Behavior Research Methods, Instruments, & Computers
1995, 27 (3), 338-350

Eye-gaze-contingent control of the computer interface: Methodology and example for zoom detection

JOSEPH H. GOLDBERG
Pennsylvania State University, University Park, Pennsylvania

and

JACK C. SCHRYVER
Oak Ridge National Laboratory, Oak Ridge, Tennessee

Discrimination of user intent at the computer interface solely from eye gaze can provide a powerful tool, benefiting many applications. An exploratory methodology for discriminating zoom-in, zoom-out, and no-zoom intent was developed for such applications as telerobotics, disability aids, weapons systems, and process control interfaces. Using an eye-tracking system, real-time eye-gaze locations on a display are collected. Using off-line procedures, these data are clustered, using minimum spanning tree representations, and then characterized. The cluster characteristics are fed into a multiple linear discriminant analysis, which attempts to discriminate the zoom-in, zoom-out, and no-zoom conditions. The methodologies, algorithms, and experimental data collection procedure are described, followed by example output from the analysis programs. Although developed specifically for the discrimination of zoom conditions, the methodology has broader potential for discrimination of user intent in other interface operations.

Eye Gaze in Computer Interface Control

The use of eye gaze as a computer interface control device is a recent concept with significant potential. Initially conceived for disability applications and military weapons targeting, eye gaze as an input device may readily extend to the control of advanced process interfaces, telerobotics, and camera manipulation (Hutchinson, White, Martin, Reichert, & Frey, 1989) or routine word processing (Frey, White, & Hutchinson, 1990). A great appeal of eye-gaze control is that it may serve as an effective replacement for mouse and keyboard input for high-workload tasks, thus freeing the hands to control other operations. For example, one might access items in a helmet-mounted database using eye gaze. Alternatively, eye gaze might control not only the direction of a wheelchair for a disabled individual, but also such subtle tasks as speed or route planning.

Eye-gaze tracking methodologies have been used to control two types of operations at the computer interface: spatial cursor position and object selection. Jacob (1990, 1991) presented an algorithm and demonstration of both of these in a videogame interface. He defined fixations after delays of 100 msec and ended fixations if data were received outside the current fixation area for at least 50 msec. Extensive averaging of spatial positions ensured that spurious blinks and or other anomalies were not considered. Objects were selected in this interface if at least a 150–200-msec dwell time occurred at a specific location; selections were easily reversible by fixating another object. The experimental eye-gaze-driven word processor by Frey et al. (1990) used a dwell time of 1,000 msec for object selection; it predicted probable letter combinations continuously in order to increase user speed and accuracy. These predictions effectively decreased the number of alternative characters needed to display as "lookpoints" to the user. Starker and Bolt (1990) considered varying models of required dwell time at an object for specifying user interest. Although they reported little experimental evidence, their work can aid in defining time requirements for object selection.

Although the above approaches for using eye gaze for computer interface control are only first steps, the dwell-time requirements prior to object or operation selection make them cumbersome to use in real time. Frey et al. (1990) and Hutchinson et al. (1989) were able to narrow the number of letter choices to 5–6 for eye-gaze selection, but each character still required an independent fixation and dwell for selection. Since the two eyes do not have the same independent-movement property that hands have in typ-

The activities described in this article were performed under Contract No. DE-AC05-76OR00033 between the U.S. Department of Energy (DOE) and Oak Ridge Associated Universities. The authors gratefully acknowledge the generous support of this research provided by Oak Ridge Associated Universities, Oak Ridge National Laboratory (ORNL), Center for Engineering Systems Advanced Research (CESAR, ORNL Engineering Physics and Mathematics Division), DOE Advanced Controls Program (ORNL Instrumentation and Controls Division), and DOE Robotics for Advanced Reactors Program (ORNL CESAR, Engineering Physics and Mathematics Division). J. C. Schryver is in the Human Factors & Cognitive Systems group of the Engineering Physics & Mathematics Division at Oak Ridge National Laboratory. Correspondence should be addressed to J.H. Goldberg, Department of Industrial and Manufacturing Engineering, Pennsylvania State University, University Park, PA 16802 (e-mail: jhgie@engr.psu.edu).

ing, this interface is quite cumbersome. Furthermore, their technique cannot handle the more abstract operations required for graphical user interfaces, such as object rotation or zooming in/zooming out. If controllable by eye gaze, such operations must necessarily rely on other, deeper level characteristics of eye movements.

Zooming In or Out at the Computer Interface

Zooming in for a narrower field of view (i.e., increasing the lens power of a telephoto lens) or zooming out for a wider angle view (decreasing the lens power) are two common operations in graphics, telerobotics, and process-control interfaces. Both camera mounted on the end of a robotic end effector or mobile platform and the arm itself must be controlled (e.g., Khosla & Papanikolopoulos, 1992; NASA, 1993). Although the problems involved in movement of a camera to extract three-dimensional information from two-dimensional views have recently begun to be investigated (Abbott, 1992), the control requirements of zoom-in/zoom-out detection have not yet been addressed. Both camera zoom and camera position may benefit from eye-gaze control, due both to the already frequent use of multi-degree-of-freedom hand controllers and to the high compatibility of using the eye to control one's point of view. Zooming under eye-gaze control may also benefit the control of virtual environment presentations (Stark et al., 1992).

Eye-Gaze Modeling: Samples and Clusters

A common property of the eye-gaze interface control methods described above is the search for fixations and subsequent saccades via minimum fixation time criteria. Although the eye operates in a fixate–saccade–fixate manner, identification and separation of fixations from saccades was not necessary here. Instead, a sampling approach is sufficient, and can be later used to generate fixation locations if still necessary. The sampling approach describes the *X*/*Y* gaze-point location within each sample, regardless of whether the eye is moving or stationary. Furthermore, fixation time may be a marker for processing difficulty or complexity at an interface (e.g., Just & Carpenter, 1980), but (1) the eye does not process foveal information the entire time the eye remains fixated at an object, and (2) parafoveal information, outside the immediate fixation area, is analyzed (Findlay, 1985).

As a basic unit of analysis, clusters of eye-gaze spatial locations are used here rather than temporal scanpaths. Scanpaths are essentially dynamic time-domain records of visual gaze point, whereas clusters are more stable space-domain records. Consider the comparison between a set of clusters and a scanpath shown in Figure 1. Each eye-gaze cluster represents a sample of spatially and attentionally related locations on a screen, and may contain temporally disparate samples. While the scanpath records the temporal relationship between eye-gaze samples, it does not generate clusters of common attentional locations. Moreover, scanpath analysis depends upon reliable sampling; if an eye-gaze location sample is missed, the characteristic scanpath may be greatly altered. Scanpaths are very dynamic, containing refixations and circuits. Clusters favor a more stochastic interpretation, where missed observations can be tolerated, given sufficiently large clusters.

There are several ways in which clusters are formed, and, given the importance of cluster formation and analysis in eye-gaze research, we describe some of them below.

Modeling eye-gaze clusters by k means. Latimer (1988) described a k-means method for cluster analysis based upon the earlier work of MacQueen (1967). First, the number of clusters in a distribution must be estimated, along with their mean *X*/*Y* locations. These are subjectively estimated from plots, and they define the initial conditions for further analysis. Each data observation is then assigned to the nearest cluster on the basis of minimum Euclidean distance. The cluster mean *X*/*Y* statistics are updated with each subsequent assignment. From experience with this algorithm, the cluster means gravitate toward the spatial modal locations among the data. Latimer also presented further information on theoretic metrics for avoiding initial subjective estimates.

