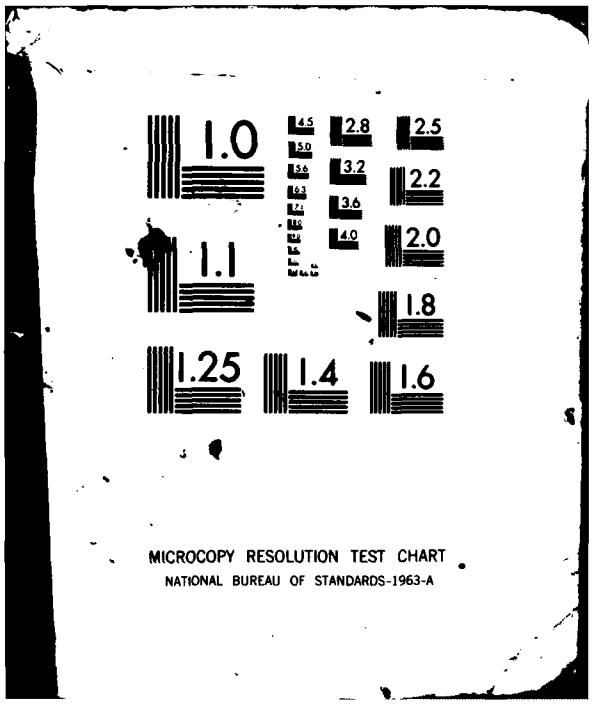
AD-A117 685 INTELLIGENT SOFTWARE SYSTEMS INC AMHERST MA F/G 9/2
THE DEVELOPMENT OF AN INTELLIGENT, TRAINABLE GRAPHIC DISPLAY AS--ETC(U)
JUL 82 A C MORSE, R KOHLER, M SUTHERLANT N00014-82-C-0136

UNCLASSIFIED NL

END
DATE FILMED
8-82
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(13)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: N00014-82-C-0136	2. GOVT ACCESSION NO.: AD-A117685
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): THE DEVELOPMENT OF AN INTELLIGENT, TRAINABLE GRAPHIC DISPLAY ASSISTANT FOR THE DECISIONMAKER	5. TYPE OF REPORT & PERIOD COVERED: Final
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Alan C. Morse, Ralf Kohler, Michael Sutherlant	8. CONTRACT OR GRANT NUMBER(s): N00014-82-C-0136
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Intelligent Software Systems, Amherst Fields Research Park, 529 Belchertown Road, Amherst, MA 01002	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62757N; RF57-701; RF57701801; NR 461-002
11. CONTROLLING OFFICE NAME AND ADDRESS: DCASMA Hartford, 96 Murphy Road, Hartford, CT 06114	12. REPORT DATE: July 26, 1982
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Dr. John O'Hare, Scientific Officer, Office of Naval Research, 800 N. Quincy Street, Arlington, VA 22217	15. SECURITY CLASS. (of this report)
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

For public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

DTIC ELECTE JUL 29 1982 S D H

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the top-level design of a trainable, intelligent graphics workstation, which is designed to assist decisionmakers to understand and analyse large amounts of data. The system has three key features: it is personalizable, it has

cont.

AD A117685

DTIC FILE COPY

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. cont.

facilities for the graphical presentation of data in a wide variety of formats; and it contains a user-programmable event monitor, which automatically searches the data for user-defined patterns. When such a pattern is found, a corresponding user-defined action is triggered.

The system was partially implemented and tested experimentally. The experiments attempted to test the key features of the system. The results indicated that the ability to present data in a variety of formats is desirable. Also, the results indicated that the event monitor was potentially a useful aid; however, these results were more ambiguous because of a high degree of inter-subject variability.

The report concludes that such a system would be highly effective and that the design is feasible.

Report N00014-82-C-0136

THE DEVELOPMENT OF AN INTELLIGENT, TRAINABLE GRAPHIC DISPLAY ASSISTANT FOR THE DECISIONMAKER

Alan C. Morse, Ralf Kohler, Michael Sutherland

Intelligent Software Systems, Inc.
Amherst Fields Research Park
529 Belchertown Road
Amherst, MA 01002

July 26, 1982

Final Report on System Feasibility

Prepared for

OFFICE OF NAVAL RESEARCH
Department of the Navy
800 North Quincy Street
Arlington, VA 22217

DCASMA Hartford
96 Murphy Road
Hartford, CT 06114

TABLE OF CONTENTS

1. INTRODUCTION

This report covers our efforts to create a flexible, personalized graphics workstation for the decisionmaker. We have designed a comprehensive system to be built in Phase II of this research. For this phase we built a prototype that included many of the important features of the eventual system.

Such a workstation would assist the decisionmaker by acting as a data analysis aid. The key elements of the workstation include:

1) A large set of formats for the graphical display of data.

2) A mechanism for choosing formats to display sets of data from a large database of information.

3) A mechanism for specifying patterns or events to search for in the set of data streams; when a pattern is found, it triggers an action, which may include the display of relevant data.

This report also describes preliminary experiments that we performed on the prototype of this system. The first set of experiments seeks to show differential behavior produced by differing display formats of the same data. The second set of experiments studies subjects' control of a simple environment that is changing in real time (a chemical factory), where a battery of display aids are provided. These aids include multiple display formats and a programmable event monitor patterned after the production systems familiar in AI research.

In both sets of experiments we have shown correlations between differences in display formats and differences in behavior. However, these correlations are not simple. They have given us insight into unexpected complexities of behaviour and provide guidance for further refinements of our system and for other systems of data presentation.

This report contains three major sections:

1) the SUMMARY OF RESEARCH, which briefly describes several aspects of our effort;

2) TECHNICAL FINDINGS, which elaborates on each of the subsections in the first section;

3) CONCLUSIONS, which states our feelings as to value of concept and asserts that we have demonstrated its feasibility.

2. SUMMARY OF RESEARCH

2.1. Design Rationale

Making effective decisions requires analyzing and understanding data (which may be generated in real time or which may be retrieved from a database). This requires being able to see patterns in numerical information. A graphics workstation is just the tool for assisting in this process.

The most obvious function of such a workstation would be to graphically encode the data, thus making the patterns evident. Additionally, the system can make evident different types of data patterns if it has facilities for graphically encoding data in many ways. However, simply being able to display all the data is not enough. With large amounts of data, the user will be overwhelmed, even if the data is represented compactly. Therefore, we augment the system with simple pattern detectors (event monitors). These event monitors scan the data for the simple patterns. When a qualifying pattern is detected the event monitor triggers an action, which may include the display of relevant data, a message to the decisionmaker, or a control response (which may alter the behavior of the system itself or the system that it is monitoring).

The user knows best the types of patterns that he is looking for in the data, so these components should be personalized. (That is, they must be programmable.) This means that he should be able to specify what events he is looking for in the data and what is to happen when those events are detected. Furthermore, he should be able to tell the system which data to display and how and when to do it. Finally, he should be able to do all this with a minimum of training.

