SUPERPOSITION OF GRAPHICS

ON LOW BIT RATE VIDEO

AS AN AID IN TELEOPERATION

by

Mark Vincent Noyes

B.S., Massachusetts Institute of Technology

(1982)

SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS OF THE

DEGREE OF

MASTER OF SCIENCE

IN MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1984

Signature of Author ______________________________
Department of Mechanical Engineering
May 11, 1984

Certified by ______________________________
Thomas B. Sheridan
Thesis Supervisor

Accepted by ______________________________
Warren M. Rohsenow, Chairman
Departmental Commitee on Graduate Studies
Department of Mechanical Engineering

SUPERPOSITION OF GRAPHICS

ON LOW BIT RATE VIDEO

AS AN AID IN TELEOPERATION

by

Mark Vincent Noyes

Submitted to the Department of Mechanical Engineering on May 11, 1984 in partial fulfillment of the requirments for the Degree of Master of Science in Mechanical Engineering

ABSTRACT

A hardware and software system was developed to allow line graphics representing the current geometry of a manipulator arm to be generated and displayed continuously on top of a slowly updated (1.5 and 0.5 frames/sec) video image of the actual manipulator arm and remote environment. The system was used to test the increase in teleoperator performance when using the graphics compared to not using the graphics.

Two manipulation tasks were performed, each with four subjects. Using the model based graphics, subject performance increases ranged from about 50 percent to 150 percent. Additionally, it was seen that when using the graphics, operator performance was relatively insensitive to display resolution and speed of image update.

Thesis Supervisor: Professor Thomas B. Sheridan
Title: Professor of Engineering and Applied Psychology

Acknowledgements

I would like to thank Professor Sheridan for his support in this project. I would also like to give thanks to Hamy Kazerooni, Sam Landsberger, Leon Charny, and Jim Roseborough for their time and effort in performing as test subjects. Also, I would like to thank the Office for Naval Research, Engineering Psychology Group for their equipment support in the project, and Computervision Corp. for their support as well.

Table of Contents

List of Figures and Tables

1. Introduction

1.1 Applications of Simulated Graphics

There are many situations in which it is necessary for a human to observe or interact with a remote environment through closed circuit video. However, there are three specific cases which might make such interaction difficult. These are: 1) communication channel bandwidth limitation, 2) time delays, 3) poor visibility.

If a single video frame consists of 240 lines of 320 pixels, each with 4 bits of gray, and it is desired to have the picture updated at a standard 30 frames per second, this requires a communication channel capable of at least 9.2 megabits/sec. In typical remote control cases, a communication channel might be limited to about 30,000 bits per second, more than two orders of magnitude too slow.

One area of particular interest where the delay is a factor is in space applications. The capability to operate a space manipulator or control a space vehicle from earth is quite valuable. Delays are introduced here both due to the packet switching method of communications used and because of the time delay for the actual transmission. Furthermore, not only is the video delayed, but control signals to a remote manipulator or vehicle are delayed in getting there so that results are not seen until a complete signal round trip has been completed.

As another example, this situation occurs in the control of unmanned undersea vehicles, it may be desirable to allow these vehicles to travel untethered. However, without a tether line, the only viable means of communication is acoustic, and the bandwidth of this medium can be very low when a distance of more than a few meters is present.

There are two components of the delay. One is delay in video from remote site to human operator, the other is from the human operator to actuation of the remote vehicle. In the case of delay in actual video, the picture one sees is what actually happened a fixed time period ago. In the actuation delay case, feedback of what control signals were sent can be current, but the effects of those signals or movements of a remote manipulator or vechicle are necessarily delayed. Of course these resulting movements are not seen until after one round trip after they are made.

The problem of poor visibty is important for submersible craft. Whether tethered or not, when the water in the remote environment is turbid, visibility is limited.

In order to deal with these problems, two different approaches can be taken. The first is to directly eliminate the problem. For example, in the case of limited bandwidth, there are many ways to compress data by an order of magnitude or more for many applications (Haskell and Schmidt, 1975). Another approach is to deal with the problem by providing a display/control aid to the human operator through the use of video graphics. Non-graphic aids using of sound, force feedback, and automatic motion compensation are also possibilities.

1.2 Background Research

Ferrell (1965) has measured performance in remote manipulation with transmission delay, with and without delayed force feedback. Ranadive (1979) has studied the effects of varying frame rate, resolution, and gray scale in performing certain tasks. Deghuee (1980) has furthered that work by experimenting with operator adjustable frame rate, resolution and gray scale tradeoffs.

Fyler (1981) has developed various programs for sensing an environment using a manipulator with touch sensor and then displaying that data in various ways. Winey (1981) has worked on graphics displays in which all of the operator's visual information is given through graphics drawn on the screen, and has experimented with several depth perception cues.

1.3 Problem Approach

In this project, a system was put built to continuously superpose computer generated graphics of the current commanded telemanipulator configuration on a real, intermittently updated (and therefore delayed) video image of the actual configuration of remote arm/hand and environment. This was used to experminent with the effect of such visual aiding on various aspects on teleoperator performance.

The possiblity of data compression in the image transmission was not studied. However, it was indirectly taken into account. In experiments performed, image display speeds were used which would actually require some data compression in order for the same display rate to be achieved in actual practice.

2. Hardware-Software Package Description

2.1 Hardware

Manipulator

In this experimental system a model E-2 master slave manipulator built by Argonne National Laboratory was used. The master and slave were geometrically similar and each had seven degrees of freedom including end effector gripping. This is shown in figure 2-1. The manipulator can be seen to consist of three distinct sections: shoulder, forearm, and hand. The shoulder has two degrees of freedom in angles 1, 2. The forearm has two degrees of freedom in angles 3,4. The hand has 3 degrees of freedom in angles 5, 6 plus the gripper opening.

PDP-11 Computer

A PDP-11/34 computer running the RSX-11M operating system was used to do most of the calculations. Through an MDB DR11-C parallel I/O board, the PDP-11 communicated with an Analogics AN5400 series interface system. With this interface system, the PDP-11 had 32 A/D and 32 D/A channels, as well as two each 16 bit input and output ports.

LSI-11 Computer

As an auxiliary system, we used an ADAC LSI-11/23 computer system running stand alone FORTH in 32 Kb of memory. This system handled all video and graphics displays.

Installed in the LSI-11 system was a custom dual-ported memory board to allow the LSI-11 to receive data from one of the PDP-11's 16 bit output ports. This memory board contained only 1K word of memory, but was ample for its designed task.

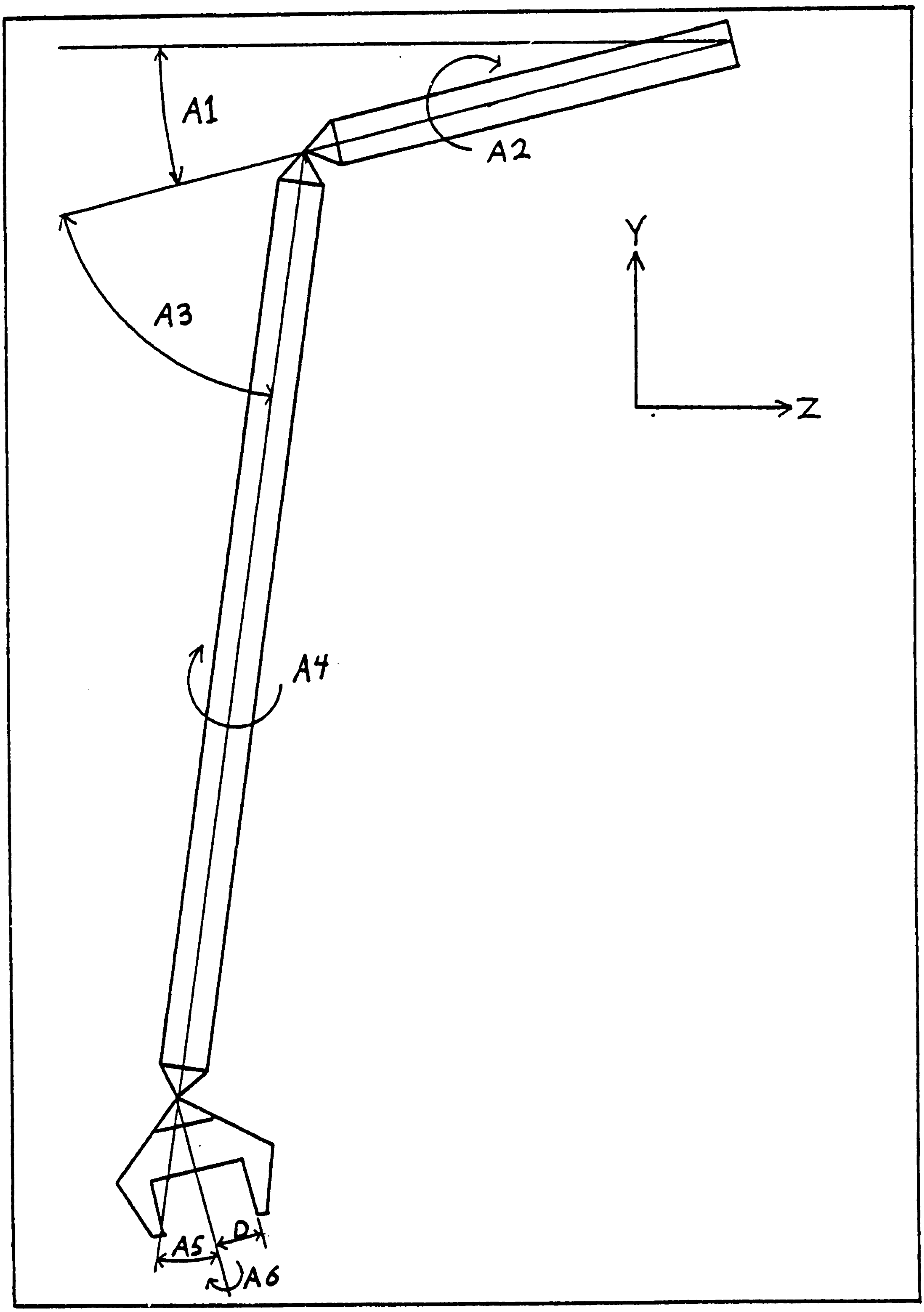

Figure 2-1
E-2 Manipulator

Also installed in the LSI-11 system was a Datacube QAF120/QVG120 video digitizer. This system provided a 320x240 pixel display with 8 bits of gray scale. This video digitizing board could digitize video frames, display them, and allow the frames to be frozen and then written to and/or read from. The Datacube board got its video input from a Panasonic video camera. Installed in the LSI-11 system was a Lexidata DMA interface to a Lexidata series 3400 display system.