Tullis (1983) used a k-means, nearest neighbor approach for grouping alphanumeric characters on a display. Used for interface complexity analysis, this algorithm was based upon methods discussed by Zahn (1971). The Euclidean distance was computed between each observation and its nearest neighbor. A graph was formed by connecting any character pairs separated by less than a threshold value,

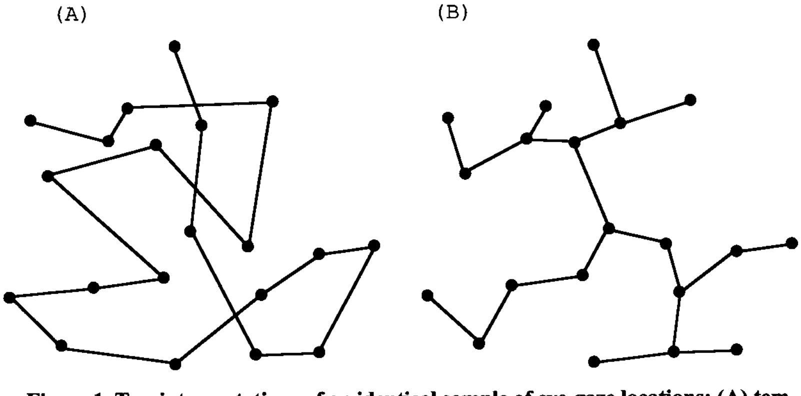


Figure 1. Two interpretations of an identical sample of eye-gaze locations: (A) temporal scanpath record, and (B) minimum spanning tree from graph theory.

typically twice the mean distance. A group of characters was then defined as a set of interconnected characters. This method was quite rapid, as the number of screen character locations was relatively small compared with graphical clustering methods that have much more spatial uncertainty.

The k-means approach can rapidly compute spatial cluster means, but may be criticized on the basis of flexibility and data representation issues. Once a series of observations is connected into a common cluster, regular search of the data points within a cluster is difficult. Clusters contain closed circuits of interconnections, making it difficult to span the data efficiently. The clustering approach used by Tullis (1983) is nonadaptive in that criterion or minimum threshold distances for separation of clusters is based upon statistics from the entire data graph as opposed to local data alone. Thus, graphs with local changes in data density will not separate into intuitive clusters.

Modeling eye-gaze clusters by minimum spanning trees. Zahn (1971) also described general algorithms for locally adaptive clustering methods. These usually require the initial formation of a minimum spanning tree (MST) prior to a locally adaptive search for clusters. The MST approach provides greater flexibility and control than does the k-means approach. These advantages include: (1) efficient tree search as the number of nodes becomes large, (2) user-controlled nonconsideration of nodes near cluster edges, (3) cluster separation based upon locally adaptive criteria, (4) potential for introducing additional cluster characterization parameters, and (5) no closed circuits (i.e., a branch never reconnects with an existing tree).

Modeling eye-gaze clusters by other approaches. Although cluster creation can be a data-intensive, highly quantitative process, justification of the final validity of a set of clusters is necessarily intuitive. Other valid clustering methods have been introduced for specific applications. For example, Ramakrishna, Pillalamarri, Barnette, Birkmire, and Karsh (1993) developed an algorithm to describe user-selected clusters of fixations. After the user forms a desired set of fixations, the program computes the vertices of a convex polygon containing all cluster locations. Cluster characterization variables include height and width, mean fixation position (both unweighted and weighted by time), area, and other indices. Scinto and Barnette (1986) used a similar subjective strategy for deciding whether fixations were part of the same cluster. They specified the minimum number of fixations required to establish a cluster and the minimum distance permitted between fixations before separation into multiple clusters. The program coupled adjacent clusters, as in a hierarchical clustering algorithm (Johnson, 1967). Belofsky and Lyon (1988) presented a rule-based algorithm for predicting visual attention clusters in an instrument-monitoring task. Major transitions between clusters were used to continuously update the size of each cluster. In effect, the system "learned" to discriminate legal transitions between instrument displays from underlying noise. Neural network-based approaches for cluster separation (e.g., Vinod, Chaudhury, Mukherjee, & Ghose, 1994) also hold outstanding promise due to their inherent ability to adapt to local cluster features.

Objective

The ultimate goal of this research is to determine whether signature characteristics of eye gaze precede user-driven interface operations such as zoom in or zoom out, and to harness them to control these operations. The objective of the present study was to develop and demonstrate a flexible off-line analysis methodology that could stimulate the development of on-line techniques for discriminating real-time user intent. For example, if a stable zoom-out discrimination heuristic were discovered in the off-line procedures presented here, the heuristic could be programmed into a very rapid on-line zoom-out discrimination demonstration. Little emphasis was placed here on the speed of software operation; great emphasis was placed on flexibly locating marker variables that could be used to advantage in later, faster on-line discrimination techniques.

PROGRAM DETAILS

Overview of Zoom Intent Modeling

The approach used here for inferring whether an operator would like to zoom in, zoom out, or do neither differs substantially from prior eye-gaze interface control methodologies. A multistep modeling procedure is used, as detailed in Figure 2. While the display is being viewed, a time-limited sample of *X*/*Y* monocular eye-gaze locations is collected, at the maximum sampling rate of the eye-tracking system (e.g., 30–60 Hz). The remainder of the intent-discrimination analyses were performed off line, allowing a broad search for variables that might impact the zoom-intent discrimination. Using Prim's algorithm (adapted from Camerini, Galbiati, & Maffioli, 1988), the spatial locations are connected to form a graph, forming an MST without circuits. The MST is separated into multiple clusters on the basis of adaptive and defined statistical tests. For each cluster, an associated mean *X*/*Y* location, mean and standard deviation (SD) diameter, mean and SD pupil diameter, and other parameters are computed. Clusters are formed and characterized from each subsequent (and possibly overlapping) sample, defining separate frames. As an additional step, clusters are mapped between frames on the basis of minimum distance, with each cluster of a frame mapped to a corresponding cluster in its preceding data frame. Multiple discriminant analysis next provided a means for classifying zoom conditions and providing heuristics for the separation functions. Emergent heuristics may be user dependent or trainable, or may be generalized to a broader population given similarities in natural eye-gaze tendencies among users. Future research will determine if such between-user similarities exist.

The zoom distinction is made by analyzing changes in clusters between frames. As an example, it might be expected that a user will focus his attention in smaller and smaller areas over time to signal an area to zoom in on. Conversely, focusing one's attention toward the outer areas of a

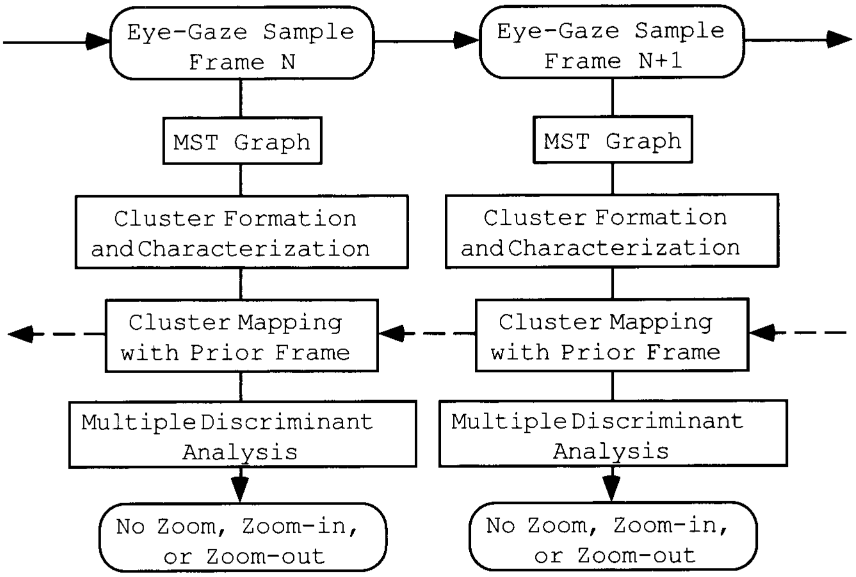

Eye-Gaze Sample Frame N
Eye-Gaze Sample Frame N+1
MST Graph
MST Graph
Cluster Formation and Characterization
Cluster Formation and Characterization
Cluster Mapping with Prior Frame
Cluster Mapping with Prior Frame
Multiple Discriminant Analysis
Multiple Discriminant Analysis
No Zoom, Zoom-in, or Zoom-out
No Zoom, Zoom-in, or Zoom-out

Figure 2. Conceptual steps of zoom-discrimination process, for two arbitrary data frames, *N* and *N*+1.

window may signal a desire to zoom out. Computable for any set of cluster characteristics (including changes in characteristics between frames), the multivariate data analysis (MDA) can define the criteria for best separating the zoom-in, zoom-out, and no-zoom conditions. These are displayed by projected lines onto variable scatter plots. Code for the MDA, adapted from Murtagh and Heck (1987), assigns the zoom groupings to the closest group mean in discriminant function space, using the Mahalanobis distance.