2.2. Overall System Architecture

Our system would have the following components: A Data Collector (item #1 in Figure 1), which takes data from a database or from a stream of input data and inserts them into a State Variable Area (2). The State Variable Area contains the data of interest to the user. The Event Monitor (3) watches the data as it is collected. When one of a set of user-defined conditions occurs, the event monitor executes an associated action, which may be a modification of a state variable or which may be the initiation of some display action. The Display Subsystem (4) executes this display action by graphically encoding relevant data. The User Interface (5) allows (among other things) the definition of these event/action pairs, which are inserted into the Knowledge Base (6). Further, the User Interface allows users to interact with the display subsystem to define display actions. Finally, the system requires a System Monitor (7) to coordinate the actions of all the above components. (See detailed description in section 3.)

The user would interact with this system in two modes: training mode and running mode. Training mode is for personalizing the system. In this mode the user tells the system

what types of patterns to look for in the data and what to do when these patterns are encountered. Among these "things to do" are display actions, which the user must also describe. Describing a display action involves specifying what and how data are to be displayed. Once the system has been personalized, it is set into running mode. Then, when interesting events occur, the system takes the corresponding action (depending on the importance of the event and what was on the display already).

We chose the Chromatics CGC 7900 as the device on which to implement a prototype of this system. It is a high-resolution (1024 by 768) bitmap color (up to 256 different colors) microcomputer with relatively sophisticated software (including a UNIX-like operating system, Fortran, and C).

2.3. Limitations of Experimental Environment

It was impossible for us to implement the complete system in the time allocated, so we decided to implement part of the system and simulate the rest. Then we could expose users to the simulated system to test the value of the concept.

In the first experiment, we tried to simulate the personalized display format aspect of the system by having subjects view the display of the same data in several graphical formats.

For the second experiment, we implemented the display subsystem in the context of the real-time control of a primitive simulated chemical plant. The simulated system displays the state of the "chemical plant" using a fixed set of dials. The display can be augmented with graphs of the history of the data displayed in the dials. (See Figures 2 and 3 for an example display from this simulation. Figure 2 shows the left half of the display, which contains the dials, and Figure 3 shows the right half of the display, which contains the history displays.) Furthermore, the user can interactively define simple events that the system will look for. When these events occur, the user is notified. The results from these experiments are presented in Section 3.2.

We believe that these two experiments test very important aspects of the general design presented. They point to the value of the general, flexible, and trainable system proposed for Phase II research and development.

2.4. Display Subsystem

The display subsystem primarily comprises subroutines from a system we developed called DIGR. These routines, which we implemented on the Chromatics, facilitate the display of data in a multitude of ways.

DIGR allows the user to separate two parts of the data display task, specifying what to display and specifying how to display it. Furthermore, the "how to display it part" is done in such a way as to be as independent of display format as possible. This means that changing display formats involves a minimum of effort on the part of the interactive user. It simply requires that he invoke one of several display formatters. (For more

details, see the discussion in section 3.1.5.)

2.5. Event Monitor Subsystem

Although we didn't implement the event monitor with the full flexibility that is desirable, we describe it in this report because we feel that it is an important part of the eventual system. The event monitor subsystem comprises an event monitor and an associated knowledge base. The knowledge base consists of event/action pairs (also called, condition/action pairs). The event monitor watches the incoming data as it is put into the state variable area. It constantly checks this area to see if any of the conditions (events) specified in the knowledge base hold true. When they occur, the event monitor initiates the corresponding action, which may be a display action or which may involve setting another state variable. This, in turn, may allow more complex events to be recognized.

These event/action pairs are interactively defined by the user. (In the simulated system, the user was allowed to define a logical tree of simple binary relations on the chemical plant's state variables. For example, subjects could define the following event: if the temperature of Tank Z is greater than a threshold X and the heater for Tank Z is on then tell me to turn the heater off.) In defining an action in the full system in the future, the user would have the opportunity to interact with the display subsystem to define a display action.

2.6. Experimental Results

We ran two experiments, which were to test what we considered to be essential features of the system. One of the experiments was to test the value of flexible personalised displays in the interpretation of data; the other was to test this in conjunction with an event monitor. We wanted to test the value of having a variety of alternatives when displaying data. Since the display system that we implemented did not match our speed requirements, we tested this by presenting data in a variety of formats simultaneously so that subjects could "select" a display quickly by simply moving his eyes to the appropriate position on the CRT. The results indicate that the subjects were more accurate when more than one format was available; although their time to respond increased.

It may turn out, upon further investigation, that this simultaneous presentation of alternate display formats would be better than a sequential presentation. The waiting time between frames may disrupt the user's concentration. In any case, our envisioned system would allow both simultaneous and sequential presentations of alternate formats.

Furthermore, we found that different display formats were suitable for different information processing tasks (no surprise here). If we could quantify exactly what each display was good for, we could put that information into the knowledge base so that

the user could automatically be presented with the display that was most appropriate to his task.

The results were not as clearcut in the chemical plant experiment. Most notable among the findings was the superiority of having the event monitor and the alternate displays for one of the subtasks (keeping the chemical plant pressure and temperature within safety limits). Because of high variability among the subjects, the significance of the result was only $p<.1$. We feel that this variability would be reduced if the subjects had had more training in the system and if the system had been more easily customized and more flexible.

3. TECHNICAL FINDINGS

3.1. Details of System Architecture

In this section, we describe in more detail the system that we designed. (Remember, we only implemented a portion of this system for this phase of the grant.) See Figure 1 for a block diagram of the system as we envision it.

3.1.1. Data Collector

The Data Collector subsystem is the interface between the workstation and the world containing the information to be analyzed. This information may reside in a database or it may consist of readings from a collection of sensors. In the former case, the data collector would act as a database management system (DBMS) under the control of the user (via the system monitor). As such, it could retrieve user-requested data for analysis. In the latter case, the Data Collector would collect data (synchronously or asynchronously) from user-selected sensors and store it in the State Variable Storage Area. This latter case is our primary interest; we are designing the system as if it were to be used in a real-time process monitoring and control system, which may involve many data streams at high input rates.

3.1.2. State Variable Storage Area

The State Variable Storage Area contains the information of direct interest to the user. In the case where the workstation is used to monitor real-time data, this area would contain 1) the raw sensor data (with recent histories and/or time stamps), 2) statistical data generated from the raw data (such as, running averages, convolutions, cumulative sums, derivatives); 3) event flags and variables (generated and used by the event monitor); and 4) action demon flags.

Of these, 3) and 4) require further explanation. Event flags and variables provide a mechanism whereby events can trigger other events (rather than simply triggering a display action). The action associated with a detected event may involve altering one of these event flags or variables. This, in turn, may cause some other event condition to be true since event conditions may include these event flags and variables. Thus, event/actions may be chained or arranged as a logical AND/OR tree. This is a powerful feature; however, it leads to the sequencing problem described in the next section. (An example of event/action pairs that involve event variables is given in section 3.1.4.)

