High-speed interaction on a vibrating-mirror 3D display

Peter H. Mills, Henry Fuchs, Stephen M. Pizer

Department of Computer Science, University of North Carolina
Chapel Hill, N.C. 27514

Keywords: 3-D display, varifocal mirror, vibrating mirror, raster graphics, interaction.

Abstract

An implementation of a true 3-dimensional display-system using vibrating mirror is described. This implementation is characterized by using a standard raster graphics system, and supports user-interaction for independent manipulation of multiple objects that is near real-time. The high-speed performance of the object-manipulations, which include translation, rotation, scaling, intensity-windowing, spatial-clipping, intensity-highlighting, and blinking, is achieved through five means: exploiting the power of the user-programmable graphics processor, fully utilizing the raster-device characteristics, coordinating concurrent execution of the host and graphics computers, localizing the image changes effected by interaction, and by relying upon successive refinement of the initially coarse object-display during periods of reduced interaction. Recent enhancements include the capability to display more points (circa 120,000), the integration of the functions into one system allowing multiple simultaneous manipulations, and the support for independent object motion. The system is currently being used to display CT and NMR data for medical imaging and electron microscope tomographs for molecular modeling.

Introduction

True 3-dimensional display is concerned with the presentation of images of 3D objects so that they are perceived, ideally, as if actually possessing 3D form. Human perception of 3D structure is based on various cues for depth: among these are stereopsis (different views from left and right eyes), kinetic depth effect, head-motion parallax (perception of a different image from each viewing position), perspective, and obscuration of further parts by closer ones [Fuchs, Pizer, et al., 1979]. Current systems that achieve true 3D display can be minimally characterized as providing head-motion parallax (and thus stereopsis): these methods include holograms [Lesem & Hirsh, 1968; Huang, 1971; Gabor, 1972; Benten,1977; Perlmutter,1982], integral photography using arrays of lenses [Lippmann, 1908; de Montebello, 1977a], and vibrating or rotating mirror and lens devices [Withey, 1958; Muirhead, 1961; Traub, 1967; de Montebello, 1977b; Fajans, 1979]. Holography and integral photography are subject to long image production times which preclude interactive manipulation. Most vibrating or rotating mirror and lens devices have mechanical properties that prevent routine use: however, the varifocal mirror is a promising exception [Muirhead, 1961; Traub, 1967; Rawson, 1968,1969; Hobgood, 1969; Fuchs, Pizer, et al., 1979, 1982a, 1982b; Cohen, 1979; Baxter,1981; Pizer, Fuchs, et al., 1982c, 1983]. This paper will describe an implementation of a 3D varifocal mirror display which can be constructed as a low-cost add-on to a standard frame-buffer raster graphics system, yet which provides highly interactive manipulation features.

History

The concept of a varifocal mirror was first introduced by T. Muirhead [Muirhead, 1961], and later implemented in an autostereoscopic (i.e., true 3D) imaging system by A. Traub at MITRE [Traub, 1967]. It was further developed by E. Rawson at Bell Labs [Rawson, 1968,1969] and by S. Hobgood at UNC [Hobgood, 1969]. L. Sher of Bolt Beranek and Newman Inc. [Sher, 1979] developed a rigid fiberglass mirror later used by Genisco Computers for their Spacegraph Terminal [Stover, 1981; Stover; 1982]. At the University of Utah, a varifocal mirror was developed which, like the Genisco model, could display in either point-plotting mode (fully described later), or in raster mode [Baxter, 1978]. This system was later extended to allow mixed modes as well as enhanced object definition [Johnson and Anderson, 1982].

At the University of North Carolina, development of a varifocal mirror system was first performed in 1969 by Sandy Hobgood under the direction of F.P. Brooks [Hobgood, 1969], using a vector display unit. A second implementation used a non-interactive flash-tube projector [Fuchs, Pizer, et al., 1979]; this was later discarded in favor of an implementation similar to that described herein, supporting display of a single object with manipulation capability using a standard raster display system [Fuchs, Pizer, et al., 1982a, 1982b; Pizer, Fuchs, et al., 1982c]. Earlier in 1979, a detailed analysis of mirror properties was undertaken by J. Cohen [Cohen, 1979]. Since 1982, work at UNC has focused on speed and complexity of manipulation, as well as on increased point capacity for display. Applications have included display of CT, ultrasound and NMR data in medical imaging [Pizer, Fuchs, et al., 1983] and display of images derived from electron microscope tomography [Olins, Levy, et al., 1984].