MST Formation Algorithm

The MST is formed using Prim's algorithm (see Camerini et al., 1988). Below, *nodes* are defined from the set of unconnected *X*/*Y* eye-gaze samples, and *vertices* are from the connected graph. In effect, the search space increases with the number of nodes, but decreases with increasing numbers of vertices. An *edge* of the graph is created with each vertex-node connection.

1. Starting at an arbitrary node, search the entire set of nodes for minimum Euclidean distance from the vertex. Connect this node to the starting vertex, forming the initial edge.

2. Remove the connected node from a list of available nodes, and add it to a list of graph vertices.

3. Find the minimum Euclidean distance, across all vertices and nodes, from a connected vertex to an unconnected node, and connect these.

4. Continue with Step 2, until no more nodes exist or the number of vertices is equal to the original number of nodes. For *n* original nodes, $n-1$ edges are created.

MST Search Algorithm

The connected MST can be searched to generate edge length mean and SD statistics. The MST is recursively spanned via depth-first searching, an efficient and systematic technique for visiting all vertices of a graph (Gibbons, 1984). Figure 3 displays an MST with numerical labels for the first 10 vertices searched, starting from an arbitrary vertex.

1. Start at an arbitrary vertex (Vertex 1 in Figure 3) and label as level 0.

2. Visit the first vertex connected to the parent vertex (Vertex 2). If such a vertex exists, increment and label its level. If no such connected vertex exists, decrement the level and return to its parent vertex. If the level is negative, the search is completed.

3. Continue exploring sibling branches at a parent node until all branches have been explored, then return to the prior parent vertex. In this manner, a vertex is revisited only by returning via edges that have already been traversed (Gibbons, 1984, pp. 20-21).

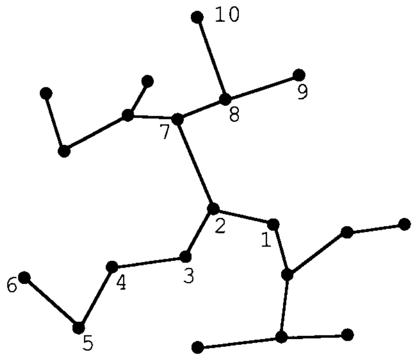


Figure 3. Sample minimum spanning tree, showing depth-first search order for the first 10 vertices, starting from Vertex 1.

4. When all branches at a vertex have been visited, the maximum branching depth can be defined from the maximum value of level. In Figure 3, the maximum branching level from Node 1 is 5 (achieved at Vertex 6).

Statistics such as mean and maximum branching depth can provide a means for characterizing the size and shape of clusters, rather than just defining cluster location.

Cluster Formation Algorithm

Clusters are formed by iteratively determining if each edge is longer than its local neighbors, ensuring that cluster formation is adaptive to local graph vertex densities. The cluster formation algorithm requires user-selected values for branching depth (BD), edge ratio (ER), and edge standard deviations (ESD), as explained below.

1. The data frame sampling is defined by user input values for sample size and sample offset. A frame of vertices of sample size is offset from the prior frame by the offset (except the initial frame, which has no offset). If the sample size and offset are equal, each sample is nonoverlapping, containing no vertices from the prior sample. If the offset is zero, one frame sample is repeatedly analyzed. Typical offsets of 5–10 observations for sample sizes of 15–30 observations provide sufficient memory between data frames for smooth cluster transitions. At 30 Hz, 15 frames represent a 0.5-sec sample of eye-gaze locations.

2. To be considered a potential cut edge separating two clusters, an edge, *e*, defined by vertices *i* and *j*, must meet or exceed user-input branching depth requirements; otherwise, it may be too close to the border of the MST. An edge that does not meet this requirement is not further considered as a potential cut edge. Starting with vertex *i*, the graph is searched in a direction away from *e*. Each successive connection away from the vertex forms a deeper level connection, as explained for the MST search algorithm. To fulfill the branching depth requirement, level ≥ BD, for both vertex *i* and vertex *j*. Figure 4 shows an example set of identical graphs for BD = 3 (Figure 4A) and BD = 1 (Figure 4B). Smaller values of BD effectively decrease the number of considered edges when generating local means and SDs. Small values of BD (e.g., 0 or 1) minimize or defeat the depth check, allowing potential cut edges to lie at or near graph boundaries. Larger values of BD (e.g., > 5) effectively require that potential cut edges lie well embedded within the graph, and that computed clusters contain a large number of vertices.

3. The potential cut edge is now compared to determine if it is long enough to separate two clusters. Note that, due to its lack of connected circuits, the MST data representation ensures that a cut edge separates no more than two clusters. Repeating the depth-first search described above, edges are collected starting at vertices *i* and *j*, proceeding up to a branching depth of BD, in a direction away from *e*. The mean and SD edge lengths are computed from this edge set. The edge is a cut edge if two criteria are satisfied:

$$\text{Length/mean edges} > \text{ER}$$

and

$$\text{Length} > \text{mean edges} + \text{ESD (SD edges)}. \tag{1}$$

Values of ER and ESD in the range of 2–4 provide intuitively conservative cluster separation. Larger values force clusters to be separated by greater distances. Increasing ER relative to ESD places more emphasis on mean distance than on edge-length variance for cluster separation. The presence of both ratio and variance criteria provide dual mechanisms to control the clustering process.

Statistical assumptions underlying the cluster criteria tests are minimal. Large edge samples, created from large sample sizes and values of BD, produce relatively symmetric distributions of edge lengths with smaller mean edge variance than smaller samples. Edge-length samples may

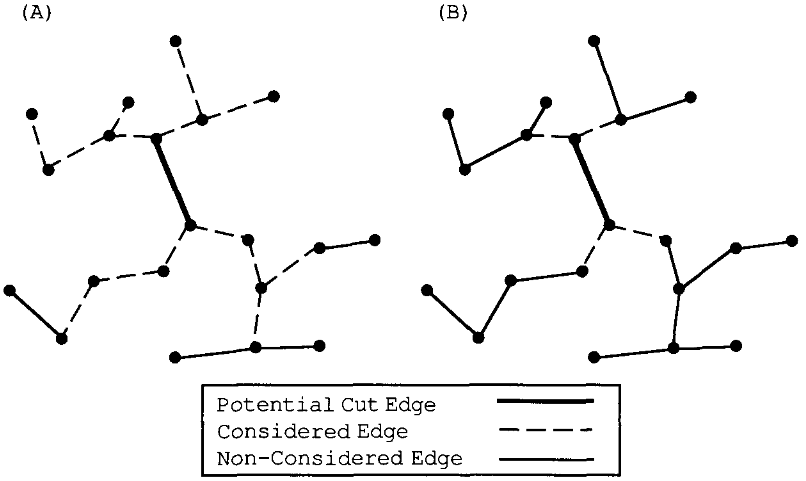


Figure 4. Example graphs, illustrating effect of branching depth (BD) on the clustering process (sample size = 20). (A) BD = 3, on each side of a potential cut edge, producing 14 local edges from which a mean and standard deviation are obtained. (B) BD = 1, producing only 4 edges for computation of the statistics.

be skewed, regardless of their sample size, due to a small number of extremely long edges. In these cases, means will likely be small and have larger SDs. Here, the mean edge (but not the SD) criteria are easily met. Thus, smaller edge-length samples result in more conservative cluster separation than larger edge-length samples. To guard against excessive skewness in the edge-length distributions, the user can input the number of SDs from the mean edge length, beyond which edge lengths are considered outliers in the cut-edge decision. This outlier criterion is individually computed for each considered edge.