Action flags provide a means for initiating display actions. When the event monitor detects an event, it triggers a display action by setting the corresponding flag. However, the action doesn't actually start until the system monitor, which is continuously polling the action flags, sees that one of them has been set. The system monitor then initiates the appropriate action and clears the action flag.

3.1.3. Event Monitor

The job that the event monitor performs can be simply stated: it watches the state variables and looks for certain conditions (events); when a condition is found to be true, the event monitor initiates an associated action. The conditions and their associated actions are stored in the Knowledge Base described in the next section.

However, the event monitor is more complicated than this description makes it appear. Timing is one technical problem, because the system comprises (conceptually) parallel processes; data is being collected while it is being monitored while it is being displayed. How does the event monitor know that it is time to look through the state variables for an event? What if a variable gets updated before it can get to it? Sequencing is another problem, because the condition/action pairs are partially ordered and because the state variables include event variables. Therefore, sequencing is the problem of trying to apply the condition/action rules in the right order to the state variables in the right order. While we have not completely solved these problems, they are not intractable.

3.1.4. Knowledge Base

This contains the condition/action pairs (production rules) that the event monitor uses. They are essentially if-then statements, which the event monitor repeatedly executes. That is, the Knowledge Base comprises statements of the form "if condition X is true, then perform action Y." The condition can be arbitrarily nested logical (AND, OR, NOT) expressions containing event flags or binary relations (greater than, equal to, less than, etc.) on state variables. An example: "IF the tank temperature is GREATER THAN the safety threshold AND the tank's furnace is on AND the two previous conditions have been true for 10 minutes THEN print the message 'something has probably gone wrong with the system that turns off the furnace.' "

Note that the third part of the condition involves an event variable (see section 3.1.2.) that keeps track of how long the first two conditions have been true. There would have to be in the Knowledge Base other event/action pairs that would be responsible for maintaining this event variable. Suppose we call this event variable DANGER_DURATION. Then there would have to be event/action pairs: 1) "IF the tank temperature just now became GREATER THAN the safety threshold AND the tank furnace is on THEN set DANGER_DURATION to 0 minutes and set the DANGER_START_TIME to the current time"; and 2) "IF the tank temperature is GREATER THAN the safety threshold AND the tank furnace is on THEN update DANGER_DURATION to elapsed time (current time - DANGER_START_TIME).

The example illustrates that the ordering of the if-then statements is important, since event/action pair 1) sets an event variable that 2) uses, namely, DANGER_DURATION and DANGER_START_TIME. In fact, condition/action pairs are only partially ordered, since some of the condition/action pairs are completely independent of each other.

The Knowledge Base contains both pre-defined "expert" rules

and user-defined "personalized" rules. The latter rules are inserted interactively via the User Interface and saved for later use in the user's library.

3.1.5. Display Subsystem

The Display Subsystem graphically encodes selected data from the State Variable Storage Area. It operates in two modes: define mode and display mode. The former mode is for describing display actions while the latter is for executing them. (This corresponds to the training and running modes that we described in section 2.2.)

When the user defines a display action, he is really doing two things; first, he describes what he wants displayed, then he describes how to display it. The first part is a straightforward description of the data: where it is in the State Variable Storage Area, what it looks like, etc. The second part involves defining an abstract display and then specifying a display formatter that is to turn that abstract display description into a concrete display.

What is an abstract display and how does one create it? An abstract display is a generic display; it has all the elements that you would expect to find in any display, regardless of the graphical encoding scheme that the display used. (That is, these elements are independent of display format.) For instance, an abstract display has a title, axes (scales with tick marks and labels), axis labels, legends, assorted flags, datum/color associations (which result in the display of the datum always being a certain color), and the screen viewport in which it is to appear. The user creates the generic display by simply giving it a name and specifying its parts: "I want a display, call it X, of state variables w and z", "I want the title to be such-and-such", "I want the display to appear on this part of the screen."

Once the user has created an abstract display, he can call a specific display formatter to turn that abstract display into a concrete one. That is, the display formatter graphically encodes the data described in the abstract display. (This is similar to the concept of device independence, where the abstract display is like a set of device independent commands and the display formatter is like a device driver.) The advantage of this approach is that the user can look at the same data in different ways by simply hooking in a different display formatter. Furthermore, he can devote his full attention to figuring out what he wants to see without having to simultaneously figure out what would be the best way to view it.

The system will include a large library of (25 to 50) predefined display formatters. The system further will include facilities for defining new display formatters. We would like to make this facility interactive, such that a non-programmer could define a new display format on-line. We believe that we have designed the system's data structures so that this is possible. We can do this because we have broken up the definition of a display formatter into several modules (for example, a display context layout module, a display space allocator, and a graphical

data encoder). The user could then be able to combine predefined modules in each of these categories to create a new display formatter; "one from column A, one from column B, etc."

3.1.6. User Interface

We envision the user interface as a tree of menus, which control all aspects of the system. Ideally, the menus would appear on a CRT separate from (but adjacent to) the display CRT, so that menus don't compete with the displays for screen space. We would like the menu CRT to include a touch panel and graphics capabilities so that the menus could contain icons (to represent the various display formats, for instance).

The menu tree will have two major branches, one for training the system and the other for controlling its execution. In training mode, the user will have menus for 1) selecting which data is to be placed in the State Variable Area and how it is to be stored (with history, with time stamps, etc.); 2) selecting statistical functions to be performed on the data; 3) defining what events the system should look for in the data and what actions to perform when the events are encountered; 4) defining display actions, which require specifying which data from the State Variable Area to display and what graphical format to use; 5) defining the priority level (importance) of actions; 6) defining the screen layout.

In running mode, the user should spend most of his time monitoring the screen; although interaction is possible. He may trigger display actions independently of the event monitor; that is, he can interactively define and view any data from the State Variable Area. When events are detected by the event monitor and a display action is triggered, one of three things may happen, depending on the action's associated priority: 1) the display will automatically be displayed (possibly displacing a lower priority display on the screen); 2) the user will be sent a message indicating that an event has occurred and "would he be so kind as to notify the system if he wishes to see the associated display"; and 3) the display action is completely ignored by the system. Which of these three alternatives is taken depends on priority thresholds, which the user can interactively set.

Designing the user interface first requires designing the user's mental model of the system. This should be a simplified but accurate version of the "real" system, where complications in the system design are hidden. This mental model must be taken into account when designing both the User Interface and the system as a whole. More effort on engineering this user model would certainly be an important component of any eventual system.

3.1.7. System Monitor

The System Monitor is the yet-to-be-designed coordinator of all the above modules. It's primary functions are to 1) mediate communication between the user and the Knowledge Base, Data Collector, Event Monitor, and Display Subsystems; 2) to provide for the synchronization of the various modules; and 3) to handle the display action ordering and management for the Display

Subsystem.

