, statistically-based clustering from the user defined parameters.

Using a depth-first search (Gibbons, 1984), the mean and standard deviation (SD) of internodal (between eye-gaze location samples) distances were computed, in screen pixels. Analyst-selectable inputs for defining clusters included the required branching depth (BD), distance ratio (DR), and distance SDs (DSDs). Due to the lack of circuits in the MST, two clusters were separated by only one internodal edge (i.e., one branch connecting two eye-gaze location samples). Thus, clusters could be separated if the distance between these nodes was greater than that of the locally surrounding nodes.

The clustering process considered each edge in the MST. At a given edge, the actual branching depth had to meet or exceed the user-input BD at each of its end nodes, to insure that cut edges separating clusters were not too close to the cluster edges. An edge not meeting this requirement at both nodes was not further considered as a cut edge. (Note, however, that BD=0 effectively disabled this check.) Given a successfully passed check of branching depth for an edge, edge lengths (in pixels) were collected from each of its defining nodes outward, to a depth of BD. The BD essentially defined the local edge environment with which a potential cut edge was compared. The mean and standard deviation edge lengths were

computed from this collection. The considered edge was marked as a cut edge if two criteria were satisfied:

$$\text{Mark as cut edge if: } \begin{cases} \frac{\text{Edge Length}}{\text{Edge Mean}} > \text{ER, (Edge Ratio) and} \\ \text{Edge Length} > \text{Edge Length Mean} + \text{DSD (Edge SD)} \end{cases}$$

Values in the range of 2-4 for ER and DSD generally provided intuitive spatial clusters. Larger input values forced clusters to be separated by greater distances, and were thus more conservative. Increasing ER relative to DSD forced greater emphasis on mean distance as the separation criteria, whereas increasing DSD relative to ER forced more emphasis on variance in edge lengths.

### Characterization of Spatial Clusters

Each cluster of eye-gaze samples was characterized for further analysis. For the present study, nine variables were computed for each cluster; many more are possible, for specific purposes. The variable symbols, units, and descriptions are presented in Table 1. The software also allowed all clusters within a data frame to be pooled into one representative cluster. In this case, the variables for each cluster were averaged, weighted by the number of nodes contained within each cluster. The frame is then represented by only one cluster, expressing the central cluster tendency on that frame.

*Table 1. Statistics generated from each cluster*

Symbol	Units	Brief Description
N	--	Number of nodes within cluster
M <sub>x</sub>	pixels	Mean horizontal spatial cluster location
M <sub>y</sub>	pixels	Mean vertical spatial cluster location
M <sub>e</sub>	pixels	Mean of cluster edges
SD <sub>e</sub>	pixels	Standard deviation of cluster edge length
M <sub>d</sub>	pixels	Mean distance from nodes to cluster x/y mean
SD <sub>d</sub>	pixels	SD distance from nodes to cluster x/y mean
M <sub>p</sub>	mm	Mean pupil diameter across cluster samples
SD <sub>p</sub>	mm	SD pupil diameter across cluster samples

### Cluster Mapping Between Frames

Because all eye-gaze samples within a data frame are considered to be simultaneous in time, additional characterization was necessary to discover temporal changes between data frames. For example, a large cluster on the first frame may split into two or more clusters on subsequent frames. Each of these may further split, or may reconsolidate as the user's attentional focus dynamically changes. Tracking of these clusters between frames requires a mapping procedure,

whereby each cluster within a frame is mapped to its precursor cluster in the immediately preceding frame.

The cluster mapping procedure implemented here used a constrained closest-distance algorithm. Between frames, clusters whose spatial means were closest were mapped to one another. The assignments considered all possible mappings, to provide an optimally closest mapped set. Note that additional constraints could be implemented, such as mapping clusters of approximately similar size or numbers of nodes. The direction of cluster movement could also be modeled to extrapolate the location of the mapped cluster on the next frame.

The mapped clusters provided a second set of parameters to discriminate the zoom-in/zoom-out/no zoom conditions. The same basic statistics that were presented in Table 1 were computed, except that computations were made between successive frames. For example, the mapped  $M_x$  described the mean difference in horizontal distance between each cluster in a frame, and its mapped predecessor in the preceding frame. These provided a glimpse into the dynamic characteristics of the eye-gaze clusters, in addition to the previously computed static characteristics. Note that additional sets of variable mappings could have been constructed for frames lagged by more than one. These may describe longer-term changes in the computer variables.