The algorithm finishes by computing a set of statistics for each cluster, shown in Table 1. The M_d measures the physical size of a cluster using the absolute distance separation of each node from its cluster mean. The SD_d measures the variation about this distance; small means with small variation indicate tightly clustered vertices, as opposed to those from small means and larger variation. This set of generated cluster statistics provides a rich variable environment for subsequent discrimination algorithms. The MST data representation was required to generate this variable set.

Cluster Mapping Algorithm

In addition to discrimination within sampling frames, the zoom-discrimination methodology can detect regular changes in clusters between successive sampling frames. Each cluster within a frame is mapped or identified with a cluster in its preceding frame. Figure 5 presents three data frames, with cluster mappings noted by arrows. This is a visually intuitive process, as enlarging, moving, or contracting clusters are quite apparent when inspecting across data frames. The computer used here cannot efficiently consider every shape and detail characteristic considered by the human eye. Instead, the present algorithm maps these clusters using constrained minimum separation distance, assigning clusters to each other whose spatial means are the closest between two frames. Intuitively, the outcome of this algorithm agrees quite well with those produced from visual inspection. Conversely, those cases where mapped clusters are separated by longer distances are also harder to visually map. The steps of this algorithm are presented below.

1. The mapping algorithm starts with two adjacent data frames, each containing a set of clusters. Using M_x and M_y, the shortest Euclidean distance to each first-frame cluster is found for each cluster in the second frame. These clusters are matched.

2. The same matching is then repeated in the opposite direction, except that no match is made if two clusters are already mapped to each other.

3. The set of remaining intercluster distances is then sorted in ascending order, with the maximum number of cluster mappings defined by the maximum of the number of clusters in each of the two frames.

4. The remaining cluster mappings are assigned by descending the intercluster distance list from shortest to longest distances. On the first pass through the list, those clusters that are uniquely present among list members from the first and then second frame are assigned.

5. Remaining cluster mappings for which duplicate clusters exist in either frame are now assigned until all clusters have been assigned. The algorithm ensures that all clusters are assigned, and that minimum intercluster distances are used.

Pooled cluster option. A pooled cluster option is also available. In this case, statistics from each of the clusters within a frame are pooled by computing means, weighted by the number of vertices in each cluster. Each frame is then only represented by one cluster, which expresses the central tendency of cluster characteristics on that frame. The interframe mapping procedure proceeds by mapping this pooled cluster to the prior frame's pooled cluster.

A pooled cluster differs from a characterization of the unclustered graph on a frame. The entire graph in many cases contains several clusters, separated by relatively long distances. The mean distance from vertices to the spatial center of the graph does not represent individual cluster characteristics, as does the pooled cluster option. Note that in cases where the input number of data samples per frame is larger than the number of samples on a trial, only one cluster and data frame are created.

Interpretation of mapped clusters. The mapping of clusters between data frames provides a basis for a second set of variables that may track changes in cluster characteristics from frame to frame. For example, increasing cluster sizes produce positive values for changes in M_d. As dynamic entities, clusters may also expand, absorbing other clusters, or contract, spawning new clusters. These changes are consistent with interframe switches in attention, which may broaden or contract to specific display areas. The pooled-cluster option automatically maps the

Table 1
Statistics Generated From Each Individual or Pooled Cluster

Symbol	Symbol Meaning	Units	Brief Description
N_v	Number of vertices		Number of vertices within cluster
M_x	Mean *X*	pixels	Mean *X* spatial location of cluster
M_y	Mean *Y*	pixels	Mean *Y* spatial location of cluster
M_e	Mean edge length	pixels	Mean edge length within cluster
SD_e	*SD* edge length	pixels	*SD* edge length within cluster
M_d	Mean vertex distance	pixels	Mean distance from vertex to *X*/*Y* mean
SD_d	*SD* vertex distance	pixels	*SD* distance from vertex to *X*/*Y* mean
M_p	Mean pupil diameter	mm	Mean pupil diameter of cluster samples
SD_p	*SD* pupil diameter	mm	*SD* pupil diameter of cluster samples

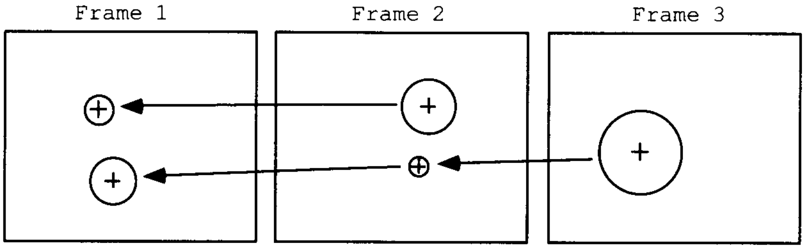


Figure 5. Example of interframe cluster-mapping process, based upon minimum distance between spatial means of clusters.

representative cluster on a frame to that on the preceding frame, and an aggregate interpretation of attentional changes is appropriate.

MDA Modeling

The objective of MDA is to discover discriminating axes that will achieve optimal separation among predefined classes. A discriminant function minimizes within-group variance while maximizing the distance between class means (or between-group variance). The zoom-in, zoom-out, no-zoom discrimination creates three classes; at most two discriminant functions are computed in this case. The second function is used only when it adds substantially to the discriminatory power of the solution. Linear discriminant functions were computed to optimally separate the zoom classes. The MDA used here, adapted from code presented by Murtagh and Heck (1987), attempts to classify observations into one of the zoom classes on the basis of a user input set of 2–5 model variables. The data used here were from an experiment in which the group membership of each observation was known a priori. The MDA classification is therefore a descriptive methodology here, and the input data are treated as a training set for MDA classification under a supervised classification paradigm. Classification criteria are developed with the training set, and can then be used to predict group membership for observations where membership information is unknown. If similar zoom-in criteria were discovered across individuals and situations, the resultant classification heuristic could be programmed into on-line application programs for rapid discrimination.

Each observation may be derived from either an individual or a pooled cluster. In addition, frame-to-frame differences in each of these can be modeled. No distributional assumptions or other properties are necessary with regard to the data.

MDA Visualization

The classification criteria are specified using projections of the class means into discriminant function space. Each point in discriminant function space is assigned to the nearest class mean. However, the criteria are of little value in discriminant space and must therefore be translated into parameter space. Criteria are described in the parameter space as the intersection of a pair of linear inequalities containing all input variables. Each inequality defines a region in the parameter space separated by a hyperplane. Figure 6 shows a typical linear separation for three classes and two discriminant functions in three-parameter space. Three half-planes partition the three-dimensional parameter space into three regions. When only one discriminant function is used, two parallel planes always separate the three classes. One class is then bounded on two sides; the others are bounded on only one side.

The decision criteria from the MDA are shown in Figure 6 as three linear functions separating the group means. When more than three input variables are used, the classification criteria shown on a single plot represent a slice across the hyperplane separators at the grand mean of all variables not shown on that plot. Normally the linear separators will capture class differences, but separation is not guaranteed in a two-dimensional plot. It is possible that a class mean will fall on one side of a linear cut but on the other side of a hyperplane in depth.