3.2. Multiple Display Formats Experiment

3.2.1. Design

The first experiment allowed us to explore two primary questions: Are multiple displays of the same information more useful than single displays? Are different display formats suitable for answering different kinds of questions?

The experimental procedure focussed on the presentation of 16 by 16 arrays of numbers (see figure 4 for an example) graphically encoded in four ways: intensity encoded (figure 5), surface encoded (figure 6), size encoded (figure 7), or angle encoded (figure 8). A subject, sitting at the Chromatics workstation, would simultaneously be shown one, two, or three of these encodings of a particular 16 by 16 array. For each such screening, the subject would be asked such questions as, "How many peaks (maxima) are visible?", "Are there any pockmarks (discontinuities) in the display?", "Which of the four extreme corners of the array has the highest value in it?" Questions of how to define a peak, a pockmark, or a corner had been addressed prior to running the experiment. The subjects' response was nothing more than pressing the appropriately numbered key on the workstation's keyboard. The subjects had been informed that both accuracy and response time were of interest.

Fifteen subjects completed the full forty trial experiment. The order of display and question presentation was the same for all subjects. The particular 16 by 16 arrays and their associated displays were organized in such a way as to allow inferences about format effectiveness, display effectiveness, and potential differences between arrays.

3.2.2. Results

The unreduced data from this experiment consists of 15 subjects' responses to two primary questions ("How many peaks are there?" and "Which corner has the highest value?") over 40 trials. There are also two secondary questions ("How many pockmarks are in the data?" and "Was there a clear maximal value?") asked of each subject, but usually in only 20 of the 40 trials.

Our analysis consists of looking at the given responses as either correct/incorrect or as some deviation from correct and then simply comparing the performance for the different display formats and for different numbers of displays. That is, we wanted to know if display type was important and if the number of display types displayed simultaneously was important. Our analysis takes place across subjects and usually depends on straightforward applications of t-test, ANOVA, and other group comparison techniques.

Results for the accuracy with which subjects count the number of local maxima in the data clearly indicate the following ordering of format types from most to least accurate:

Surface)) Intensity) Size)) Angle

("))" means that the difference is significant at the .05 level. A single ")" means that the result represents a trend in the data that is not significant at that level.) The primary result comes from the t-test procedure and is confirmed by a Duncan's Multiple Range test and its associated creation of homogeneous subgroups. The relationship between the formats holds up as one conditions on particular images or on the number of formats in the display.

The number of displays on the screen at one time also affect performance. Two displays are better than one ($p<.01$) and three displays are worse than two ($p<.01$). The latter result is surprising. One would expect that the more displays there are, the more accurate the subject would be, since he would be more likely to find the display that suited the task. One explanation is that the increase in the display number led to an increase in time pressure. (The subject would be worried about getting the questions answered quickly and would not look at the alternate displays properly.) Another explanation is that increasing the number of displays increased the probability that one of the displays would be an angle display. It seems that angle displays are so bad that there mere presence is a confusing influence; some subjects did not learn to ignore it when it wasn't appropriate. A third explanation may be that the resulting display complexity confused the user.

Both the "pockmarks" question and the "highest corner" question show that as one changes the question one should realize that alternate displays are called for. For instance, in spotting pockmarks (discontinuities) in the data, intensity and surface encoding were better ($p<.05$) than size and angle encoding. For comparing widely separated data points (corners), size and angle encoding were better ($p<.05$) than surface or intensity displays. Again, these statements are based on simple differences in average performance as measured by t-test, Duncan's Multiple Range and related conditional tests on subpopulations.

3.3. Chemical Plant Experiment

3.3.1. Description of simulated system

The chemical factory simulation provides an adequately rich and complex environment in which experimental subjects are asked to maximize the output of the factory while keeping reaction temperatures and pressures within safe limits. The factory consists of three reaction vessels labeled X,Y, and Z. For each reaction vessel the subject must monitor the temperature and pressure given a minimum, maximum, and optimum temperature and pressure. The subject controls the temperature by activating and deactivating a heater associated with each tank. Similarly, the subject controls the pressure in the tank by opening or closing either one of two feedstock valves or by opening or closing the product output valve. Figure 2 shows the arrangement of the three tanks in the factory. The top two tanks (X and Y) produce the two feedstocks needed for bottom tank (Z).

The factory model defines the reaction rate in each vessel according to the amount of the feed chemicals available to react, the storage space available for the product chemical, and the current distance from the optimum temperature and pressure in the reaction vessel. The temperature in the tank is increased if the heater is on and decreased by heat loss to the environment. The reaction may be either endothermic or exothermic and affects the temperature in proportion to the reaction rate. The pressure is simply proportional to the temperature and the amount of empty space in the tank. The rate of chemical flow into and out of the tanks is proportional to the feedstock pressure and the valve settings.

In order to help the user control this environment subjects were sometimes given a set of nine alternative displays which provided historical information about the various tank states while at other times subjects were given a primitive event monitor to help monitor the factory state. The left side of figure 2 shows the complete state of the chemical factory at a given instant. The displays on the right side of that figure should help the user to understand how the system variables are changing over time and how the variables are related to each other. The event monitor should remind the user when critical events happen while the subject is attending to another portion of the factory or display.

The user can interactively define the event/action pairs. When a specified event is true or becomes true, the system displays a user-defined message (the only type of action allowed in this simulation). If the message is displayed only when the event becomes true the event is referred to as a triggered event. In the chemical factory the subject could define triggered events to remind him to turn on and off the heaters as soon as the tank temperature deviated from the optimum by a specified amount. An event that displays its associated message as long as the event is true is called a continuous event. The subject could define this kind of event to continuously remind him that the pressure in a reactor was too high (since this might damage the factory). The events can be simple relations between a tank state and a constant, such as, "Tank X temperature < .2" or they can be more complicated, such as in the event/action pair: "if the temperature in tank X > .7 and the tank X heater is on then generate the message "Turn off Tank X heater".

3.3.2. Design

We asked each subject to spend approximately one hour familiarizing himself with the keyboard responses, the visual displays, and the basic workings of our simple event monitor editor. Subjects accomplished this by reading a set of instructions, asking questions of the experimenter, and, most importantly, by running the system several times. (The subjects found the system fun to play with, to the extent that some even asked to play it outside of the experimental context. Therefore, we refer to the chemical plant control simulation as "the game.")