Basic Principles of Varifocal Mirror Display

The oft-described principles of varifocal mirror operation are as follows.

1. Overview:

One may present slices of an object at different perceived depths quickly enough for the eye to fuse into a coherent 3D image. This variation in perceived depth is accomplished through reflecting the slice in a mirror whose depth is varied; however, mirror depth is varied by deforming the mirror curvature rather than by moving the entire mirror. Each mirror depth gives a different perceived 2D image depth, placing the slices throughout an image volume.

2. Varifocal mirror concept:

More specifically, a highly flexible reflective mylar surface can be deformed via air pressure to form a mirror whose curvature is roughly spherical at low amplitudes of excursion: this curvature, and thus focal length, may be varied in a controlled manner by changes in pressure [Muirhead, 1961]. A CRT screen is positioned so that the viewer sees the reflection of the CRT in the mirror. For a given degree of mirror-deformation, the 2D image displayed in the CRT is related to the perceived image in the following way: for each point in the CRT, the perceived point will have x and y coordinates as a function of CRT screen position; the perceived z position will be a function of the mirror curvature and thus focal length; the perceived intensity (or color) will be the same as that on the CRT. (See Figure 1.)

3. 3D Display:

The display of a 3D object is achieved by serial presentation of component z-slices on the CRT, synchronizing the CRT display with alteration of the mirror curvature so that for each z-slice, its perceived depth as a function of focal length corresponds to its actual depth within the object. By repeatedly presenting all slices of an object at a sufficiently high frequency (circa 15HZ or higher), the persistence of vision will perform the fusion required to see a coherent 3D image. The required cyclic mirror deformation is easily accomplished by placing a loudspeaker behind the mirror frame and driving it at (in our case) 30HZ: the object slices need then be displayed in an ordering which correctly relates depth with the induced variation in mirror curvature [Traub, 1967].

4. Varifocal Mirror Display Characteristics:

Such a vibrating mirror display possesses a number of characteristic advantages as well as liabilities. It provides the depth cues of stereopsis and head-motion parallax while failing to support naturally induced obscuration, since the 3D images are translucent and essentially luminous [Rawson, 1969; Fuchs, Pizer, et al., 1979]. Hidden surfaces explicitly removed through computation are consistent to only one observer position [Johnson & Anderson, 1982] and so are not usually utilized. Kinetic depth effect, although a weaker depth cue than head-motion parallax, can be provided at the expense of extra computation [Fuchs, Pizer, et al., 1979]; the manipulation capabilites later described can sometimes generate sufficiently detailed moving images to provide this effect. The spherical curvature of the mirror surface, while inducing display-volumes that are centimeters in depth for excursions on the order of millimeters, unfortunately also magnifies an image's xy-size as it is moved along the depth-axis away from the viewer, an effect termed "anomolous perspective" [Traub, 1967; Rawson, 1969]. Compensation for the latter effect can be made through software or through analog circuitry altering the x and y signals prior to input to the CRT device; the analog approach is used at the University of Utah [Baxter, 1984].

5. Correspondence of Slice-depth and Mirror-position:

Synchronization of the object-slice display with proper mirror position can pose problems: while the beginning of the mirror's motion from front (closest to viewer) to back (farthest from viewer) to front again may be synchronized through various means with some external 30HZ signal, the uniformity of the shape of the mirror motion from cycle to cycle is more difficult to maintain as one gets further away from the sync pulse within one cycle. Furthermore, within one cycle, the back-to-front stroke (termed "back" stroke) of the mirror may not be of the same shape or offset as the front-to-back stroke (termed "fore" stroke), forcing measures to be taken to ensure that object points of the same depth are displayed at the same position of the mirror regardless of whether this is during back or fore stroke. Lastly, if the mirror-driving waveform is sinusoidal, the focal length will be a nonlinear function of time, which may result in irregular distribution of the set of image planes along the depth axis. A filtered sawtooth for the driving waveform voltage can be used to enforce linear image sweeps as well as to increase fore-stroke duty time [Rawson, 1968]; however, to our knowledge this technique is not used due to the induced side-effect of audible high-frequency noise.