MDA Significance

The ability of the discriminant functions to classify the three zoom conditions is measured from a confusion ma-

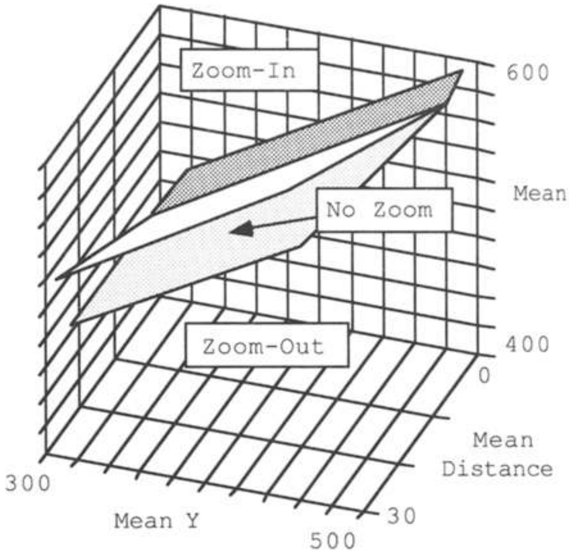


Figure 6. Three-dimensional representation of multivariate data analysis zoom classification for three input variables: mean X, mean distance, and mean pupil diameter.

Table 2
Example Confusion Matrix and Statistics

	MDA Assignment			% Correctly
Actual	No Zoom	Zoom In	Zoom Out	Predicted
No Zoom	17	9	10	47
Zoom In	3	28	7	73
Zoom Out	5	7	22	64
Overall				62

Note—$Q = (108 - 3*67)^2/2*108 = 40; p < .001$.

trix, as shown in the example in Table 2. This matrix tallies predicted versus actual observations in a 3 × 3 matrix, with successfully classified observations located on the diagonal of the matrix. Predicted observations were assigned from heuristics obtained from the mapped discriminant functions; observed data were the actual training data. The significance of the discriminatory power can be measured using a test statistic presented by Press (1972, pp. 381-383). The hypotheses for the present problem are:

H_0: $P(\text{zoom-in}) = P(\text{zoom-out}) = P(\text{no-zoom}) = .33$.

H_1: At least one condition had $P > .33$.

The test statistic was:

$$Q = (N - nK)^2/N(K-1) = (N-3n)^2/2N, \qquad (2)$$

where N, n, and K, respectively, are number of observations classified, number of observations correctly classified, and number of classification groups (always 3 here). The statistic is distributed chi-square with 1 *df* (Press, 1972). A significant rejection of H_0 in favor of H_1 is evidence that the MDA classifies observations significantly better than chance. The greater the value of Q, the better the classification. Note that the example shown here is for illustration only; the classification success will probably decline with new observations. Practical implementation of an algorithm will require routinely accurate classification on both training and new data.

SYSTEM HARDWARE AND EXPERIMENTAL DATA COLLECTION

Eye-Gaze Apparatus

Eye-gaze data are collected from an unobtrusive camera mounted below a workstation display. Figure 7 shows the major components of the data-collection system. The camera system (LC Technologies, Inc., Fairfax, VA) emits an invisible infrared light. The pupil and glint produced from the IR reflection off the cornea provide the system with sufficient data to compute a gaze angle, following calibration. The camera is interfaced to a host 386 PC, via a video digitizer card. The PC sends eye-gaze data across an RS-232 port to a Sun Sparc 2 workstation. The application software, written in C, read the data from the serial port while presenting and recording the experiment.

The present eye-tracking system has an average angular bias error of .45°, translating to .15 in. at a 20-in. eye–screen distance. Its tolerance to head motion, without head tracking apparatus, is about 1.5 in. laterally and vertically, and .5 in. in depth. A chinrest is currently used to stabilize the head during viewing. With head-tracking hardware, this tolerance can be increased to 10–18 in. The system works equally well with or without glasses or contact lenses.

Data Collection

Initial gaze-point calibration required viewing several known locations on the application computer display screen. The calibration was automatically repeated until a criterion distance accuracy was achieved. These calibration indices were sent over the serial port to a host PC file. Actual data collection then ensued, following explanations and practice trials.

To generate sufficient data to test the zoom-discrimination methodology, the experimental procedure presented about

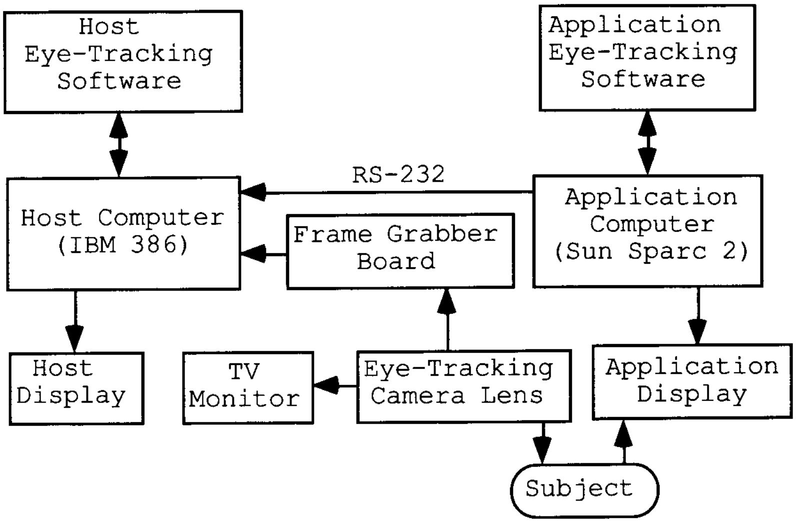


Figure 7. Block diagram of eye-gaze apparatus.

100 trials to each participant. The participant controlled the "s" and "d" keys with his/her left hand, and the left and right mouse buttons with his/her right hand. At the start of each trial, a simple test shape, which was to be memorized, was displayed for 2 sec. The shape spanned 20° of visual angle at the 20-in. viewing distance. Following a short pause, a comparison shape was presented. The experimental task was to determine if the comparison shape was the same as the memorized test shape. On many occasions, sufficient information was presented in the comparison shape to allow the participant to respond "s" for "same" or "d" for "different." On other occasions, however, a zoom in (by pressing the left mouse button) or zoom out (right button) was necessary to gain further information about the comparison shape. An "s" or "d" response was then made following the zoom operation. Eye gaze was collected only from the appearance of the comparison shape, and collection terminated with the first keypress event. In this manner, only eye-gaze locations immediately preceding a zoom-in, zoom-out, or no-zoom decision were considered in subsequent analyses.

EXAMPLE RESULTS

A commented example of one individual's results is provided below, showing actual screen graphics. These example results are provided to illustrate the methodology's capabilities, as opposed to providing a general solution to the zoom-discrimination problem. Converging results across many individuals and conditions will be necessary to provide a general zoom-discrimination solution, which may then be used on line. Figure 8 illustrates the off-line MST and cluster formation, under two sets of criteria, using a sample size of 30. Eye-gaze locations are shown here as small circles located with respect to the viewed shape, and of size proportional to the pupil diameter at each location. In Figure 8A, input values of BD, ER, and ESD were each equal to 3. Only one cluster, illustrated by a circle of radius M_d, was created from the MST. When BD and ER were decreased to a more liberal value of 2 (Figure 8B), the same MST was divided into two clusters, separated by one cut edge. Currently, a cluster may be as small as 2 samples, spanning a diameter of 1 pixel.

Following cluster formation, characterization, and interframe mapping, clusters may be plotted by experimental trial. Figure 9 shows clusters from two example trials under different clustering conditions. Observation of clusters by trial is important for developing theories underlying user intent discrimination. For example, Figure 9A shows clusters superimposed on a viewed shape just prior to zooming out, over a 1.5-sec period. Illustrating one cluster per frame across three frames, the observer started beneath the center of the shape, then moved his attention toward the center. During this time, the cluster size initially decreased, then increased. Figure 9B shows three data frames superimposed on a shape just prior to a zoom in. Here a more complex set of 2–3 clusters per frame were observed over a 1.2-sec period; the frame sizes were 0.4 sec here. Cluster sizes were initially large, then became smaller, as the observer's attentional focus moved to the center of the viewed shape.