During this training period, the subjects tried three different versions of the system: one that had only the standard

dial display; one that had the dial display and an alternate history display simultaneously on the screen; and one that had the standard display with an event monitor/editor available. The subjects also tried the game at different speeds, so that we would be able to estimate time pressure effects. We told all the subjects what the minimum, maximum, and ideal settings were for the temperature and pressure in each of the three vats. These settings were also incorporated into dials and color-coded so as to be easily distinguished. Optimum values were represented by green lines, while the minima and maxima were represented by red lines. (In some versions of this report, the reproduction of the display in figures 2 and 3 is in black and white, so the reader may expect to be unclear as to which line is which. Rest assured the subjects suffered no such confusion.) We also told them what the proportions of the chemicals should be for optimal performance. Their goal was to stay within the minimum-maximum range while trying to maximize their output. They could use the "Danger Count" displayed in the lower left corner of the display (see figure 2) as a measure of how well they were keeping the system below the maximum pressure and temperature limits.

After the subjects were familiar with the system, we asked them to play six games of 100 iterations each. Each of these simulations ran at a speed of either 3.5 seconds per iteration or 7.0 seconds per iteration. Each simulation had either the standard (s) display, the alternate (a) plus the standard display, and the event monitor/editor (e) plus the standard display. So each subject ran each type of simulation at each speed. We gave each subject a particular ordering of types and times (for instance, a-3.5, s-7.0, e-3.5, a-7.0, s-3.5, e-7.0) so as to control the sources of variation and to allow for unconfounded inference.

3.3.3. Results

Each subject generated 600 lines of data, 100 lines per game, where each line gave a snapshot of the the chemical factory's status at each of the 100 iterations in the game. Initially, we focussed our data reduction and analysis on the following variables for each of the six runs of each subject: total danger counts (a measure of how often the subject was above the upper limit of temperature or pressure in a tank), total out-of-bounds count, and total output (how much product chemical the subject produced). We explored the behavior of these variables in aggregate, by person, by trial number (1 through 6), by type of trial (s, a, or e), and by interactions formed from these variables (for instance, output as a function of both person and trial type).

The results are satisfying, but, as with many results in human factors studies, they compete for attention with the overriding result that there is more natural variation between people than there is induced variation between treatments. Our sample of eight individuals exhibited a variety of responses to various displays ranging from total indifference to sophisticated usage. Some examples from observing subjects during their

training and trial periods:

* Some subjects immediately took advantage of the alternate time history displays to spot trends in temperature and pressure, while other subjects found these displays confusing, useless, and eminently ignorable.

* While all subjects mastered the event monitor/editor, it was clear that some immediately turned to it with long lists of events to be warned of, but others took many trials to see the usefulness of even a meager list of events.

* We purposely gave the subjects a modestly ambiguous task -- "maximize production, but stay in bounds" -- but some were far more sensitive to production, while others focussed on maintaining all subsystems within prescribed bounds.

Nonetheless, the essential message is that the type of display, the length of time for decision, the presence of a trainable event monitor, and practice make a significant difference in the number of danger counts recorded and in the amount of time spent out of bounds.

However, the story is different when we look at the total amount of output chemical produced: most of these design variables have no noticeable effect. Two characteristics of the subjects make analysis difficult and highlight the value of a more formal training period; namely, there are both very consistent and very inconsistent individuals in our group of eight and even in the consistent groups we see large differences between a subject's performances.

To summarize the results:

* The display has an effect on danger counts, with event monitor (e) and alternate display (a) outperforming the standard display (s). ($p<.1$)

* The display type does not show an effect on total output.

* Increased time per iteration (3.5 seconds vs. 7.0 seconds) modestly increases output ($.1<p<.2$) and substantially decreases danger counts ($p<.01$).

* The trial number (1 through 6) had no effect on output amount, but it has a marked effect in decreasing the danger count. ($p<.01$, mean danger counts montonically dec- with trial count from 33 to 13.) All this means is that practice helped.

* In particular, the second trial of the event monitor has a sharp drop in danger counts, as does the second trial of the standard display. Interestingly, there is no such difference for the visually richer alternate display.

* Subjects are far more different from one another than they are from treatment to treatment, both in terms of average

scores and in terms of level of variation exhibited.

* We feel that some of these differences among subjects would have been reduced if the subjects had been required to spend more time learning the system.

Each of these results has bearing on further testing and exploration of the effectiveness of flexible, individualised graphical systems. It is particularly in the evidence of wide differences among subjects that we see the value of customisable displays and event monitors in reducing both heterogeneity (differences among subjects' averages) and heteroscedasticity (differences among subjects' variation around their averages). In sum, the system we implemented was not as personalizable as we would like. We feel that future experimentation on the desired system (whose design we have laid out above) would show that the wide variation among subjects that we found would be reduced after they had been properly trained in its use.

4. CONCLUSIONS

We feel that it is self-evident that a personalized graphics workstation would be a powerful assistant to the decisionmaker. This would be due to the following key features of the system as we have designed it: a flexible display system capable of displaying data in a wide variety of graphical formats, and an event monitor that would scan input data for user-defined patterns.

Although we couldn't test the complete system, we tried to isolate the two features described above and test their usefulness. We verified that different display formats were differentially suited to different information tasks. This would validate the desirability of the first feature, the flexible display system. While the results in the chemical plant were not as clearcut, we feel that they indicate that the event monitor would also be a powerful addition to any system that was to assist in the analysis and understanding of large amounts of data. Also, the variability we saw in subjects' performances in the chemical plant experiment underscored the need for the system being personalizable.

We have presented a viable design for such a personalizable, intelligent graphics workstation. We feel that we are capable of implementing it.