Lexidata Display System

The Lexidata graphics display was a 640x512 pixel raster system with 12 independent bit planes. The system also had 4 1024x8 bit lookup tables for the three RGB outputs and the monochrome output. The system provided independent write enable masks for each of image, graphics, and text. The output of the system went to a Mitsubishi 19 inch RGB monitor. No color was used in these experments however.

2.2 Software

Display Capabilities

The program used provided the capability to display both a slowly updated video image, and a rapidly updated graphics image on the Lexidata display at the same time. The graphics and video were displayed at the same pixel resolution at all times.

The pixel resolution could be set to either 320x240 pixels, of 160x120 pixels. The image in low resolution was obtained by digitizing the image in 320x240 format, and dividing the image into 2 pixel square blocks. For each block, only one of the four digitized pixels was displayed. Averaging of the pixels was possible, but would have slowed the system substantially.

The image was digitized to 8 bits of gray, and all 8 bits were used in the display, although some resolution could easily have been masked out.

The graphics overlay was a wire frame model of the arm, plus a possible simulated shadow of the hand. The image was drawn on the display at an increment of 50 (on a scale of 0-255) intensity levels above the underlying image pixel intensity.

Database

As described previously, the manipulator used consisted of three segments. The program assumed a knowledge of the basic manipulator geometry. However, the exact representation was loaded into the program at run time from a data file. The display data consisted of point and connectivity data for each manipulator segment. No hidden line removal was attempted, and no filling or shading was done. The input data file consisted of:

1) count of points defining the hand geometry
2) points defining the hand geometry
3) count of points defining the forearm geometry
4) points defining the forearm geometry
5) count of points defining the shoulder geometry
6) points defining the shoulder geometry
7) the number of vectors representing the hand
8) list of endpoint indices for the vectors representing the hand
9) the number of vectors representing the forearm
10) list of endpoint indices for the vectors representing the forearm
11) the number of vectors representing the shoulder
12) list of endpoint indices for the vectors representing the shoulder
13) the number of points used in shadow generation
14) indices into list of geometry points to be used in shadow generation.
15) the number of vectors in shadow
16) shadow vector endpoint list

The hand data points were indicated in a special format. In addition to an x,y,z, each point also included a flag indicating which direction, if any, the point moved in response to opening and closing of the hand grippers.

The coordinates given were in a coordinate system with the origin at the point connecting the manipulator segment to the next segment up the arm. In addition, the coordinate system was parallel to the global coordinate system when angles A1-A6 were all zero.

Manipulation Capabilities

A number of capabilities were incorporated into this software package for manipulating the display system. First, the graphics could be zoomed in or out interactively through the keyboard. Also, all three of the orientation angles for viewing the simulated graphics could be varied. The main purpose for these two capabilites was to allow the placement and scaling of simulated graphics to be manipulated to match the actual environment.

One feature which made the interactive scaling and rotating less necessary was the capability for semi-automatically adjusting the graphics to the environment. By identifying two locations on the screen with the actual manipulator, the graphics adjusted themselves correctly to within the accuracy that the operator identified the screen locations.

There also existed the capability for interactively changing the video and graphics resolution. With this, the screen update rate could be varied at a tradeoff for picture resolution. Also, the method of image update could be changed to allow either continuous image update, or buffered image update (in continuous update, the video image pixels were displayed as they were received, while when buffered, the picture was not displayed until complete).

Overall Software Scheme

The PDP-11 was responsible for monitoring the manipulator configuration and generating graphics to represent it as well as monitoring operator input through the keyboard. When the PDP-11 had generated a new set of graphics for the display, it sent them to the LSI-11 system via the dual ported memory board and indicated to the LSI-11 system that the graphics were ready.

Meanwhile, the LSI-11 system was continuously reading data from the Datacube video digitizer board, buffering it, an sending it to the Lexidata display. Upon receipt of new graphics information, the LSI-11 formated the graphics information and sent it to the Lexidata display. Also, each transmission could include optional information regarding change of display mode.

The technique of reading from the Datacube was used for two reasons. First, it was desired to simulate a low bandwidth communication channel, and secondly, its flexibility allowed the display to have its resolution and mode of update changed.

Flowcharts of the programs execution are shown in appendix C.

3. Display Theory

3.1 Graphics Display Techniques

In the system described here, it was necessary to be able to draw vector graphics and display a video image concurrently and independently. If a video image was present on the screen, we needed to be able to draw graphics on top of the image non-destructively, so that if the graphics were removed, the video image remained. To do this required that the display system treat graphics and video separately. Four basic options existed to allow this: 1) merge graphics and video optically, 2) merge graphics and video electrically after signal generation, 3) use of a raster display which supported independent graphics and image planes, 4) use a display which allowed raster and vector graphics to be intermixed.

One possible method of merging graphics and video optically is through the use of a half silvered mirror. This is shown schematically in figure 3-1. This process involves display of the graphics and video image on separate monitors. Then the half silvered mirror allows the virtual image of one to overlay the actual image of the other.

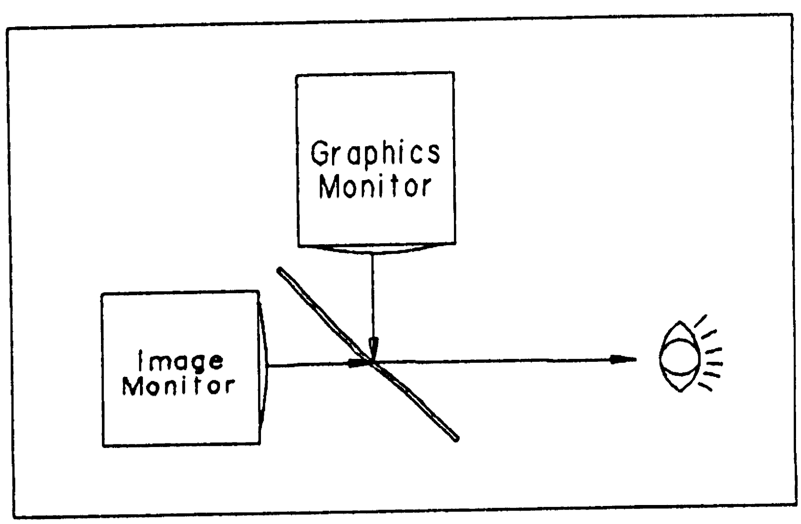

Figure 3-1
Optical Merging of Graphics and Video

A similar method of optical merging is achieved by projecting multiple video images onto the same screen. Here, the images displayed cause the appearance of overlapping.

Alternatively, we can generate the video image on one device and the graphics on a second device, then mix the two signals before entry into a monitor. With this, two separate generation systems produce their own signals from their own data sources. Upon mixing, the two signals become one, and only one monitor is necessary.

A third method of combining the graphics and image is through the use of a specialized display which can display both a raster image and stroke refresh graphics concurrently. In this way, the display system alternately displays a frame of raster-graphics, and then a frame of stroke-graphics.

A fourth possible method of combining the graphics and image is to use a single raster graphics display. Here, the display must be time multiplexed so that both image and graphics can get onto the screen.

Each of these methods has its advantages and disadvantages. The first method has the advantage that it permits the graphics and image to be treated entirely separately. This is still probably the least viable solution to the problem. One problem with this technique is that it requires careful alignment of the display monitors and optical equiptment. Further, even if properly aligned physically, the individual pictures can still wander on the screen due too change in electrical properties of components due to temperature.

The second method shares the advantage of the first in that it has two separate display systems working independently. It also has the advantage of eliminating physical alignment problems, and most electrical drift problems. However, it requires that the two systems used must work in close synchronization with one another. All of the display parameters (e.g. number of lines, number of pixels, horizontal scan time, etc.) associated with the display have to be the same. This generally requires that the two systems be set up in a master/slave configuration.

The third method has the advantage that it requires only one display system and monitor, and eliminates all need for alignment, electrical compatibility, etc. It also has advantage of being able to display stroke refresh graphics. The main point of this is that such systems typically require very little compute time for display, thus leaving more time for raster graphics. Undoubtedly the biggest drawback of such systems is the lack of supply and their high cost.

The last method also has the advantage of requiring only a single display, but it does have the drawback that it requires time multiplexing of image and graphic information sent to the display system.

For our purposes, we elected to use the last method for the advantages outlined, and the fact that it was more readily available than either methods two or three.

Using the method described above, two different display resolutions were used. In high resolution, the display was 320 pixels wide by 240 pixels high. In low resolution, the display was 160 by 120. Figure 3-2 depicts the actual display when in 160 by 120 resolution.

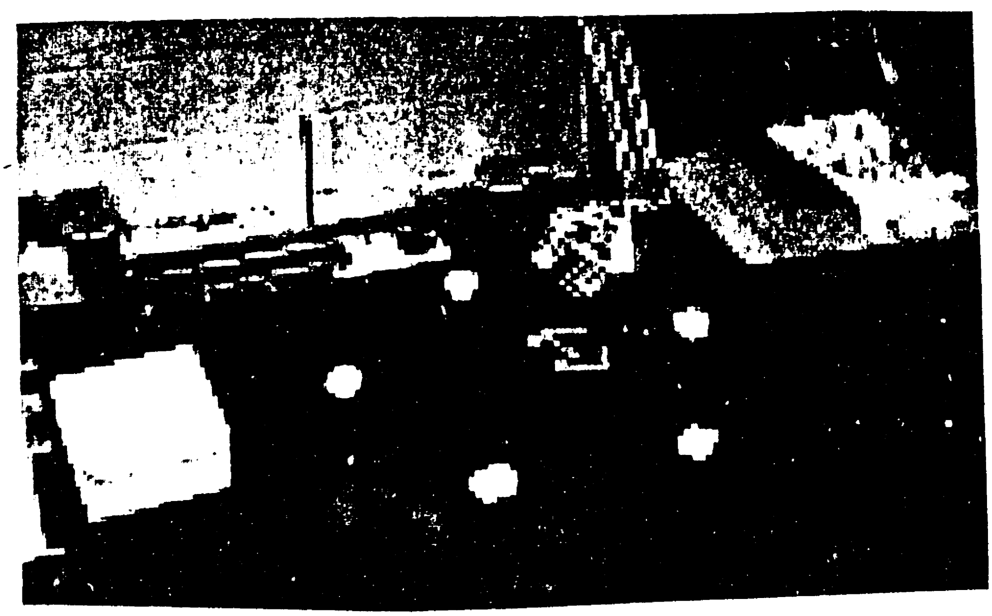

Figure 3-2
Low Resolution Display

3.2 Depth Perception and Realism

Two techniques were used to enhance realism and depth perception in the graphic images. The first technique was perspective projection

(Sproull and Newman, 1979). The other technique is the generation of a simulated shadow of the hand of the manipulator. The shadow had its light source fixed directly above the manipulator, but arbitrary directions could have been implemented if desired.