6. Generalized Methods:

It should be noted that a generalization of the above 3D display concepts is used for many volumetric display methods other than those using varifocal mirror: a series of 2D images are spread through a volume of space cyclically at a frequency higher than the eye can resolve, with the result that the human visual system creates the perception of a 3D spatial image [Traub, 1967; Rawson, 1969]. For example, the object slices comprising the 2D images may be rotational cross-sections for rotating mirror or lens devices [Fajans, 1979] as opposed to parallel cross-sections for vibrating mirror.

Implementation

Goals

The design of the implementation was oriented towards the following goals.

1. Ease of implementation:

No specially designed hardware, aside from the varifocal mirror assembly, is required. A standard raster display system [viz. Adage Ikonas RDS 3000] together with a point-plotting fast-phosphor CRT, mirror-driving waveform generator (using a microcomputer), host computer [viz. VAX 11/780], and fairly simple software, will suffice.

2. Interactive with multiple-object support:

The user has the capability to interactively define and display multiple objects, and to manipulate arbitrary groups of objects independently from the others being displayed. A wide range of manipulation functions is available, including those for specifying object-motion (scaling, translation, and rotation), those for modifying object-attributes (highlighting, blinking, fore-or-back stroke display), and those for semi-global operations (intensity windowing, spatial clipping, single vs double buffering).

To facilitate independent control of multiple objects, any subset of objects can be manipulated together, grouping them into a cluster termed a "transform class". The various control parameters of a transform class (such as rotation, translation, and scaling values, which are encoded in a "transform matrix") can be tied to any physical control device. A separate clustering, distinct from those for controlling object-motion, can be set up for intensity windowing and spatial clipping; however, implementation decisions limit the number of these "windowing classes" to two.

A menu-driven host program together with numerous graphical-input devices provides a flexible and user-friendly interface for these interactions. Support is also provided to record and playback any sequence of commands and input-device values resulting from user-action; these recordings reside in script-files on disk.

3. High-speed interaction:

Interaction, to be most effective, should have near-immediate response. Response time can be minimized by fully utilizing the available resources, by applying the concept of "successive refinement", and by localizing the effects of interaction on the image. The first entails use of the user-programmable graphics processor, exploitation of graphics-device hardware characteristics [e.g., colormap], and concurrent execution of programs on the host VAX, on the graphics processor, and on the multiplier-accumulator used by the graphics engine for point-transformations.

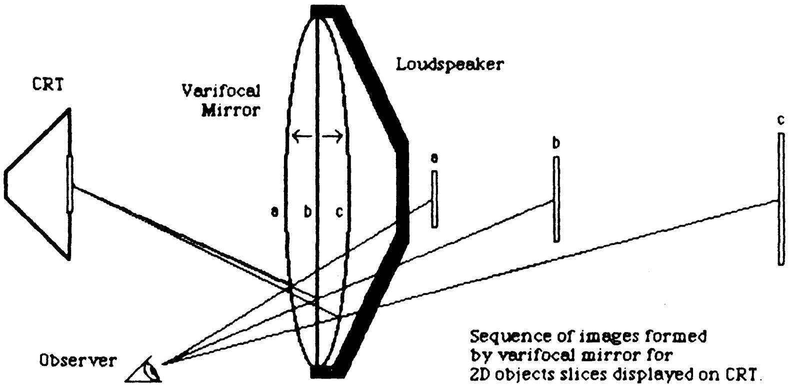

CRT
Varifocal Mirror
Loudspeaker
a
b
c
a
b
c
Observer
Sequence of images formed by varifocal mirror for 2D objects slices displayed on CRT.