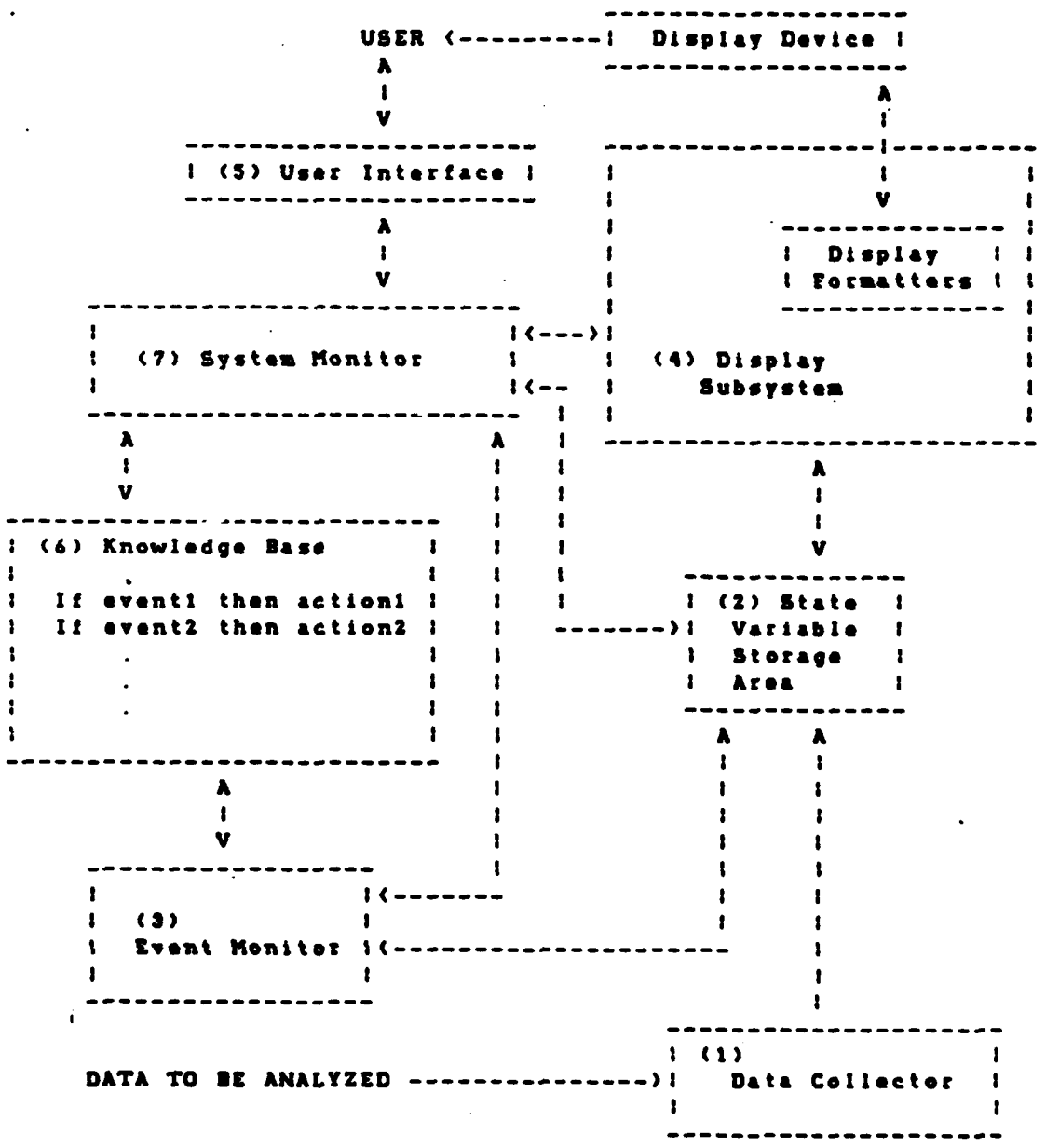


Figure 1. Block Diagram of the Envisioned System. (The connection from System Monitor to Data Collector is not shown. The numbers shown are the numbers of the components described in section 2.2.)

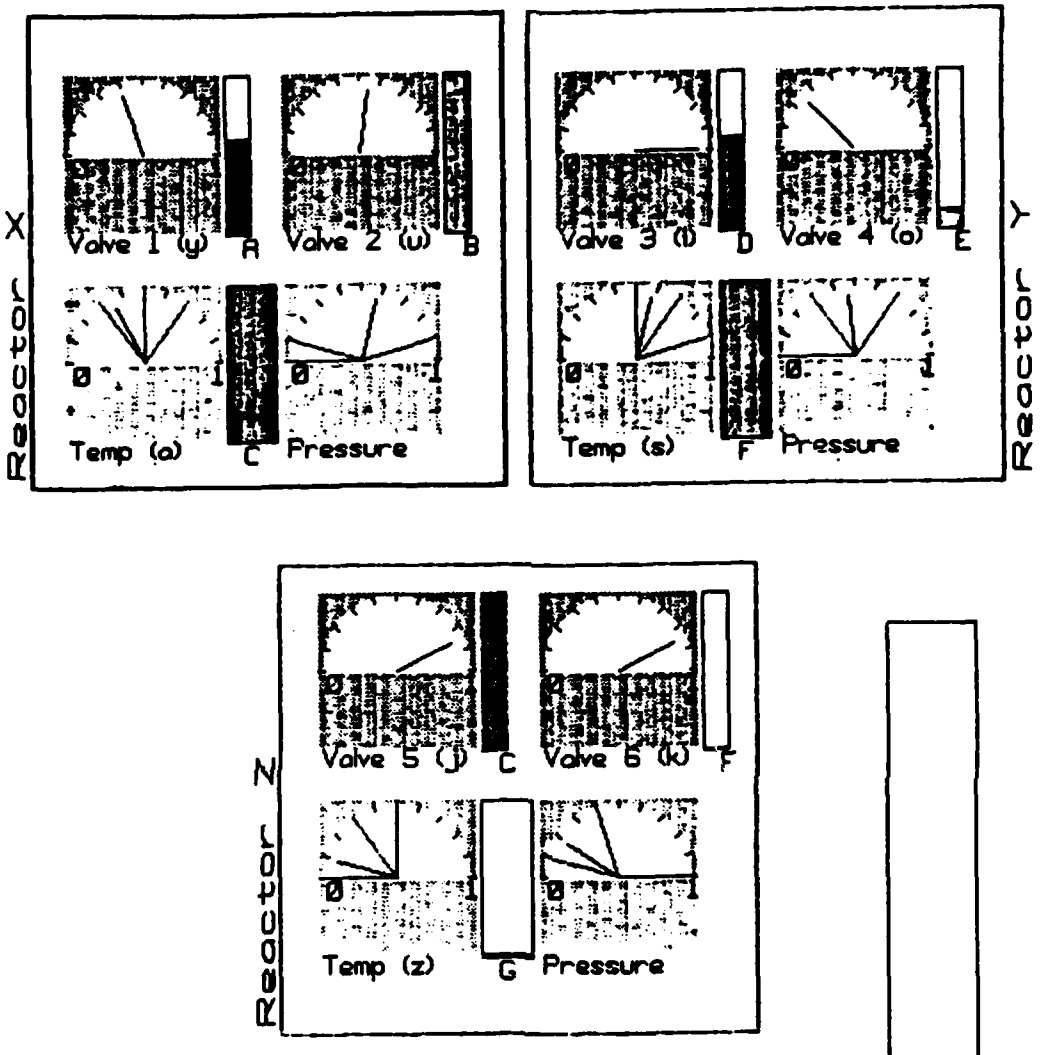


Figure 2. Left half of the display from the chemical plant simulation. The dials display the temperature, pressure, and valve positions in each tank. The colored bars represent the amount of each of the three types of chemicals in each tank.