3.3 Display Considerations

One interesting aspect of transmitting video over a low bandwidth link is the compromise of pixel resolution with screen update rate. As studies have proven , both good picture resolution and fast screen update rate contribute to better remote manipulator operator performance (Ranadive, 1979). When deciding on the best trade-off of resolution versus speed of update, one should consider effective graphic overlay techniques. In this project, the hardware constraints were such that in order to provide reasonable screen update times, it was necessary to force the graphics resolution to be the same as the video image resolution. This problem could be circumvented with proper hardware (i.e. independent zoom factors for separate bit planes).

A second trade-off problem concerns method of image buffering. Upon receipt of video image data, one of two things must be done with it: 1) it may be displayed immediately, or 2) it may be buffered until a complete image has been collected, at which time it can be displayed. The first technique is simpler, and allows the operator to see some of the new image sooner, but his picture is always changing in one area. The second appears to be much less distracting, and therefore easier to work with.

4. Experiments Performed

4.1 Task Descriptions

Two experiments were run in order to test the effectiveness of the graphics superposition as an aid to the operator under time delay and bandwidth constraint.

In the first experiment, the operator's task was to pick up five white one-inch-square blocks distributed roughly in a 16 inch diameter circle on a dark table, and place them into a white box eight inches square and four inches high. This task will be referred to as the BLOCKS task, and is depicted in figure 4-1.

In the second experiment, the operator had to follow a fixed contour with a nominal time limit for task completion. This task will be referred to as the TRACKING task. This is depicted in figure 4-2.

4.2 Task Scoring

Scoring for the BLOCKS task was based on time to completion. If all five blocks were successfully placed in the box, the score was the actual time to completion. If some of the blocks, or the box were knocked out of reach, the score was scaled by the following formula

$$\text{Score} = \text{Time} * (1 + 1.1 * (5 - N) / N)$$

where N is the number of blocks succesfully placed in the box.

In the TRACKING task, the score was based on RMS error and time for completion. The score for this task can be expressed as:

$$\text{Score} = \text{RMS} * W1$$

where RMS is the root mean squared error and W1 is described by Figure 4-3.

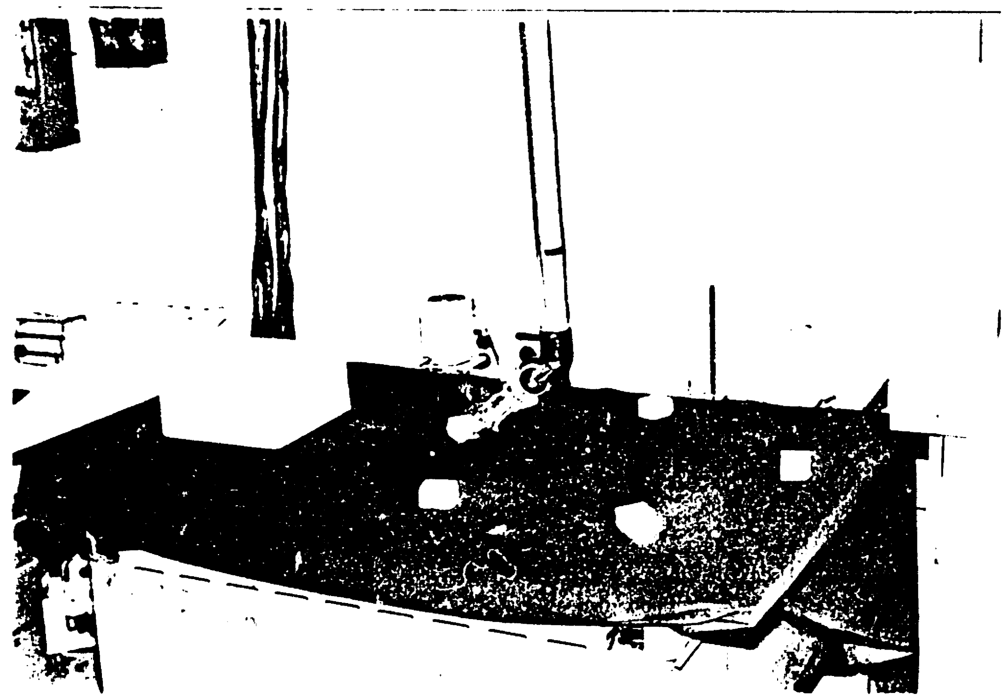

Figure 4-1
BLOCKS Task

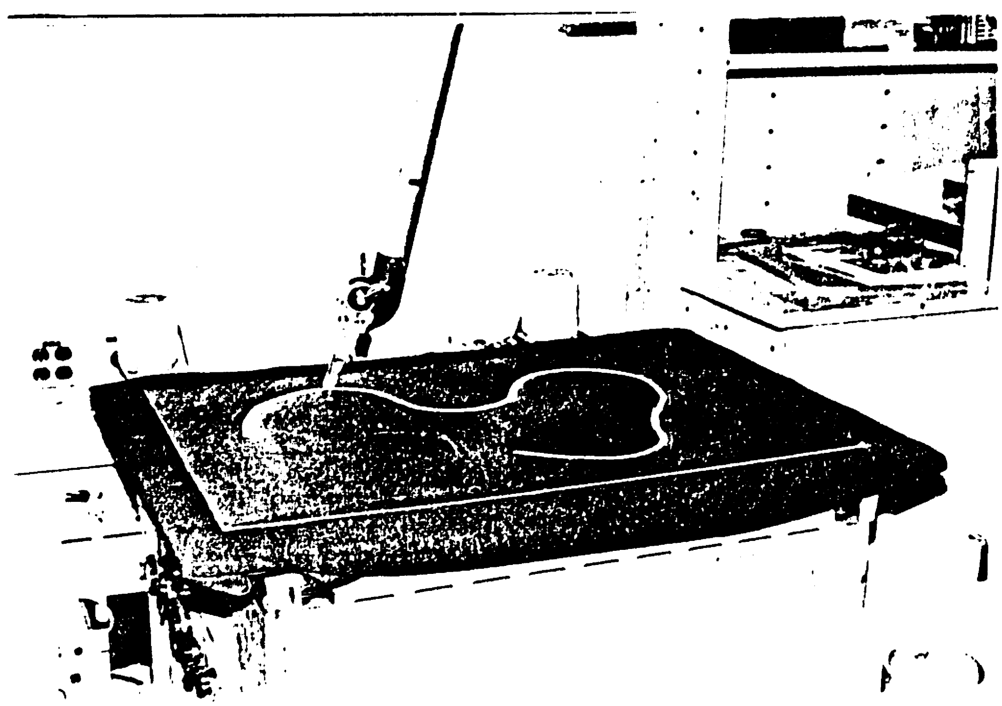

Figure 4-2
TRACKING Task

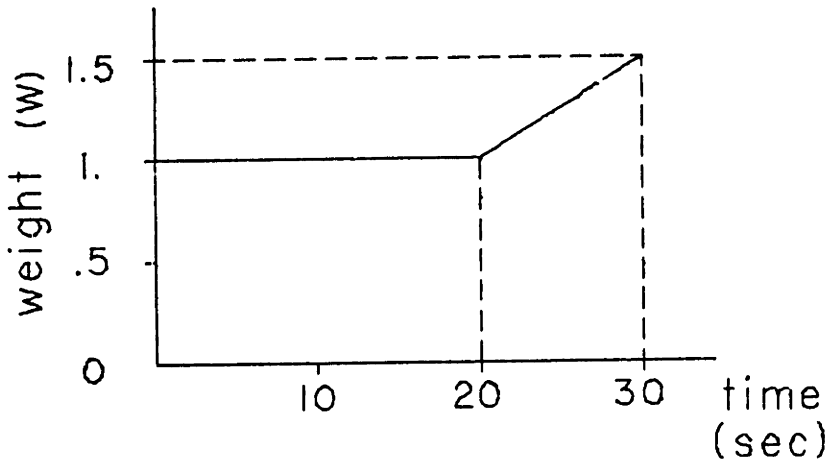

Figure 4-3
TRACKING Task Weighting Function

Times greater than 30 seconds were discarded as bad data. The rationale for W1 is that for times to completion which are less than the nominal a performance penalty will be self imposed by the operator. For times greater than the nominal a penalty factor is enforced.

The BLOCKS task was chosen as a task which involved both motion and manipulation, whereas the TRACKING task involved only motion. These two basic skills are seen by the author to be the two major types of tasks to be done by teleoperation.

4.3 Test Modes

In each experiment, the tests were performed in nine different modes. These modes are described by table 4.1. In this table MODE gives a number by which the specific mode will be referred. GRAPHICS? indicates whether graphics were superimposed and CONT./BUFF. indicates whether the display was run with the video image continuously updated (continuous) or only following the completion of a new video frame (buffered). Note that in mode 9, the image was displayed on a standard video type monitor with a refresh rate of 30 Hz. RESOLUTION indicates whether the image resolution was 320X240 (HIGH), or 160X120 (LOW).

In the low resolution mode, the screen update was performed at about 95 frames per minute and in high resolution mode the screen update was performed at a 33 frame per minute rate. Since each pixel was digitized to 8 bits of gray scale the equivalent bit rates for these displays were 240k bps for low resolution and 336k bps for high resolution.

These bits were not necesdarily all useful. It has been shown (Ranadive, 1979), that beyond 5 bits of gray scale, very little is gained by adding more gray scale. Since the data displayed had 8 bits of gray, equivalent displays could have been shown which would only require 5/8ths of the equivalent bit rate used. Assuming this equivalence, the bit rates used were more realistically 150k bps and 210k pbs.

As a result of preliminary tests of the displays, it was decided that the simulated shadow was detrimental to operator performance in the TRACKING task, and therefore was omitted. However, the simulated shadow was included in the BLOCKS task.