The MDA and scatterplots provide a necessary means for capturing the multidimensional changes observed in

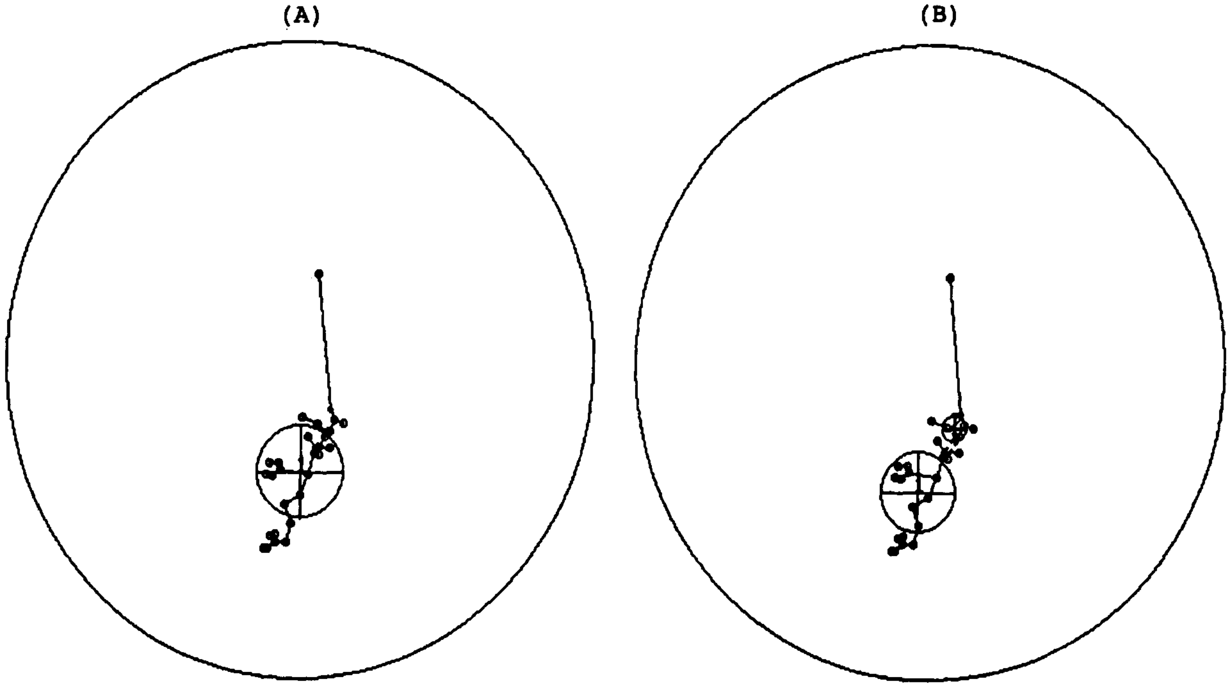


Figure 8. Minimum spanning tree and cluster formation, using sample size of 30. (A) Only one cluster is generated from BD = ER = ESD = 3. (B) Two clusters are produced when BD = ER = 2, and ESD = 3. (BD = branching depth, ER = edge ratio, and ESD = edge standard deviation.)

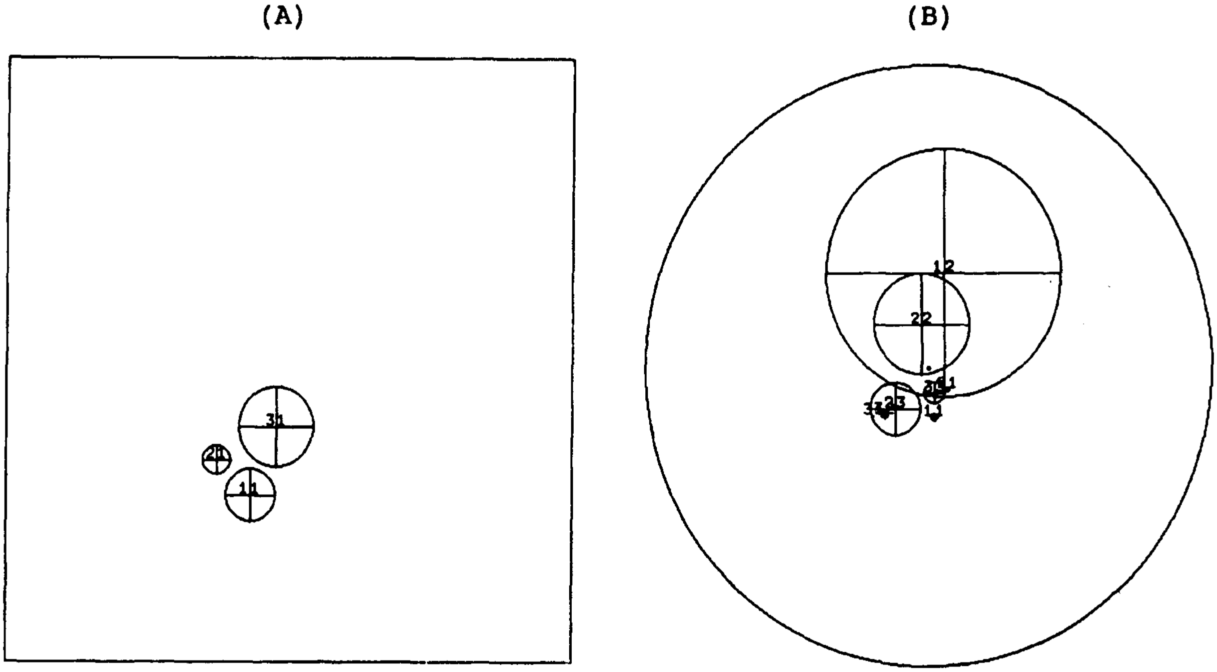


Figure 9. Examples of cluster trends on two different trials. (A) Three data frames, each with one cluster. Each cluster contains 15 samples, representing 0.5 sec. Cluster size decreased, then increased on this zoom-out trial, as eye gaze tended toward the center of the viewed shape. (B) A more complex pattern of clusters is shown on this zoom-in trial, containing three frames. Each frame represents 12 eye-gaze samples, or 0.4 sec; each contains two or three clusters. Attention moved in toward the center of the shape just prior to the zoom operation, and clusters decreased in size.

the prior two figures. Figure 10 shows a two-dimensional scatter plot for the variable pair, d MEAN X versus d AVG DIST, on Data Frame 4 of five frames. No outliers lie beyond the boundaries of this plot, whose axes are bounded by the variable mean ±3 *SD*. The d MEAN X is the change in mean horizontal cluster location (pixels) between clusters in Frame 4 and Frame 3, with positive values indicating temporal movement toward the right of the display. Other variables, such as the mean *X* and *Y* positions of each cluster also provide similar, often correlated information as the d MEAN *X* and *Y* variables, but the latter indices may provide dynamic activity information between mapped clusters that is not apparent from individual clusters alone. The d AVG DIST is the change, between the same frames, in mean distance from each node within a cluster to the spatial mean of that cluster. Positive values indicate clusters that are expanding; negative values indicate clusters that are contracting. In this figure, each cluster on the frame is represented as a shape; squares represent zoom-out, circles, zoom-in, and triangles, no-zoom conditions. The spatial means of these zoom conditions are represented by smaller dark symbols, corresponding to the symbols of the zoom conditions. The embedded numbers are the trial numbers on which each cluster appeared.