### Multiple Discriminant Analysis

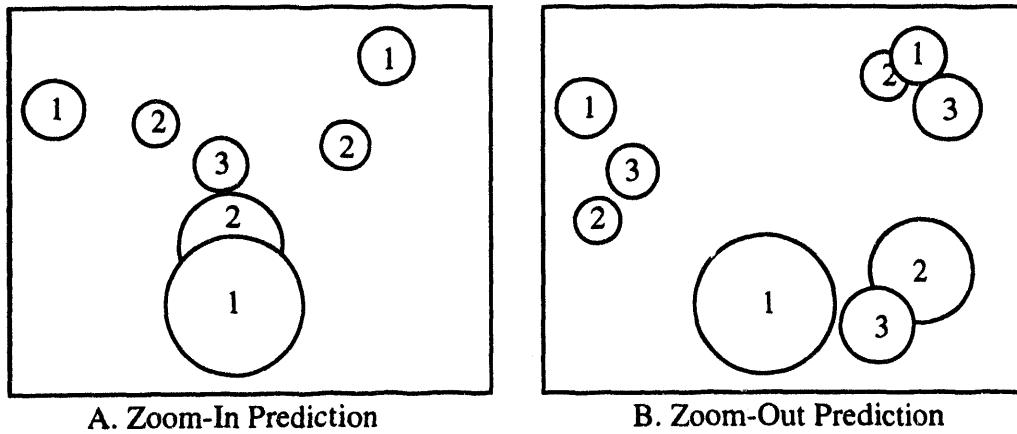
Given the static and dynamic cluster characteristics described above, a multiple discriminant analysis (MDA) procedure attempted to locate the best set of characteristics that would optimally separate the zoom-in, zoom-out, and no zoom conditions. In essence, the discriminant function minimized within-condition variance, while maximizing between-condition variance for a set of variables. The procedure computed either one or two linear discriminant functions to separate the groups; additional discriminatory power could potentially be obtained from nonlinear functions. The specific MDA used here was adapted from public domain code (Murtagh and Heck, 1987) and allowed the user to select 2-5 different model dependent variables.

The present methodology used the MDA as a classification technique. As the zoom condition classification was known by the software *a priori*, the input data was a training set for supervised MDA classification. A confusion matrix was generated to evaluate the success of a particular model. Significance of a particular classification was computed from a test statistic developed by Press (1972). The statistic tests equal probability of assignment to each of the three groups as its null hypothesis. For the three zoom conditions, or three classification groups here, the

statistic was  $Q = \frac{(N - 3n)^2}{2N} \sim \chi^2_1$ , where  $N$  and  $n$  are, respectively, the total number of clusters, and the number of clusters correctly classified. A significant statistic indicated that the MDA classification was significantly better than chance, which was equal probability of assignment across the three zoom conditions.

**Example.** To provide a concrete example of the MDA procedure, consider a two variable model. For simplicity, also consider only two assignment conditions: zoom-in or zoom-out. (Thus, we ignore the choice not to zoom.) Also assume that,

when signaling an intent to zoom-in, the eye-gaze spatial cluster sizes become smaller and tend towards one location on the screen. Assume the reverse occurs when signaling intention to zoom-out: clusters increase, and tend towards many screen locations. Fig. 4 illustrates these two predictions, by showing clusters over three data frames.