MODE	GRAPHICS?	CONT./BUFF.	RESOLUTION
1	YES	CONT.	HIGH
2	YES	CONT.	LOW
3	YES	BUFF.	HIGH
4	YES	BUFF.	LOW
5	NO	BUFF.	HIGH
6	NO	BUFF.	LOW
7	NO	CONT.	HIGH
8	NO	CONT.	LOW
9	NO	NONE	HIGH

TABLE 4-1
Test Mode Description

4.4 Subjects and Experimental Design

Four volunteer subjects were selected to perform these experiments. Some of the subjects had used the manipulator occasionally previous to the testing, but none had a great deal of practice in tele-operation.

In the testing sessions, each subject performed each task five times in each of nine different modes for a total of forty-five repetitions of each task. The two tasks were completed separately and a brief rest period was given between sessions.

In preparation for the actual task, each subject was given instructions as to how the experiments would be scored, and the rules for each task. Each subject was then put through a training session of about one and a half times the expected testing time.

Test orderings were varied across subjects. Table 4.2 gives the testing orders for each of the four subjects.

The TRACKING task used the same contour repeatedly. This repe-tition of the same contour posed a problem. If the test subjects had been given too much training, their ability to perform the task would have become more of a test of their memory than of their ability to effectively interact with the test environment. However, it was necessary to give the subjects sufficient experience to permit them to work efficiently.

The single contour was chosen despite the training problem. This was done for two reasons. The first was that generating new contours for each trial was difficult to do efficiently. The second was that the memorization of the contour was most likely to help the operator in the cases without graphics, and therefore the estimates of increase in operator performance were likely to be conservative.

	SUBJECT 1		SUBJECT 2		SUBJECT 3		SUBJECT 4
	1		2	T	3	T	4
B	7	B	8	R	5	R	6
L	5	L	4	A	7	A	8
O	3	O	6	C	1	C	2
C	2	C	5	K	2	K	1
K	6	K	3	I	8	I	7
S	4	S	1	N	6	N	5
	8		7	G	4	G	3
	9		9		9		9
T	3	T	4		1		7
R	6	R	5	B	7	B	1
A	7	A	8	L	5	L	6
C	2	C	1	O	4	O	4
K	1	K	2	C	3	C	5
I	8	I	7	K	6	K	3
N	5	N	6	S	8	S	8
G	4	G	3		2		2
	9		9		9		9

TABLE 4-2
Test Mode Ordering

5. Experimental Results

After completion of the experiments, the means and standard deviations of the scores were calculated. Figures 5-1 through 5-8 show the results for each subject in both the TRACKING task and the BLOCKS task. Figures 5-9 and 5-10 show the overall results. In these figures, the means and plus and minus one standard deviation of the raw data, and plus and minus one standard deviation of the mean are shown. The uppermost and lowermost bars represent the standard deviation of the raw data, the middle bars represent the standard deviation of the mean, and the center dot represents the mean.

The first important result which can be seen in the graphs is the relationship between the performance with and without graphics. As can be seen, the increase in performance achieved by using graphics is generally in the range of 50 to 150 percent. In no case was the performance with graphics worse than without graphics. The consistency of this result was unexpected at the start of the project.

The second result is the effect of graphics on performance variability across different modes. When graphics were on, performance tended to be independent of the particular mode of operation. The ratio of best performance to worst performance for a given operator in a given task was generally in the range of 1.2 to 1.5. When graphics were off, performance varied widely with the particular mode of operation. The ratio of best performance to worst for a given operator in a given task was generally in the range 1.6 to 2.0.

A third result is that when graphics were off, performance was much better in low resolution, fast update than in high resolution, slow update. Table 5.1 lists the order of performance from best to worst for each subject in each task when the graphics were off. Of the 16 positions in the left hand side of the table, 14 are occupied by either mode 8 or mode 6. This means that 88 percent of the time, low resolution with faster update performed better than high resolution with slower update rate.

Subject	Task	best			worst
1	TRACK	8	6	7	5
1	BLOCK	8	5	6	7
2	TRACK	8	7	6	5
2	BLOCK	8	6	7	5
3	TRACK	6	8	7	5
3	BLOCK	6	8	7	5
4	TRACK	5	6	7	8
4	BLOCK	8	6	7	5

Table 5-1
Performance Ordering Without Graphics

In the tests using graphics, an alternative test might have been to display a background image once, and then only change the graphics in the picture. This probably would have made no difference in the TRACKING task, but might have lowered performance significantly in the BLOCKS task.