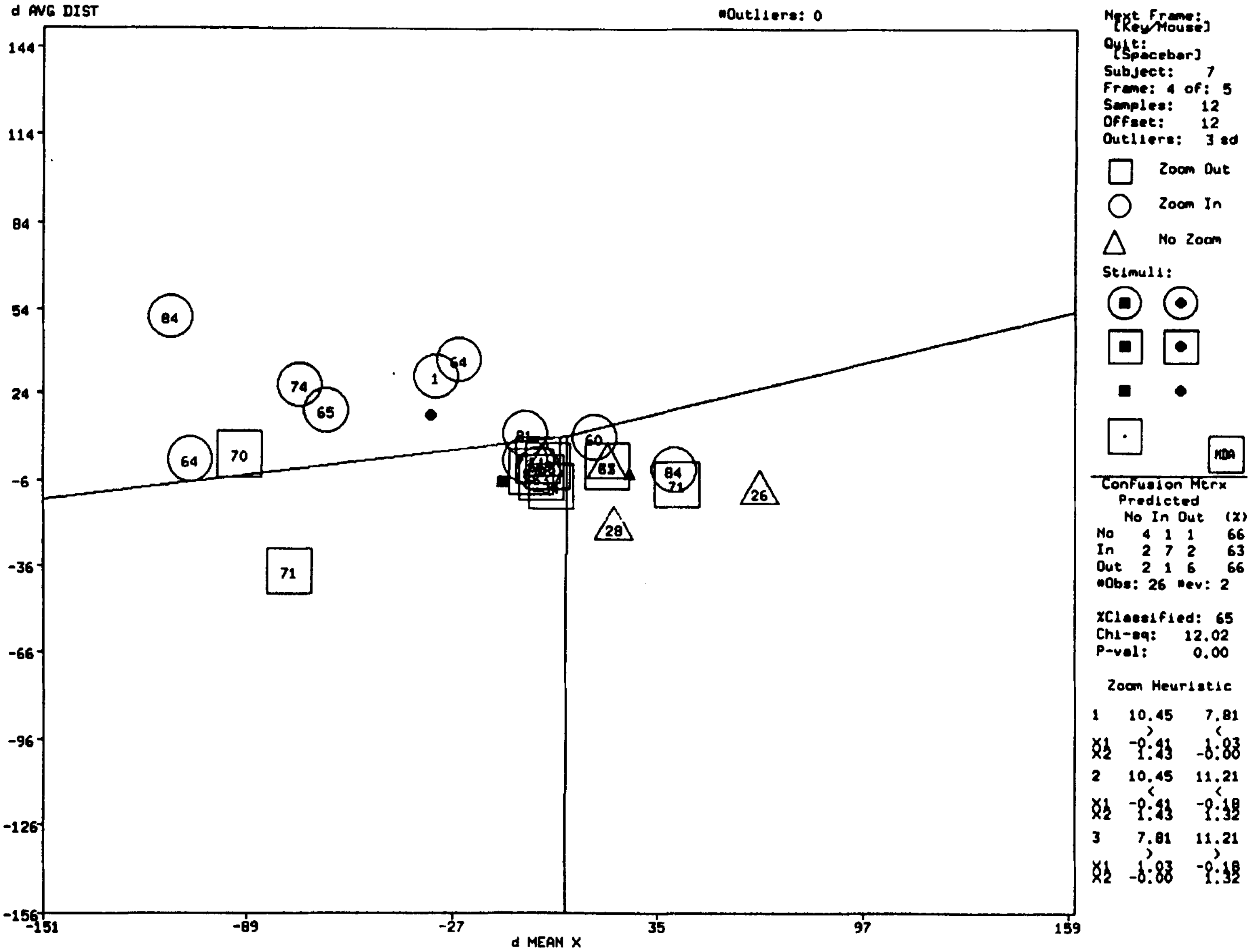
The MDA, computed from the two plotted variables, is overlaid on the example data in Figure 10 as a set of three decision lines. Because only two variables were used in the MDA, the solution is coplanar or two-dimensional. Three variable pair plots are displayed if three variables are entered, providing a three-dimensional solution; entering four or five variables provides a hyperplane solution that is harder to visualize. As indicated by the confusion matrix on the right side of the figure, the model correctly assigned 17 of the 26 clusters (65%) to their correct zoom conditions in this example ($\chi^2 = 12, p < .001$). The MDA solution used two eigenvectors to separate the three zoom conditions. The decision functions essentially form a heuristic, shown at the lower right of the figure, which may be used to classify new observations (given that one is satisfied with 65% accuracy). To illustrate, these heuristics were:

Do not zoom if:
−.41 (d MEAN X) + 1.43 (d AVG DIST) < 10.45, *and*
1.03 (d MEAN X) > 7.81.

Zoom in if:
−.41 (d MEAN X) + 1.43 (d AVG DIST) > 10.45, *and*
−.18 (d MEAN X) + 1.32 (d AVG DIST) > 11.21.

Zoom out if:
1.03 (d MEAN X) < 7.81, *and*
−.18 (d MEAN X) + 1.32 (d AVG DIST) < 11.21.

The MDA, for this example, counterintuitively assigned zoom in to clusters that expand between frames, and either zoom out or no zoom to contracting clusters. The separation between zoom out and no zoom was based on interframe changes in horizontal cluster position for contracting clusters. Right shifts indicated no zoom; left shifts indicated zoom out. Again, this example heuristic is provided



```
Next Frame:
 [Key/Mouse]
Quit:
 [Spacebar]
Subject:     7
Frame: 4 of: 5
Samples:    12
Offset:     12
Outliers:   3 sd

□  Zoom Out
○  Zoom In
△  No Zoom

Stimuli:

Confusion Mtrx
  Predicted
   No In Out  (%)
No   4  1  1   66
In   2  7  2   63
Out  2  1  6   66
#Obs: 26 #ev: 2

%Classified: 65
Chi-sq:    12.02
P-val:      0.00

 Zoom Heuristic

1    10.45    7.81
       >        <
X1   -0.41    1.03
X2    1.43   -0.00
2    10.45   11.21
       <        <
X1   -0.41   -0.18
X2    1.43    1.32
3     7.81   11.21
       >        >
X1    1.03   -0.18
X2   -0.00    1.32
```

Figure 10. Example scatterplot and multivariate data analysis (MDA) for Frame 4 of five frames. Circles (zoom in), squares (zoom out), and triangles (no zoom) represent individual clusters within zoom conditions, and solid shapes represent condition means. Overlaid decision lines are from MDA, as described by the heuristic at the lower right.

for illustration only; converging results across many individuals and conditions are necessary prior to further generalization of results.

The analysis and visualization application also offers a no-graphics option, which presents only the results of the analysis of the confusion matrix for all combinations of up to five input model variables. Results are rapidly presented for these variable subsets, allowing rapid identification and plotting of the most important variables from the MDA procedure.

DISCUSSION

An off-line analysis methodology for discrimination of zoom intent was illustrated here as an initial attempt to understand eye-gaze characteristics for subtle computer interface control operations, cursor positioning, and object selection. The discrimination was performed from a multistep clustering, cluster characterization, and MDA across frames of collected data. This general methodology holds potential advantages over temporal criterion-based dwell-time selection techniques (e.g., Frey et al., 1990; Hutchinson et al., 1989; Jacob, 1991), which can be relatively slow and cumbersome.

Cluster Analysis

Analysis of eye-gaze clusters allows an assessment of changing attentional focus at a computer interface. As an alternative tool to scanpath analysis, the dynamic mechanisms of cluster movement, contraction, and expansion were captured by the present methodology.

The multiple cluster formation and characterization methods used here relied heavily upon the spanning tree methodologies discussed by Zahn (1971). As opposed to the more subjective techniques often used for cluster eye-movement data (e.g., Belofsky & Lyon, 1988; Ramakrishna et al., 1993; Scinto & Barnette, 1986), the MST-based technique allows automated clustering based upon controlled comparison of local samples of eye movements. Efficient clustering algorithms (e.g., Camerini et al., 1988) have already been developed in other application areas, easing some of the burdens of developing new methodologies.

The relatively slow MST-based clustering used here provided a great deal of flexibility in cluster characterization, and a means for generating multiple clusters for this exploratory work. Faster approaches to clustering are possible, such as generation of best fitting ellipses from variance–covariance matrices among eye-gaze samples, but these were rejected in favor of the more flexible MST approach used here. Future work will develop real-time discrimination methods using faster clustering schemes.

Extensions to the cluster characterizations used here are numerous. While all clusters were represented here as circles, ellipses or polygons with an angular orientation may better characterize lines of movement over time. Characterization of these clusters could then include angle, minor, and major axes. Clusters were mapped between frames, here, on the basis of closest distance, but an algorithm also comparing other features, such as size or shape, may more accurately track rapidly moving or changing clusters.