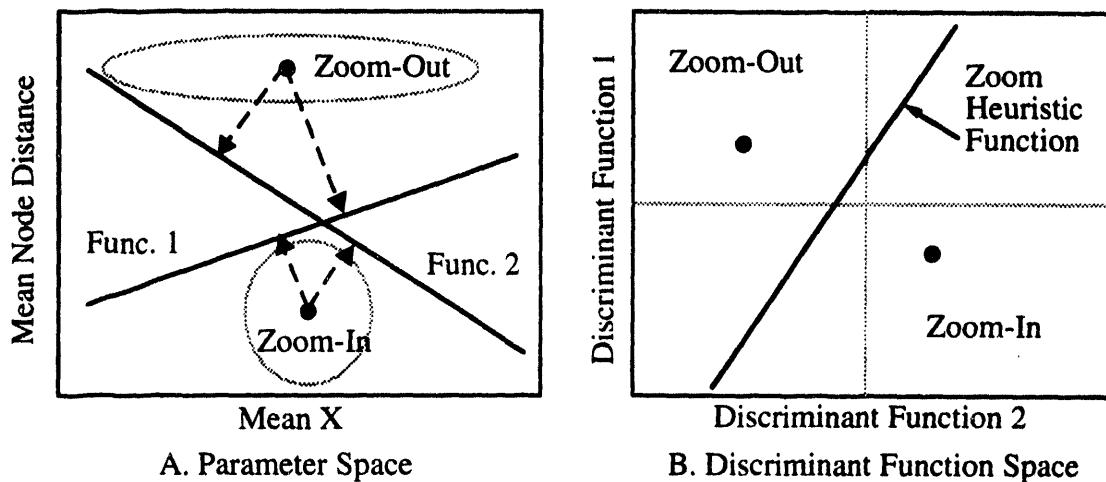
*Fig. 4. Two possible zoom condition predictions, based upon changes in eye-gaze cluster size and position over time. Clusters are shown on a screen interface, and are all prior to actual zoom-in or zoom-out. Data frame sequential number is indicated at the center of each cluster. A. Prediction for zooming-in, showing converging cluster locations over three data frames. B. Prediction for zooming-out showing cluster divergence.*

An MDA model here might include  $M_x$  and  $M_d$ , the mean horizontal cluster screen position, and the mean distance, within each cluster, from each node to its spatial x/y center. In this example, small values of  $M_d$  and mid-screen values of  $M_x$  indicate zoom-in, whereas larger values of  $M_d$ , with any value of  $M_x$  indicate zoom-out. This data pair is plotted from each cluster and zoom condition. Fig. 5A graphically shows these hypothetical data as lightly shaded ellipses, whose means are indicated by a solid circle. Two discriminant functions are formed that simultaneously maximize the data variance between these two groups and minimize the data variance within each group. The second discriminant function is used only when its eigenvalue can explain significantly more variance in the data. The projections from each of the zoom condition means onto each of the discriminant functions are computed. Fig. 5B plots these condition means and projections in discriminant function space.

The zoom heuristic is now formed by computing the perpendicularly bisecting function between these group means. After remapping back to parameter space, it will take the form:

Zoom-In if: 
$$\begin{cases} \beta_1(M_{d_1}) + \delta_1(M_{x_1}) < \theta_1, \text{ and} \\ \beta_2(M_{d_2}) + \delta_2(M_{x_2}) > \theta_2 \end{cases}$$
  
 Otherwise Zoom-Out.

The formulated zoom heuristic may now be used to classify new observations. It may remain static, or could be updated or recalibrated to suit varying users, conditions, or software needs.



*Fig. 5. Illustration of two dimensional MDA-based heuristic development. A. In parameter space for selected variable pair, projections of means from each zoom condition are mapped onto each of two discriminant functions. Actual data, represented by broader ellipses, may be highly spread. B. Zoom condition means are represented in discriminant function space. The zoom heuristic, which must be remapped to parameter space, perpendicularly bisects the group means in this space.*

### Generalization of Variable Relationships

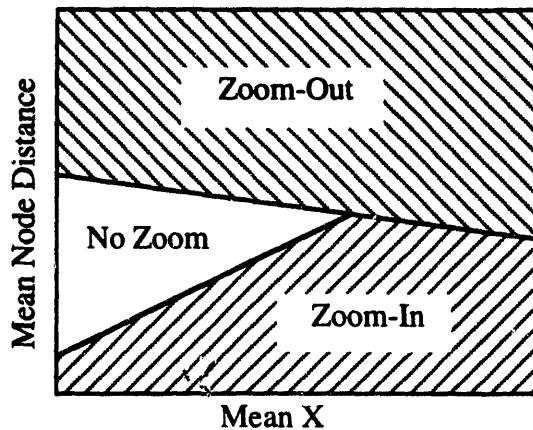
Zoom condition heuristics from the MDA are specific to an individual and a set of interface conditions. Broader generalization of these heuristics requires improved data and variable visualization procedures, both within and between individuals.

**Pooled Data.** The MDA can potentially be computed across subjects, by simply pooling cluster data into one file. Robust, user-independent phenomena will stand out, but some within-user characteristics will be lost. As an example, pooled-subject results showed that the most significant MDA assignments are found after a few data frames have been collected. That is, at least one second of eye-gaze sampling may be required before reliable zoom discrimination may occur. The model significance dramatically dropped for appreciably shorter or longer sampling intervals.