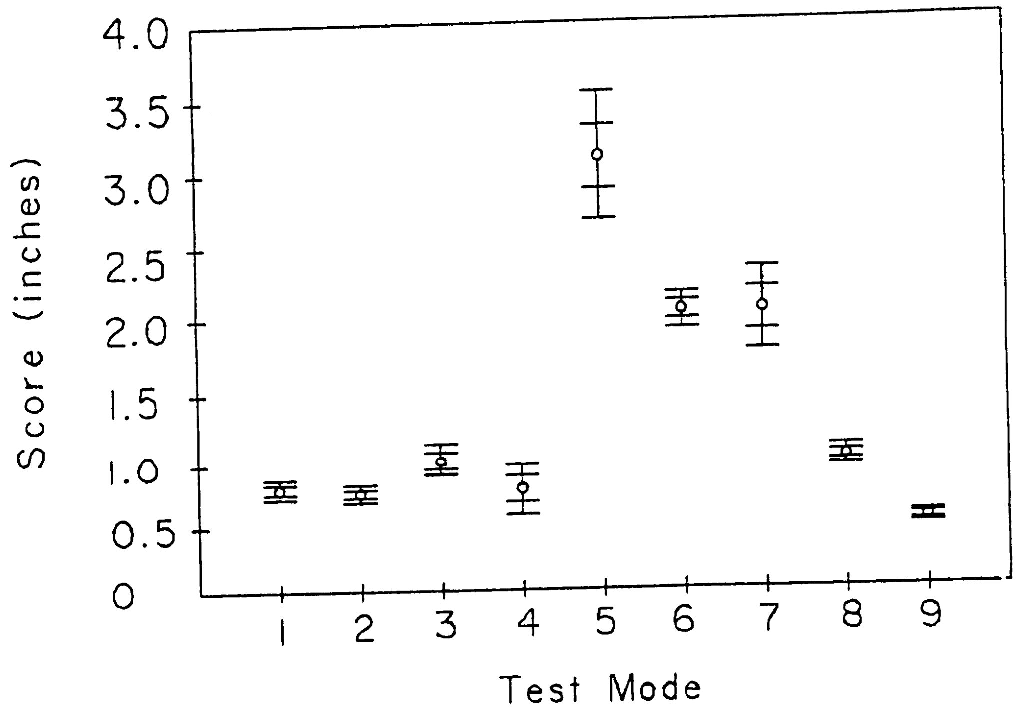

Figure 5-1
Subject 1 TRACKING Results

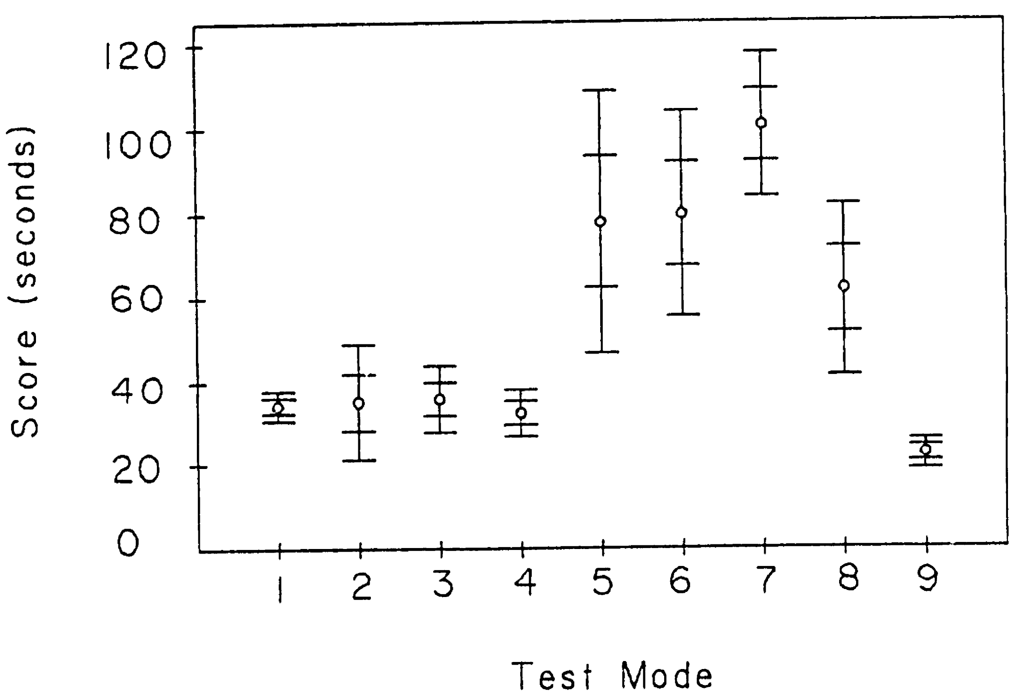

Figure 5-2
Subject 1 BLOCKS Results

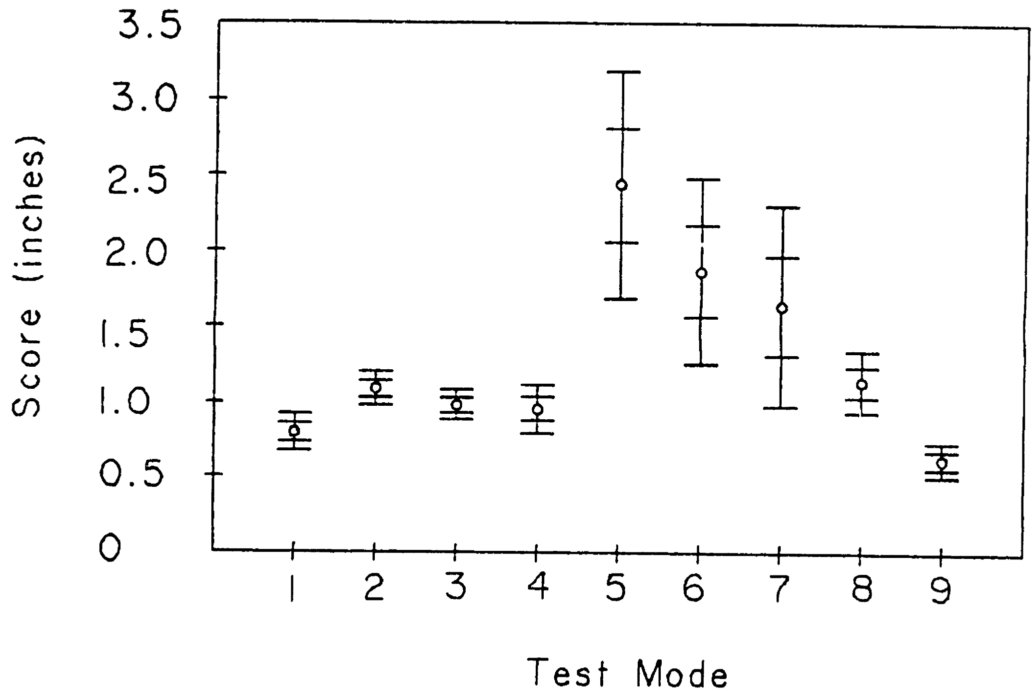

Figure 5-3
Subject 2 TRACKING Results

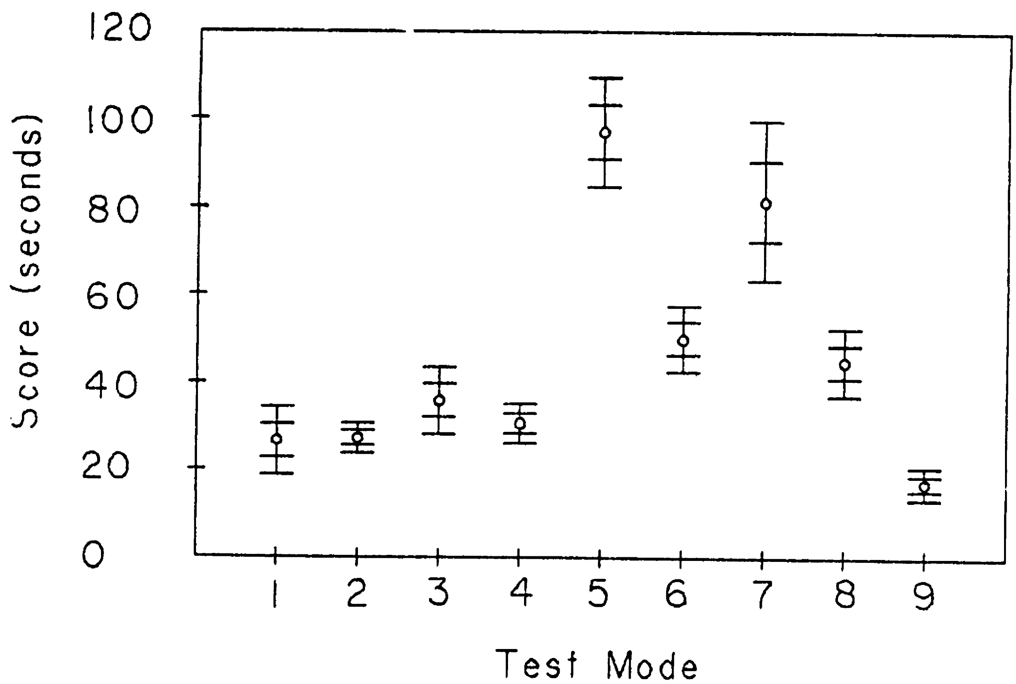

Figure 5-4
Subject 2 BLOCKS Results

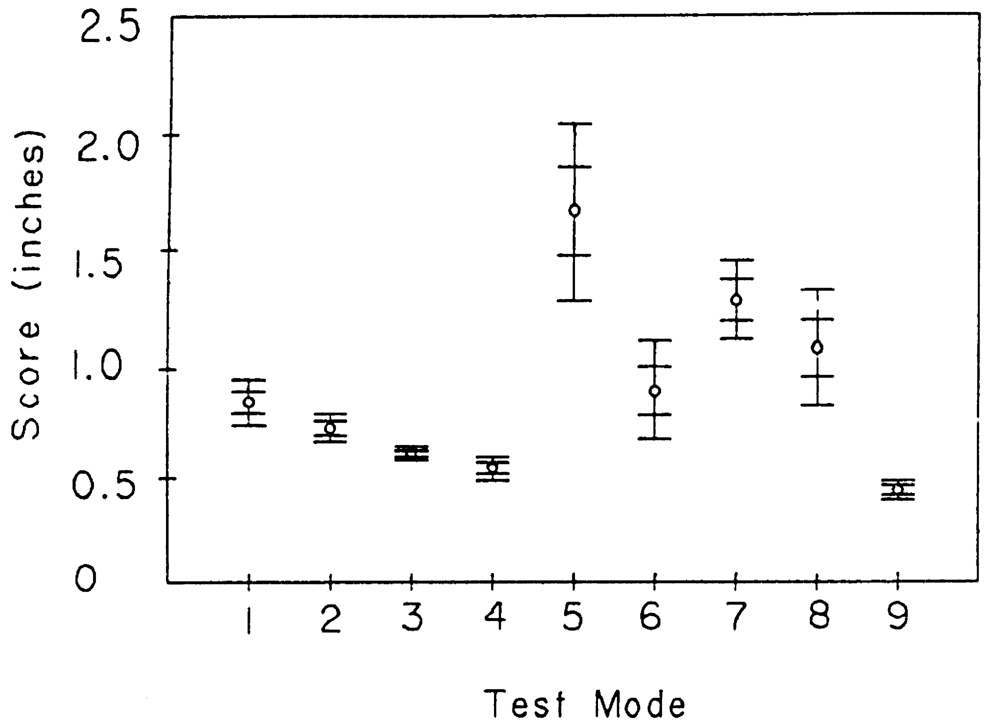

Figure 5-5
Subject 3 TRACKING Results

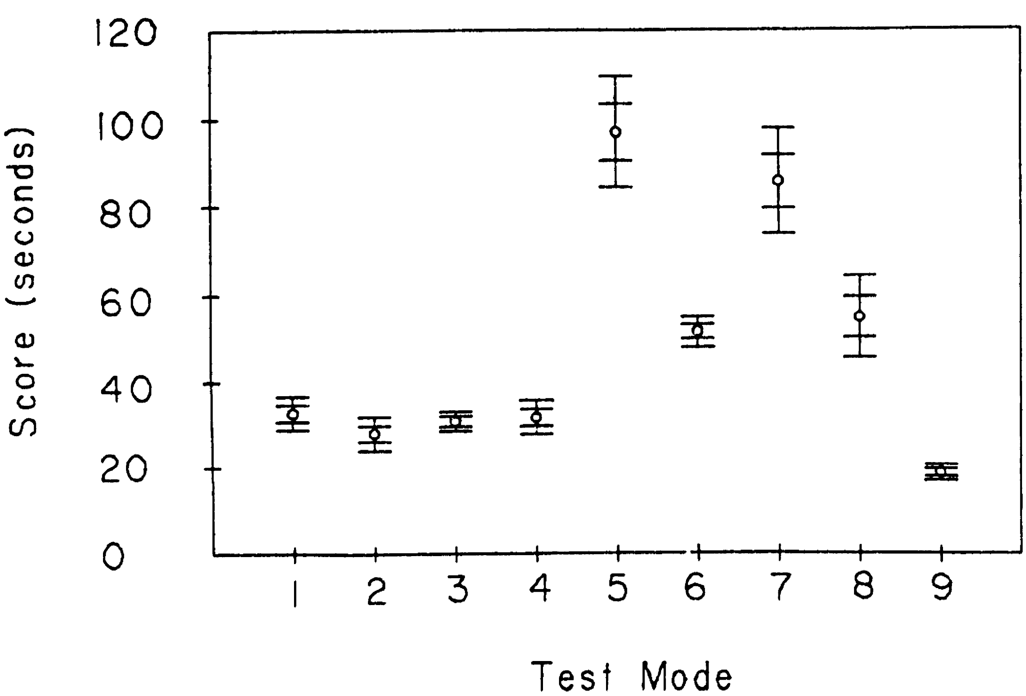

Figure 5-6
Subject 3 BLOCKS Results

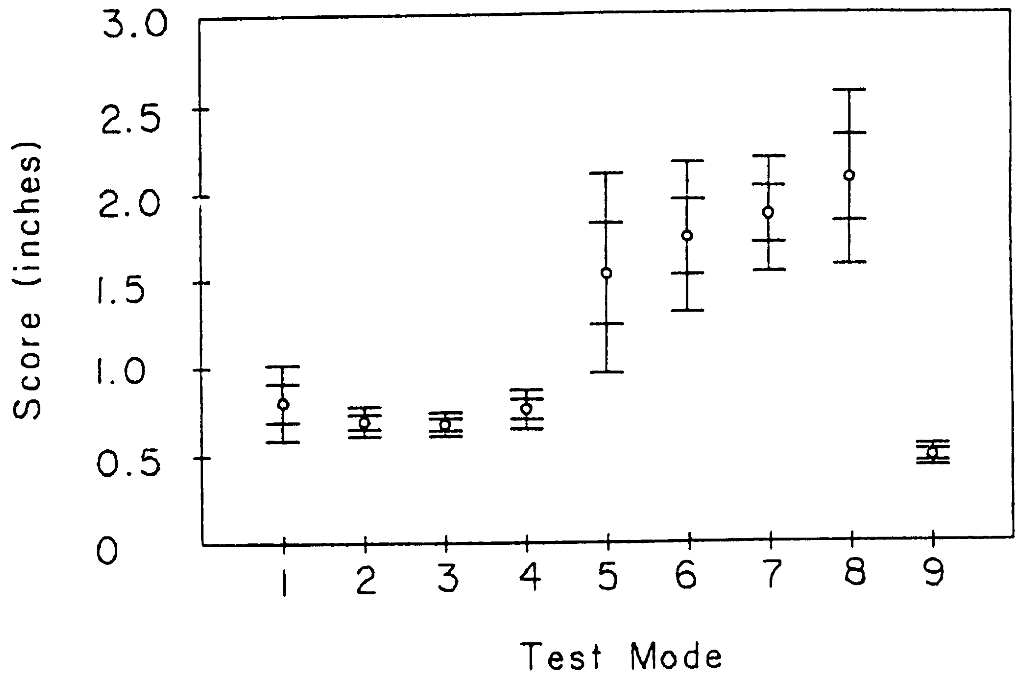

Figure 5-7
Subject 4 TRACKING Results

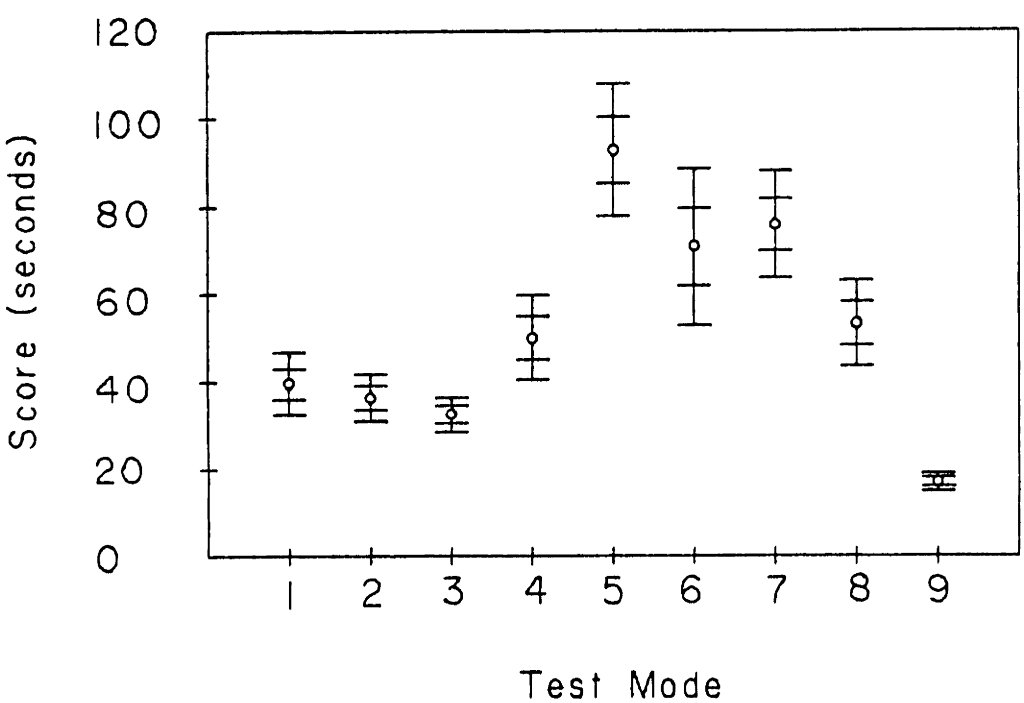

Figure 5-8
Subject 4 BLOCKS Results

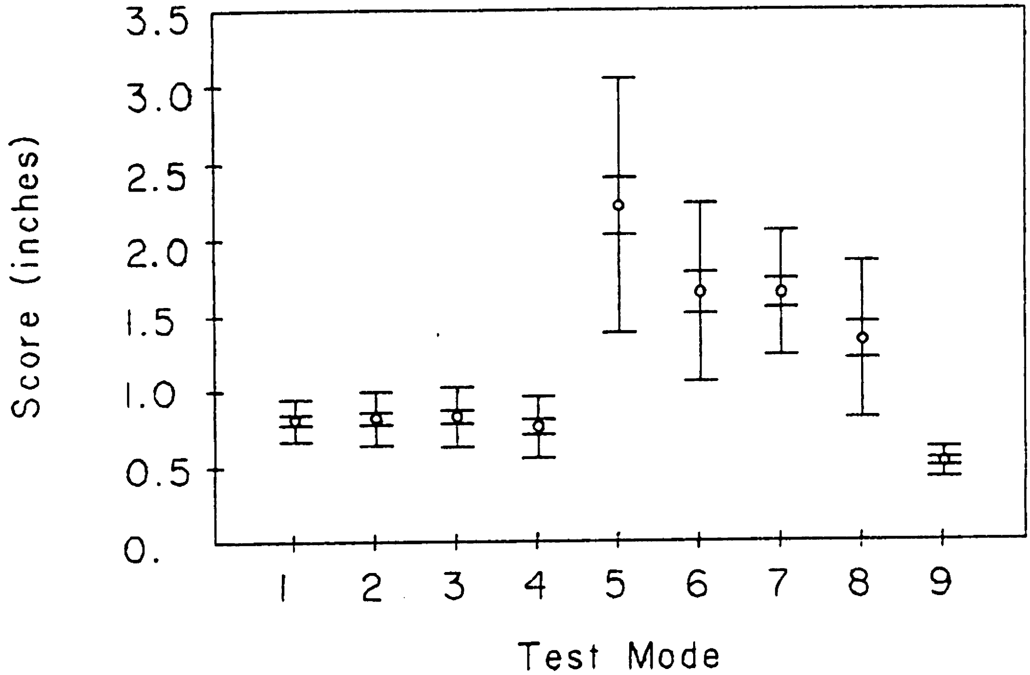

Figure 5-9
Overall TRACKING Results

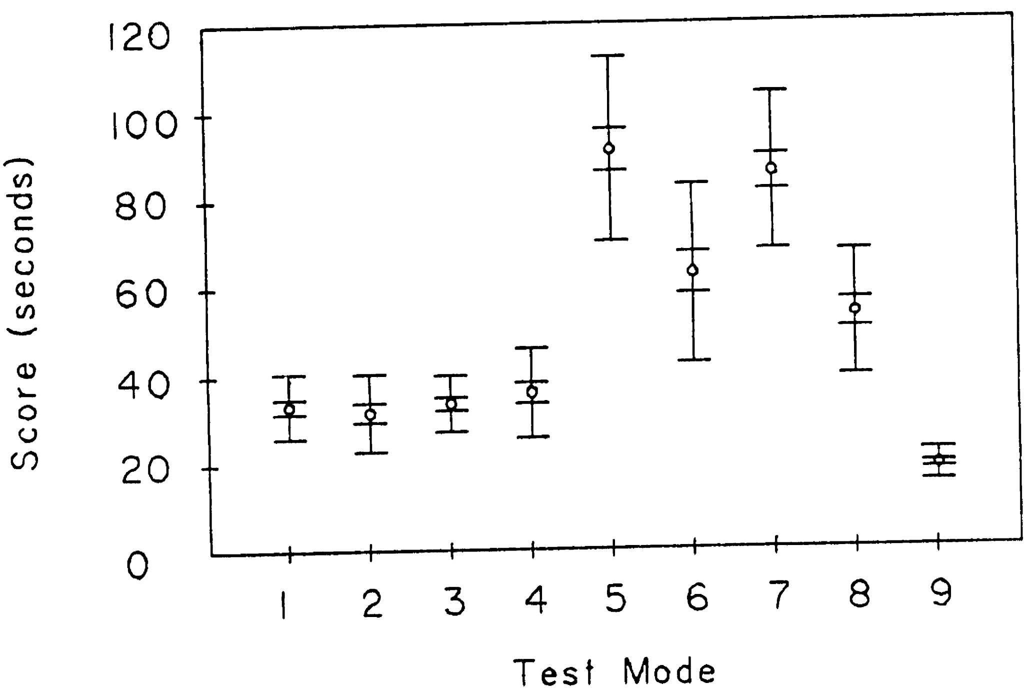

Figure 5-10
Overall BLOCKS Results

6. Conclusions and Recommendations

Results show rather consistently that the addition of undelayed simulated graphics to an intermittently updated (delayed) low bit rate video image can improve teleoperator performance. This result is clear in spite of the fact that the sample sizes of the tests performed were small and there was some apparent learning during the actual testing.

The gain in performance to be had by using such a model-based graphic predictor will vary depending on the type of task being performed, and with the display resolution and update rate.

While this study did not systematically study the effectiveness of graphics for different delays in the video, delays were in effect introduced by the buffered display. For example, the 33 frame per minute buffered mode was in a sense the equivalent of a 0.03 minute or 1.8 second delay (with intermittent frame changes). Thus it seems clear that graphics will aid performance in the case of transmission delay, even without intermittent video updating.

A systematic study with a variety of delays remains to be done. If a similar system could be configured in which predictive graphics could be superimposed on a delayed (1-3 seconds) but quickly updated (30 Hz) video picture, an excellent approximation to the aerospace teleoperation problem would be available.

Bibliography

1) Newman and Sproull, Principles of Interactive Computer Graphics, McGraw-Hill Book Company, 1979, Second Edition.

2) Bradley Deghuee, Operator-Adjustable Frame Rate, Resolution, and Grey Scale Tradeoff in Fixed-Bandwidth Remote Manipulator Control, M.S. Thesis, MIT, 1980