MDA Classification

The MDA forms an integral component of this exploratory user-intent discrimination methodology, on the basis of its ability to generate optimal decision criteria that simultaneously consider many dependent variables. While some conditions may require only a two-variable model for accurate zoom-condition assignment, others may require a five or more variable hyperplane solution. In addition to its ability to consider multiple variables, the MDA can be rapidly computed, using efficient algorithms (e.g., Murtagh & Heck, 1987). Used here in off-line discriminant analysis, the MDA provides significant flexibility in discovering which cluster-based variables are important and in the generation of quantitative heuristics. Eventual agreement in the discriminating variables and resultant heuristics across individuals and conditions will allow simple hard coding of the discrimination criteria and rapid on-line decisions.

The logical extensions to the present off-line MDA methods are nonlinear analysis and improved visualization. Rather than optimally separating observations among conditions by linear discriminant functions, the nonlinear methods can fit quadratic or higher order functions. Given well-separated, nonoverlapping observations between the zoom conditions, the nonlinear functions should provide improved classification and significance for more error-free assignments. Improved decision space visualization can aid the interpretation of three or more space decision surfaces. Additional visualization tools can aid in comparing heuristics between individuals in order to generate composite, across-user heuristics.

Improved Zoom Classification

While only extensive exploration of the present methodology across a broad range of users, conditions, and stimuli will determine whether a static set of decision heuristics can be defined, additional within-user classification methodologies may also hold promise for decision heuristics that may be highly nonlinear. For example, a neural network could substitute the MDA in the present methodology. Following a short calibration period, the neural net could effectively provide the appropriate variable weights for efficient zoom-in, zoom-out, or no-zoom determination. The layers of the net could filter the 18 variables available here to the three zoom-condition groups.

User-Intent Discrimination

This study targeted zooming as a specific interface operation, but the methodology presented here is potentially broadly generalizable to other operations. For example, the stimuli in the "same–different" judgment task could be replaced by those varying in rotation about differing axes. Translation and object selection may also be studied in the same general manner. Ultimately, an entire eye-gaze-controlled interface may be feasible given sufficient experimental observation. The concepts of intent discrimination could use other types of data, such as keystroke or limb movements.

In all of these examples, the key to the success of this methodology is the discovery of natural "signatures" of variables that precede intended operations. The degree to which individuals exhibit such common natural signatures will ultimately define whether real-time procedures may successfully supplant the off-line analysis methodology presented here. Whether all users regularly exhibit such regular eye-movement patterns preceding zoom operations is still an open issue. Such operations as object rotation or translation may ultimately be better suited to the discovery of eye-movement signatures than is object zooming.

Considerations for Real-Time Applications

While the present work makes no claim to discovering a practical algorithm for effective eye-movement-based, real-time zoom control, it does provide an exploratory off-line methodology for discovering deterministic patterns in attentional focus preceding these operations. Given the discovery of an accurate and repeatable algorithm, there are several constraints to consider prior to the development of real-time applications. First, conservative criteria for zoom control would both lower the chance of false zooming and lessen the chance of correct zooming. An appropriate criterion, determined perhaps on the basis of costs and payoffs, is required. Second, a successful zoom-control discriminator must distinguish both zoom in from zoom out and any zoom from no zoom. An intermediate solution, using a mouse or other hand control in addition to eye gaze may provide sufficient redundancy in cases of nondiscrimination. Third, additional eye tracking following unsuccessful zoom operations may be able to indicate that something is wrong and place the system in an error-recovery mode.

Software Availability

The analysis software described here can be made available to interested parties. It is UNIX-based and programmed in ANSI C. Calls are made to X-Windows routines on a Sun workstation. For further information on obtaining the software, contact either J. H. Goldberg (e-mail: jhgie@engr.psu.edu) or J. C. Schryver (e-mail: ryv@cosmail1.ctd.ornl.gov).

REFERENCES

Abbott, A. L. (1992). A survey of selective fixation control for machine vision. *IEEE Control Systems Magazine*, **12**(4), 25-31.

Belofsky, M. S., & Lyon, D. R. (1988). *Modeling eye movement sequences using conceptual clustering techniques* (Tech. Rep. AFHRL-TR-88-16). Brooks Air Force Base, TX: U.S. Air Force Human Resources Laboratory.

Camerini, P. M., Galbiati, G., & Maffioli, F. (1988). Algorithms for finding optimum trees: Description, use and evaluation. *Annals of Operations Research*, **13**, 265-397.

Findlay, J. M. (1985). Saccadic eye movements and visual cognition. *L'Année Psychologique*, **85**, 101-136.

Frey, L. A., White, K. P., & Hutchinson, T. E. (1990). Eye-gaze word processing. *IEEE Transactions on Systems, Man, & Cybernetics*, **20**, 944-950.

Gibbons, A. (1984). *Algorithmic graph theory*. Cambridge: Cambridge University Press.

Hutchinson, T. E., White, K. P., Martin, W. N., Reichert, K. C., & Frey, L. A. (1989). Human-computer interaction using eye-gaze input. *IEEE Transactions on Systems, Man, & Cybernetics*, **19**, 1527-1534.

Jacob, R. J. K. (1990). What you look at is what you get: Eye movement-based interaction techniques. In J. C. Chew & J. Whiteside (Eds.), *CHI '90 Proceedings: Empowering people* (pp. 11-18). ACM Press.

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

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, **32**, 241-254.

Just, M. A., & Carpenter, P. A. (1980). A theory of reading: From eye fixations to comprehension. *Psychological Review*, **87**, 329-354.

Khosla, P. K., & Papanikolopoulos, N. P. (1992). Telerobotic visual servoing. *Journal of Applied Intelligence*, **2**, 127-154.

Latimer, C. R. (1988). Eye-movement data: Cumulative fixation time and cluster analysis. *Behavior Research Methods, Instruments, & Computers*, **20**, 437-470.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Statistics and Probability* (pp. 281-297). Berkeley: University of California Press.

Murtagh, F., & Heck, A. (1987). *Multivariate data analysis*. Boston, MA: Kluwer.

NASA (1993, February). Movable cameras and monitors for view telemanipulator. *NASA Tech. Briefs*, pp. 54-55.

Press, S. J. (1972). *Applied multivariate analysis*. New York: Holt, Rinehart & Winston.

Ramakrishna, Pillalamarri, R. S., Barnette, B. D., Birkmire, D., & Karsh, R. (1993). Cluster: A program for the identification of eye-fixation-cluster characteristics. *Behavior Research Methods, Instruments, & Computers*, **25**, 9-15.

Scinto, L. F. M., & Barnette, B. D. (1986). An algorithm for determining clusters, pairs or singletons in eye-movement scan-path records. *Behavior Research Methods, Instruments, & Computers*, **18**, 41-44.

Stark, L. W., Ezumi, K., Nguyen, T., Paul, R., Tharp, G., & Yamashita, H. I. (1992). Visual search in virtual environments. *Human vision, visual processing, and digital display III, Volume 1666, Proceedings of The International Society for Optical Engineering*, 577-589.

Starker, I., & Bolt, R. A. (1990). A gaze-responsive self-disclosing display. In J. C. Chew & J. Whiteside (Eds.), *CHI '90 Proceedings: Empowering people* (pp. 3-9). ACM Press.

Tullis, T. S. (1983). The formatting of alphanumeric displays: A review and analysis. *Human Factors*, **25**, 657-682.

Vinod, V. V., Chaudhury, S., Mukherjee, J., & Ghose, S. (1994). A connectionist approach for clustering with applications in image analysis. *IEEE Transactions on Systems, Man, & Cybernetics*, **24**, 365-384.

Zahn, C. T. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, **C-20**(1), 68-86.

(Manuscript received June 18, 1994;
revision accepted for publication July 29, 1994.)