Danger Count
18

9.1

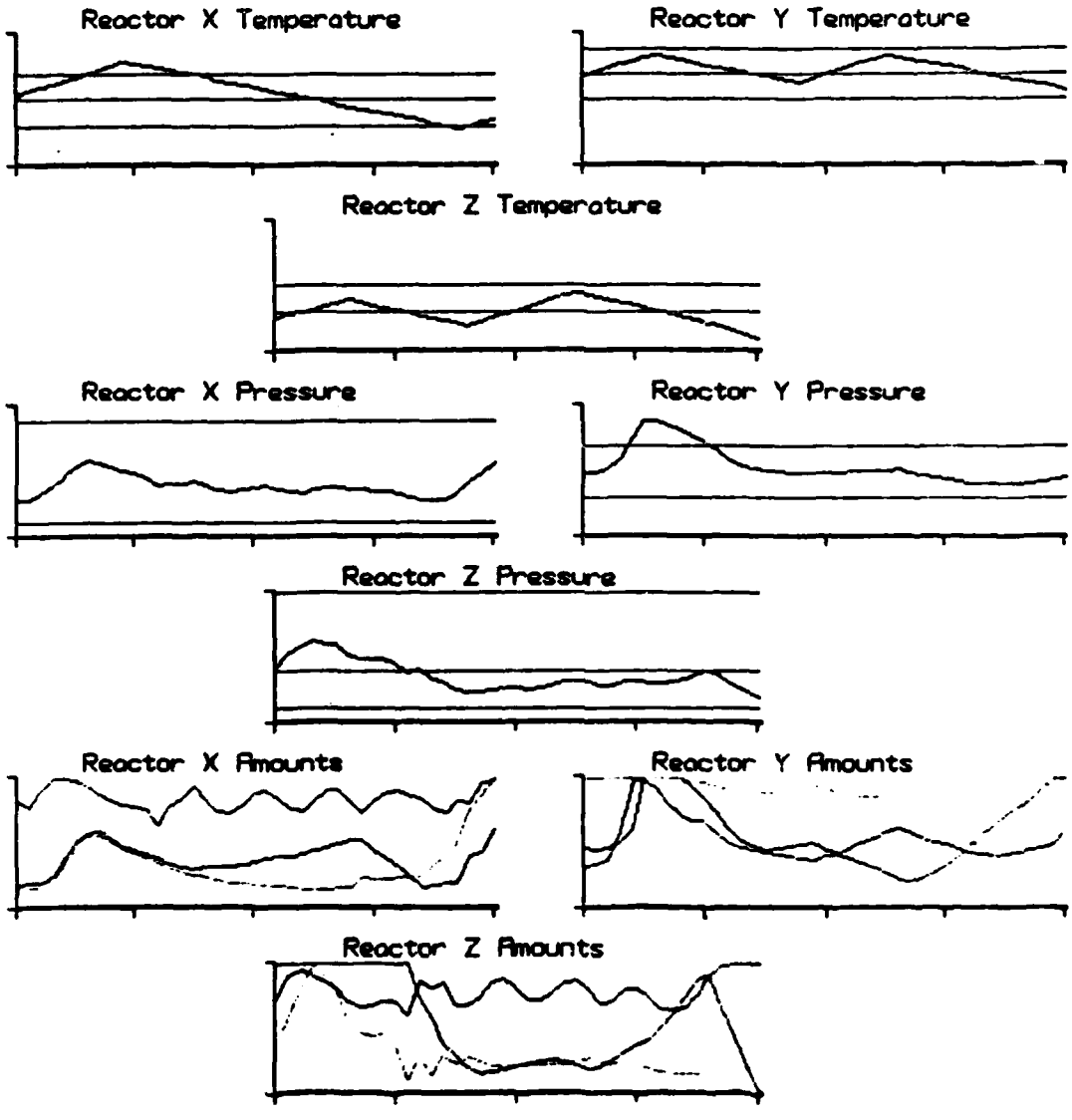


Figure 3. The right of the display from the version of the chemical simulation that had the history displays of key state variables. (Note that in the actual display the lines were in color so as to match those of the left half so subjects had no difficulty figuring out which lines in a graph corresponded to which state variable.)

10	21	25	20	14	19	33	38	24	9	2	1	0	0	0	3
13	25	28	17	8	7	13	15	11	7	5	3	2	1	1	4
9	17	18	11	5	3	4	5	8	14	16	13	8	5	3	3
4	6	7	5	4	6	7	7	13	25	34	32	26	19	11	5
2	2	2	3	9	18	19	14	15	29	45	54	51	39	21	8
2	0	1	4	15	30	32	20	13	22	41	57	58	44	24	8
1	0	1	4	14	27	30	18	8	11	23	35	36	27	14	5
0	0	0	2	7	14	15	9	4	4	9	13	13	9	5	2
0	0	0	0	2	4	4	3	2	3	5	6	5	3	1	0
0	0	0	0	0	1	2	4	5	7	10	11	7	3	1	0
0	0	0	0	0	1	4	9	13	13	15	16	11	4	1	0
0	0	1	2	2	3	6	14	18	15	14	13	8	3	1	0
0	2	5	9	9	7	10	16	17	12	8	6	4	1	0	0
1	4	13	21	21	18	22	26	19	9	3	2	1	0	0	0
2	7	18	28	29	30	42	47	31	11	3	0	0	0	0	0
5	12	20	24	24	31	50	56	36	13	3	0	0	0	0	1

Figure 4. Example of a 16 by 16 array of numbers that was used in Experiment #1 -- the Multiple Formats Experiment.

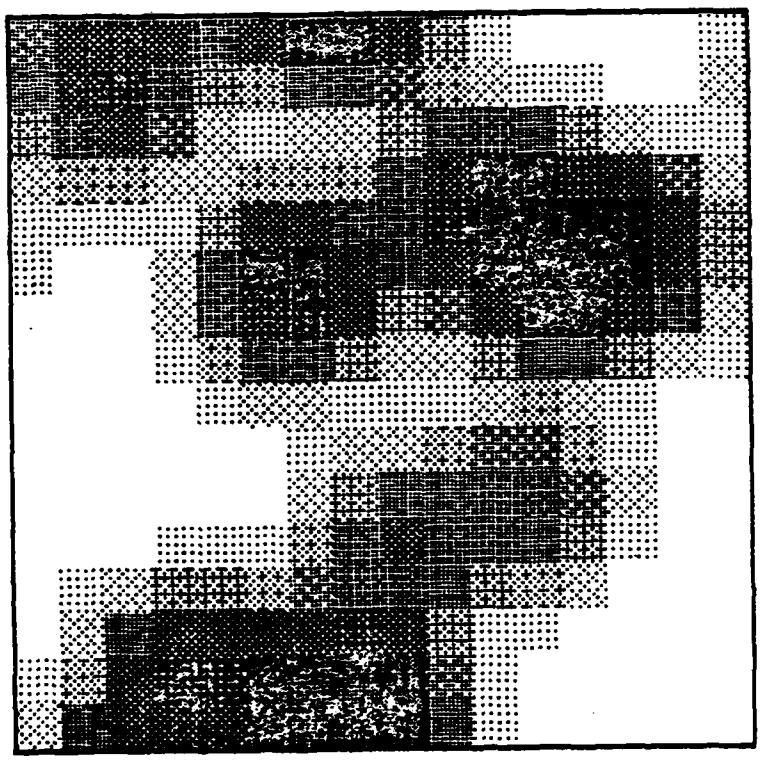


Figure 5. Intensity encoding of data in Figure 4. Higher data values are encoded by darker spots.