3) Vivek Ranadive, Video Resolution, Frame Rate, and Grey Scale Tradeoffs Under Limited Bandwidth for Undersea Teleoperation, M.S. Thesis, MIT, 1979

4) W.R. Ferrell, Remote Manipulation with Transmission Delay, IEEE Transactions on Human Factors in Electronics, HFE-6 No. 1, Sept. 1965

5) W.R. Ferrell, Delayed Force Feedback, Human Factors, Vol 8, No. 5, Oct. 1965

6) Calvin Winey, Computer Simulated Visual and Tactile Feedback as an Aid to Manipulator and Vehicle Control, M.S. Thesis, MIT, 1981

7) F.W. Mounts, "A Video Encoding System Using Conditional Picture-Element Replenishment", Bell System Technical Journal, Vol. 50, No. 1, 1971

8) B.G. Haskell and R.L. Schmidt, "A Low-Bit-Rate Interframe Coder for Video Telephone", Bell System Technical Journal, Vol. 54, No.8, 1975

9) V.C. Anderson, "Acoustic Communication is Better Than None", IEEE Spectrum, October, 1970

10) Don Fyler, Computer Graphic Representation of Remote Environment Using Position Tactile Sensors, M.S. Thesis, MIT, 1981

Appendix A

Special Display Techniques

When using a raster type display system for generation of video images overlayed with dynamic graphics, it is very useful to have a graphics system which has multiple, function-selectable bit planes. This means that for any particular type of write operation (e.g video image, graphics, text) a write enable mask can be applied to all bit planes available. In this way, graphics can be written onto one set of bit planes without interfering with video, and vice versa.

Another feature that is provided by the use of multiple bit and function selectable bit planes is that vector graphics can be displayed dynamically without flicker. Suppose there is already a graphic image on the screen drawn on bit plane 1. When generating an updated graphics frame, we can write the new image on plane number 2, and then erase plane number 1. By always keeping the image on at least one of the bit planes, the display does not flicker when the graphics are erased.

Generally, systems which provide independent bit plane capability also provide lookup table capability. This is useful for controlling the relative intensity of the graphics and video images. For example, the display system can be programmed to display the image at an intensity level proportional to the pixel value unless overlayed by graphics.

Additionally, look-up table capability enables the picture brightness and contrast to be programatically modified as well.