**Composite Variable Plots.** Superimposing several subjects' derived zoom regions on composite plots of variable pairs can provide valuable intuition on user-independent trends. Fig. 6 demonstrates this, using the same two example

variables, for a hypothetical set of users' data. There may be extensive overlap between zoom regions, but central tendencies are still quite apparent.

---



*Fig. 6. Example composite heuristic plot across several hypothetical users. Each area represents a zoom heuristic for a variable pair.*

## Discussion

A methodology was presented to assess and discriminate user zoom intent, at the computer interface, from eye-gaze cluster characteristics. The technique is empirical and adaptive. It may use either static or temporally dynamic characteristics. The methodology may also be generalized well beyond analysis of eye-gaze characteristics.

### Other Interface Operations

Besides determining whether to zoom-in or out, this technique may be applied to several other computer interface operations. Object selection, rotation, dragging, and inversion are three spatial operations that could potentially be controlled by eye-gaze.

### Other Classification Techniques

An MDA was used here to discriminate among the zoom conditions, but other techniques are also available. A neural net is one important example. Consisting of at least three layers of 'neurons', the input layer must contain as many neurons as parameters of the variables of interest. Here, 18 variables were defined; 9 defined characteristics within frames, and 9 between frames. The output layer in this zoom example would always contain 3 neurons; zoom-in, zoom-out, or no zoom. Once the neural net is set up, it need only be trained on a fixed set of zoom trials, for a particular user under fixed conditions.

## Acknowledgment

This work was sponsored by the Office of Technology Support Programs, U.S. Department of Energy, under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc., and U.S. DOE and Oak Ridge Associated Universities, under Contract DE-AC05-76OR00033.

## References

Belofsky, M.S., and Lyon, D.R. (1988). Modeling eye movement sequences using conceptual clustering techniques. *Air Force Human Resources Laboratory Technical Report AFHRL-TR-88-16*, Air Force Systems Command, Brooks Air Force Base, Texas.

Camerini, P.M., Galbiati, G., and Maffioli, F. (1988). Algorithms for finding optimum trees: Description, use, and evaluation. *Ann. Op. Res.*, 13, 265-397.

Frey, L. A., White, K. P., and Hutchinson, T. E. (1990). Eye-gaze word processing. *IEEE Trans. Sys., Man, and Cyber.*, 20(4), 944-950.

Gibbons, A. (1984). *Algorithmic Graph Theory*. Cambridge, England: Cambridge University Press.

Gips, J., Olivieri, P., and Tecce, J. (1993). Direct control of the computer through electrodes placed around the eyes. pp. 630-635 in *Proc. 5th Int. Conf. Hum.-Computer Inter.*, Amsterdam: Elsevier.

Hutchinson, T. E., White, K. P., Martin, W. N., Reichert, K. C., and Frey, L. A. (1989). Human-computer interaction using eye-gaze input, *IEEE Trans. Sys., Man, and Cyber.*, 19(6), 1527-1534.

Jacob, R. J. K. (1991). The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Trans. Info. Sys.*, 9(3), 152-169.

Just, M. A. and Carpenter, P. A. (1976). Eye fixations and cognitive processes. *Cog. Psych.*, 8, pp. 441-480.

Latimer, C.R. (1988). Eye-movement data: Cumulative fixation time and cluster analysis. *Beh. Res. Meth., Instr., & Comp.*, 20(5): 437-470.

Murtagh, F., and Heck, A. (1987). *Multivariate Data Analysis*. Boston, MA: Kluwer and Dordrecht.

Pillalamarri, R.S., Barnette, B.D., Birkmire, D., and Karsh, R. (1993). Cluster: A program for the identification of eye-fixation-cluster characteristics. *Beh. Res. Meth., Instr., & Comp.*, 25(1): 9-15.

Press, S.J. (1972). *Applied Multivariate Analysis*. New York: Holt, Rinehart and Winston, Inc.

Scinto, L.F., and Barnette, B.D. (1986). An algorithm for determining clusters, pairs or singletons in eye-movement scan-path records. *Beh. Res. Meth., Instr., & Comp.*, 18(1): 41-44.

Starker, I., and Bolt, R. A. (1990). A gaze-responsive self-disclosing display, *CHI '90 Proceedings: Empowering People*, Seattle, WA: Association for Computing Machinery, pp. 3-9.

100  
100  
100  
100

honey  
FED  
MED  
DATE