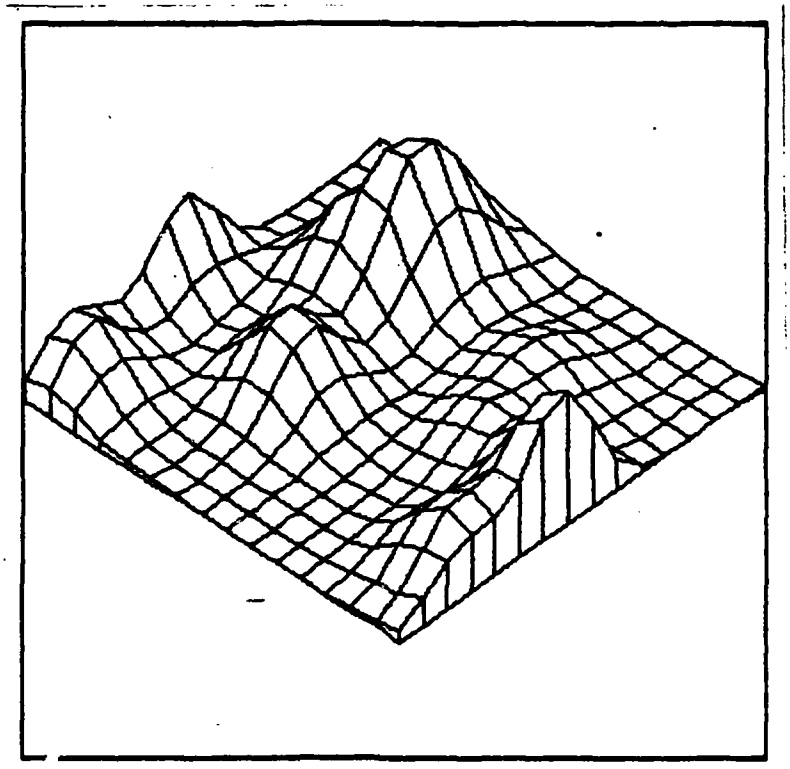


Figure 6. Parallel projection of a 3-D surface that encodes the data values from figure 4. The peaks correspond to local maxima in the data.

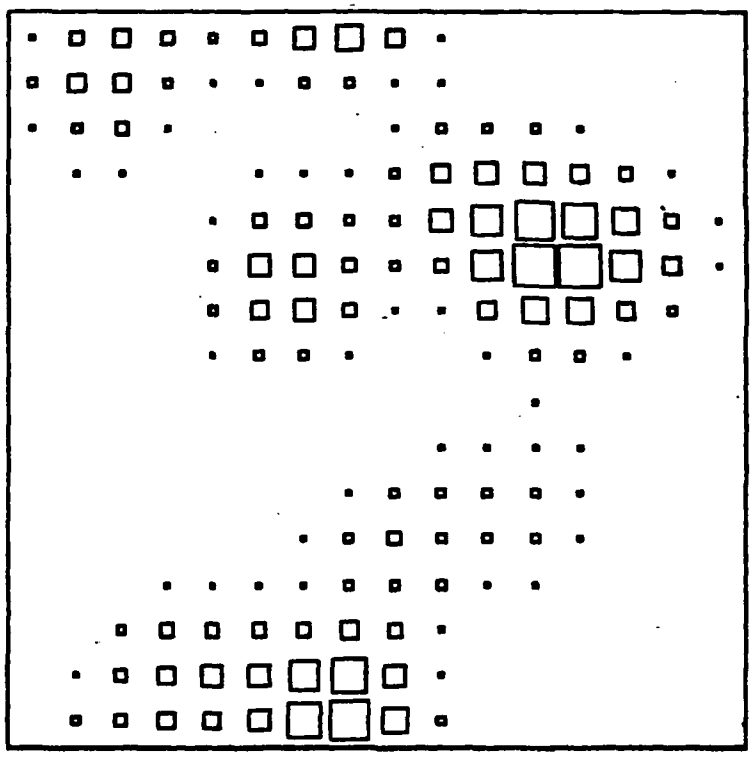


Figure 7. Size encoding of the data in Figure 4. Larger squares correspond to larger data values.

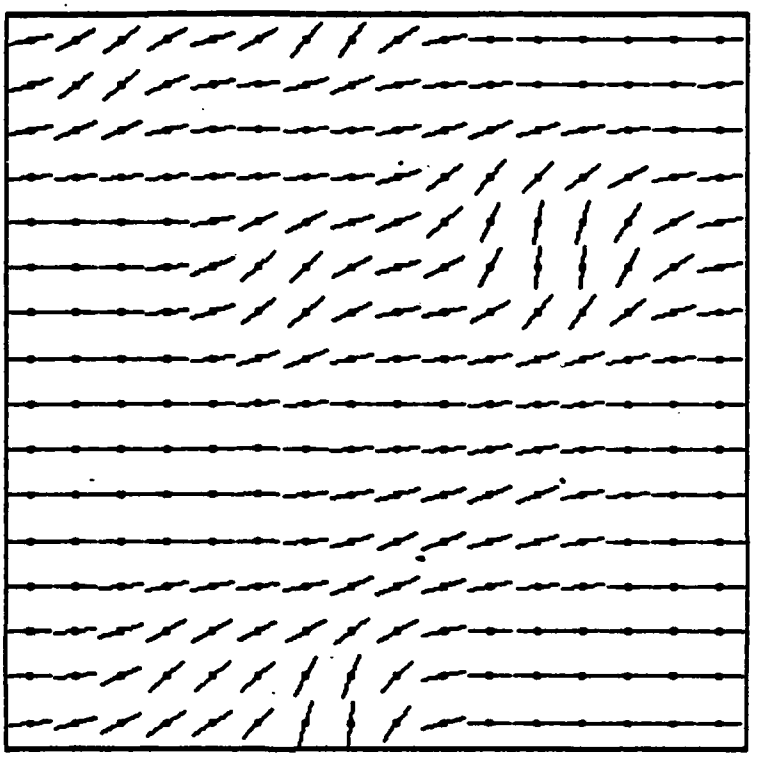


Figure 8. Angle encoding of the data in Figure 4. Horizontal lines correspond to zeroes in the original data. Vertical lines correspond to the maximum possible values in the original data.

OFFICE OF NAVAL RESEARCH

Code 442

TECHNICAL REPORTS DISTRIBUTION LIST

Procurement Contracting Officer
Office of Naval Research
Code 614A: CEA
800 North Quincy Street
Arlington, VA 22217

Office of Naval Research
Code 442EP
800 North Quincy Street
Arlington, VA 22217 (2 copies)

Defense Technical Information Center
Cameron Station, Building 5
Alexandria, VA 22314 (12 copies)

Naval Research Laboratory
Technical Information Division
Code 2627
Washington, DC 20375 (6 copies)

Commanding Officer, ONREAST
Barnes Building
495 Summer Street
Boston, MA 02210