Appendix B

Graphics to Image Scaling

In the United States, the standard monitor aspect ratio is 4 to 3. This means that the horizontal length of the screen is 4/3 as long as the vertical length. One implication of this is that in order to display square pixels, the horizontal to vertical resolution ratio must also be 4/3. This is important because in order to be able to display shapes which are not distorted, extra calculations must be done for non-square pixels. If the pixels are not square, graphics must be scaled along one of the two axes (either compressed or elongated) to allow proper display.

In the program described, the graphics were scaled to the screen at program startup time if necessary. This is to accomodate any movement or disturbance of the camera's orientation. In order to do this, two small crosses are displayed at the same height and on opposite sides of the screen. Then, the operator moves the manipulator to cause a marked spot on it show up at the same location as the two crosses on the screen, in sequence. The computed world coordinate for the manipulator mark is recorded at each location.

In order to scale the graphics to the screen, 3 camera orientation angles and a scale factor must be determined. In order to solve for these unknowns, 4 constraints must be met. These four constraints can be the screen x, y coordinate values for the graphics generated for the two marked locations. Alternatively, as was done in the program, the constraints can be two x locations, one y, and the distance between the two locations. This choice was made in order to simplify the calculations.

For each world coordinate, its screen coordinate is given by:

$$S = P * WV * W$$

where S is the screen coordinate, P is the transform from viewpoint coordinates to screen coordinates, WV is the transform from world coordinates to viewpoint coordinates, and W is the coordinate in the world coordinate system. The scaling factor will be in the P transform, and the 3 camera angles will be in the WV transform.

To solve the constraints, we use an iterative technique. First, the distance between locations is first solved holding the camera orientation angles constant. Doing this helps computationally because it allows us to work with a 3X3 system of constraints instead of a 4X4 system. The partial derivatives of the screen coordinates with respect to the three camera angles are computed. The solution can be condensed to the form :

$$\begin{matrix} \frac{dXs}{dA1} & \frac{dYs}{dA1} \\ \frac{dXs}{dA2} & \frac{dYs}{dA2} \\ \frac{dXs}{dA3} & \frac{dYs}{dA3} \end{matrix}$$

For one X coordinate and two Y coordinates, we set up the Jacobian matrix:

$$J = \begin{matrix} \frac{dX1s}{dA1} & \frac{dY1s}{dA1} & \frac{dY2s}{dA1} \\ \frac{dX1s}{dA2} & \frac{dY1s}{dA2} & \frac{dY2s}{dA2} \\ \frac{dX1s}{dA3} & \frac{dY1s}{dA3} & \frac{dY2s}{dA3} \end{matrix}$$

Approximating the functions linearly about the point of evaluation, we then set up the approximate matrix equation.

$$DCs = J * DA$$

where DCs is the error in the screen coordinates of the marked locations, J is the Jacobian given above, and DA is the vector of the change in the three camera angles necessary to cause the DCs vector. We then invert the J matrix and multiply both sides by it, producing the matrix equation:

$$J^{-1} \quad * \quad DCs \quad = \quad DA$$

This gives us the DA vector which we then subtract from our current A vector to get a new A vector. This process is repeated until satisfactorily small error in screen coordinates result.

Appendix C

Program Flow Charts

Two programs were used in the graphics display system. The program running on the PDP-11 was responsible for calculating vector coordinates for display, and the LSI-11 had the responsibility for displaying the vectors and video image. Flowcharts for the two programs are presented here.

The PDP-11 program is depicted in 4 flow charts. Figure C-1 shows the overall flow of che program. Figure C-2 shows the flow in initialization. Figures C-3 and C-4 show the steps taken during the graphics to image scaling process. The LSI-11 program is depicted in Figure C-5.

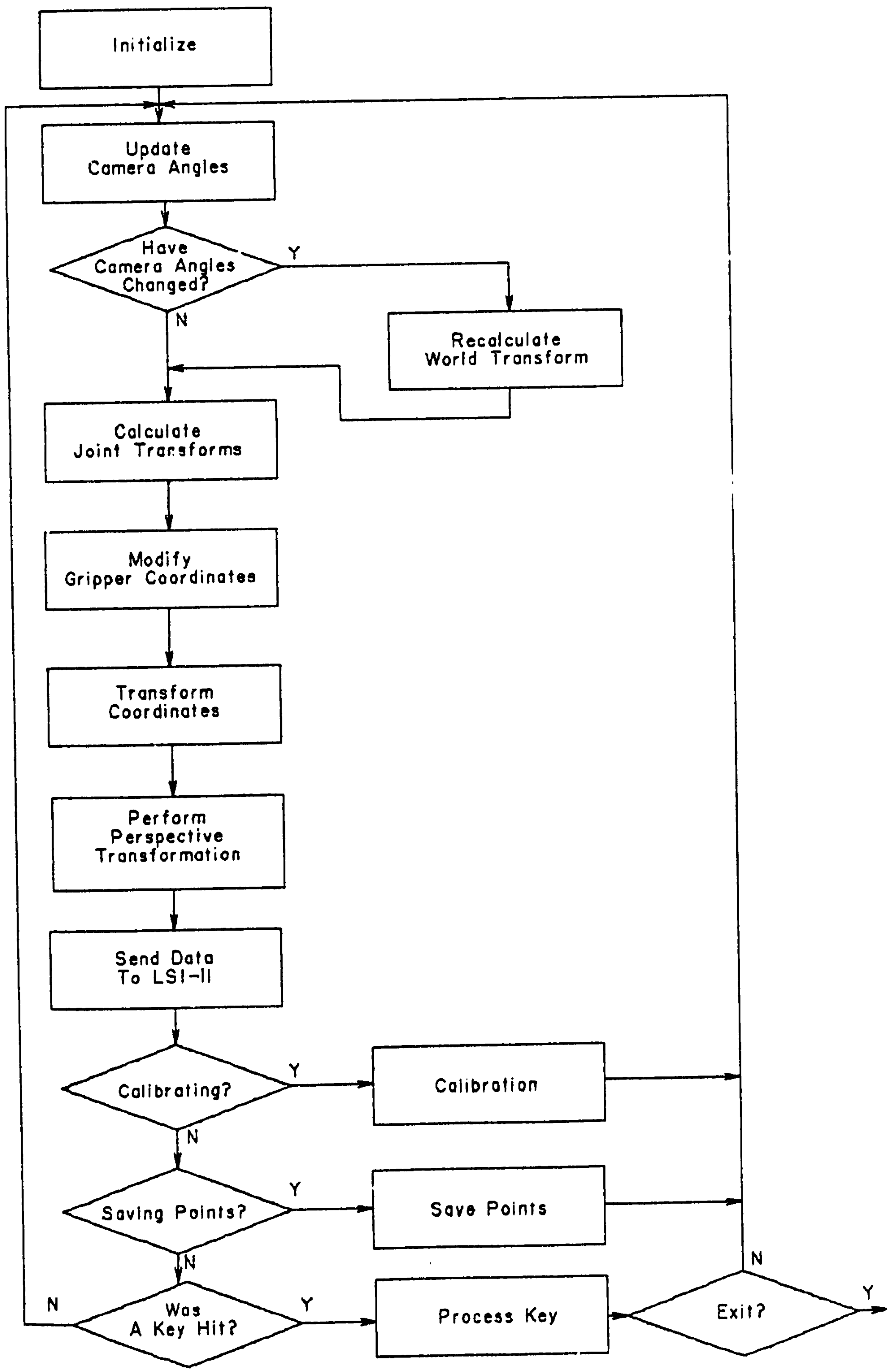

Figure C-1
PDP-11 Main Program Flow Chart

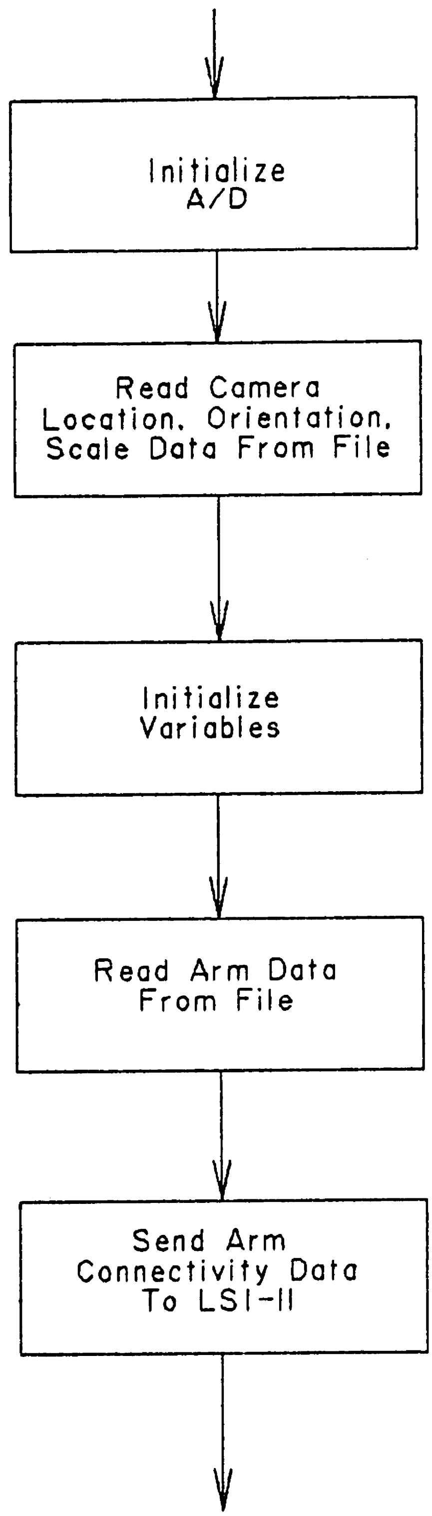

Figure C-2
PDP-11 Main Program Initialization

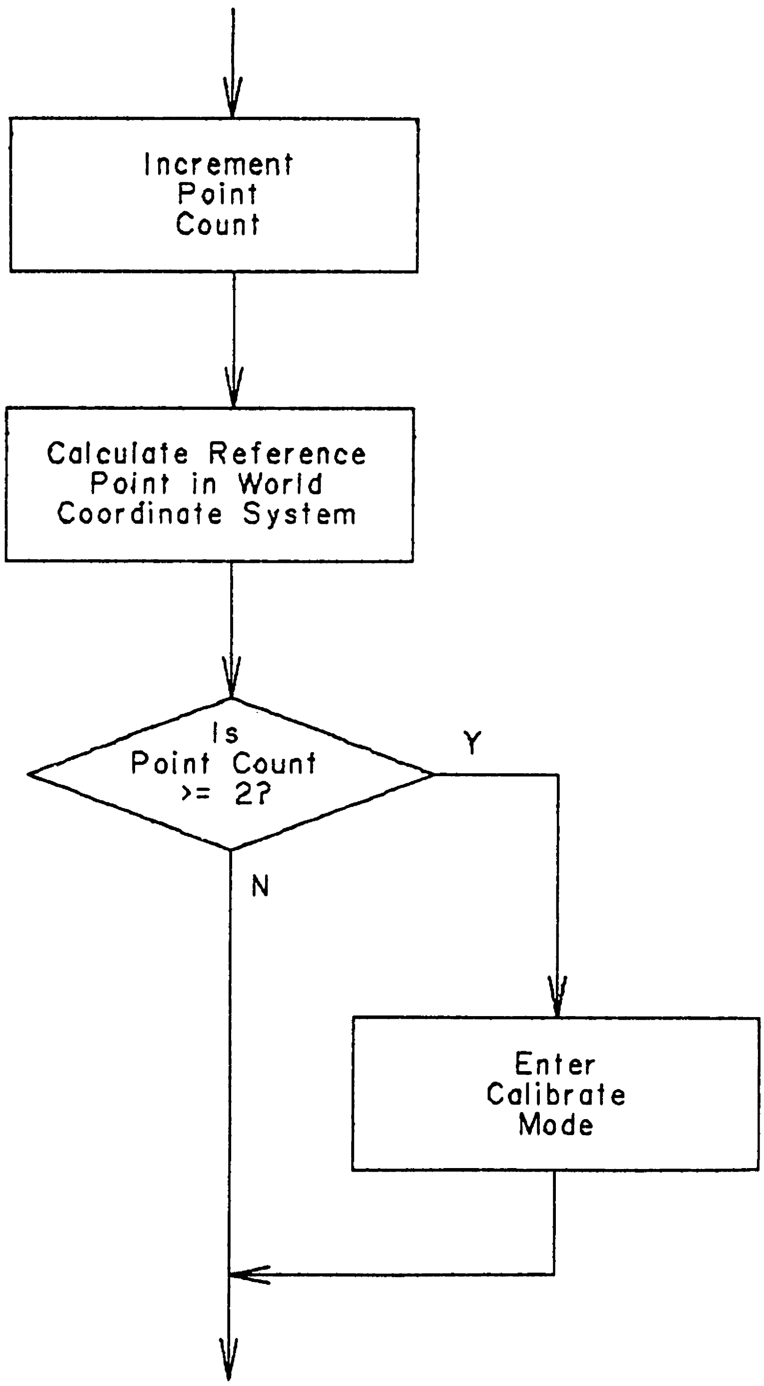

Figure C-3
PDP-11 Main Program Data Acquistion Flow Chart

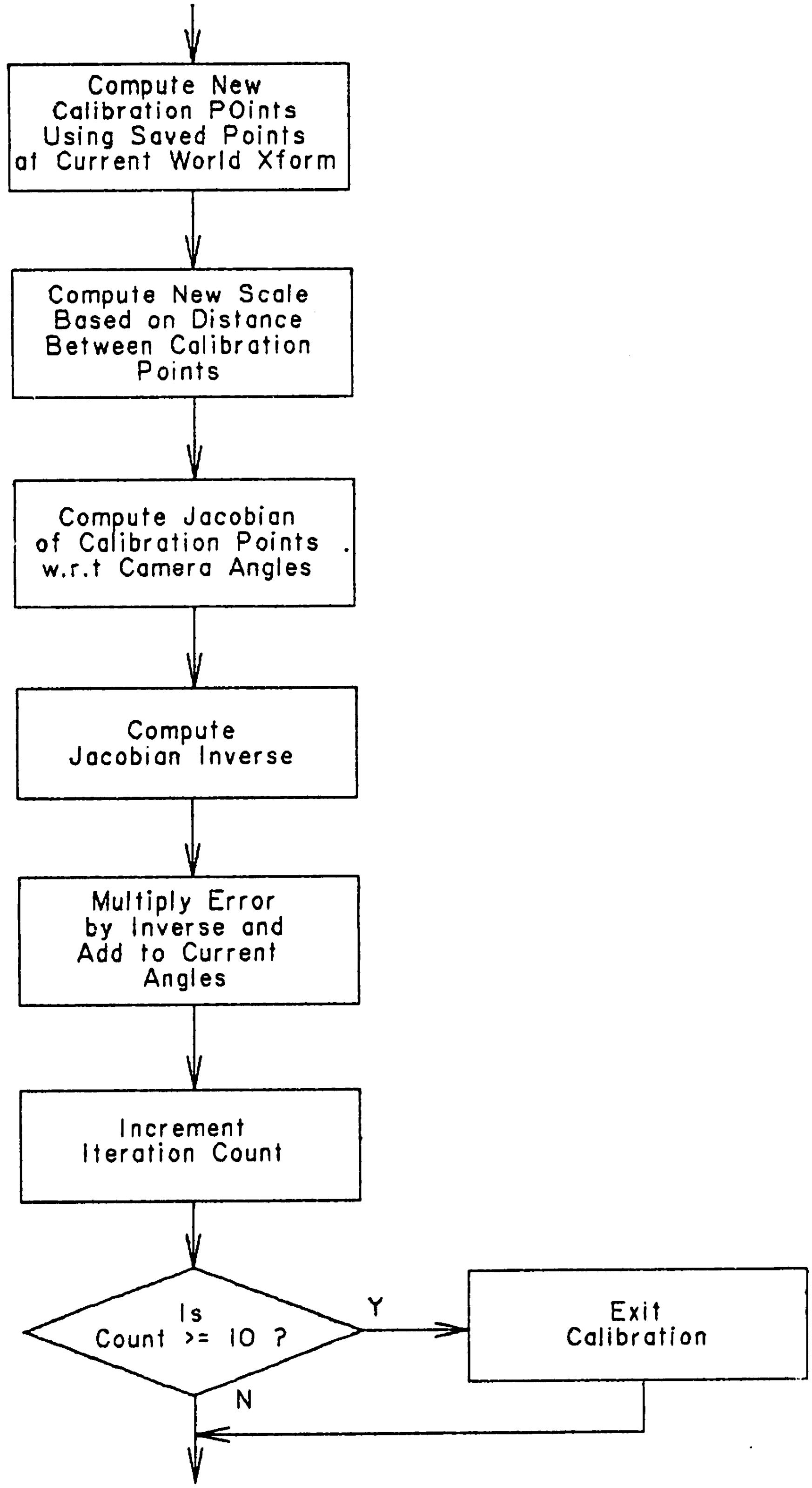

Figure C-4
PDP-11 Main Program Video to Image Scaling Flow Chart

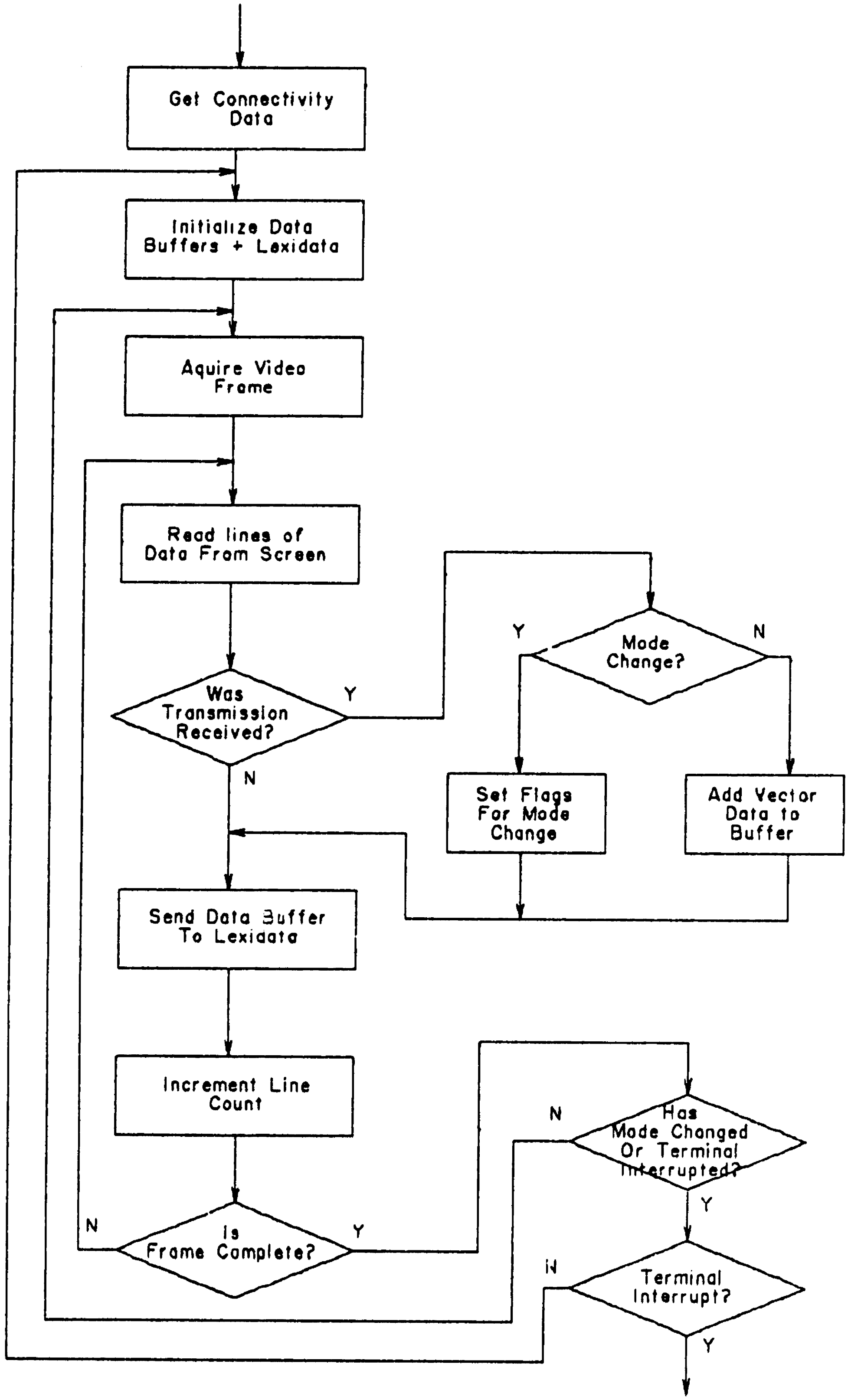

Figure C-5
LSI-11 Program Flow chart