Figure 1. Varifocal Mirror Principle

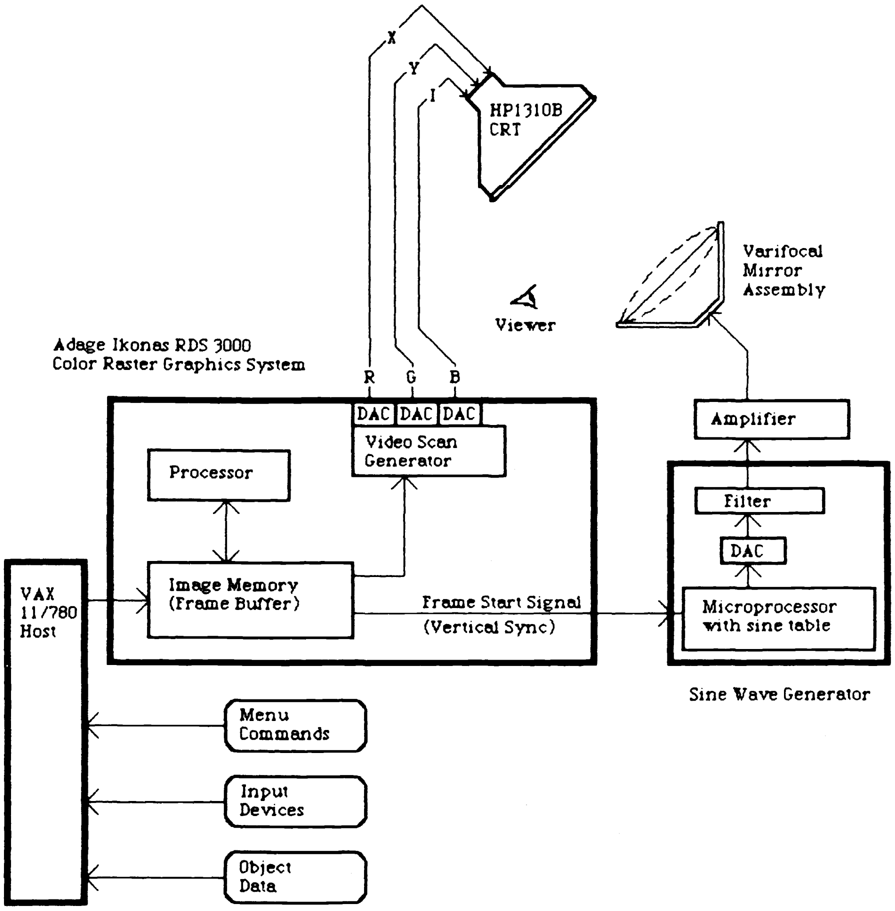


Figure 2. UNC Varifocal Mirror Display System

One way to ensure real-time feedback is to initially present a coarse image upon interaction, and to refine the image during periods of non-interaction; this method is termed "successive refinement". The coarseness of the initial presentation as well as the speed of refinement is dependent upon the power of the implementation.

Localization of change is applicable to moving a set of objects relative to a stationary background of other objects. Since hidden-surface removal is not performed, the only part of the image needing change is that part which is being moved. Ideally, therefore, the initial presentation of the 3D image upon manipulating a subset of all objects will consist of the fully detailed static background together with a coarse rendition of the dynamic objects; only that part of the image containing the dynamic objects need be successively refined. This localization of the effects of interaction to that part of the 3D object holding the manipulated objects we call "relative dynamism".

4. Capacity to Display Large Numbers of Points:

Adequate presentation of object detail relies upon the ability to display a large number of points. Maximizing the number of points displayed can be achieved by using both mirror fore-stroke and back-stroke. Unfortunately, tradeoffs between capacity for display and speed of interactive update exist: these are described in a later section entitled "Frame-buffer Use and Configuration".

Underlying Display Mechanism

The basic scheme for 3D display in point-plotting mode is described below. (See Figure 2.)

1. Raster-scanout to CRT (RGB-->XYI):

The red, green, and blue output channels from the raster-device are connected to the x,y, and intensity input channels of the fast-phosphor point-plotting CRT, whose display is reflected in the varifocal mirror. Each mirror vibration (front-to-back-to-front excursion) is synchronized with one refresh cycle of the raster-graphics-device by means of a sinewave generator (controlled by a microprocessor) which receives refresh-sync signals from the raster-device. When the pixel contents of the frame-buffer are scanned out in raster-order for one refresh cycle, each correspondingly displayed point has a unique depth which is implicit in the frame-buffer address: this is because a point's perceived depth is a function of mirror position, which corresponds to time of display within refresh-cycle, which in turn corresponds to the point's frame-buffer address. By storing a point's x,y and intensity values in the RGB slots of a pixel in frame-buffer memory, and by choosing the pixel address as a function of z-depth, each scanout of the frame-buffer in raster-order will properly spread the output pixel values throughout an image volume. One should note that actually the image volume is swept twice during one refresh cycle; for our implementation using repeat-field mode, the upper-half of the frame-buffer is displayed during the mirror fore-stroke, while the lower-half of the frame-buffer is displayed during mirror back-stroke. For interlaced mode, the odd scan-lines of the frame-buffer would be displayed during mirror fore-stroke, while the even scan-lines would be displayed during back-stroke.

2. "Bins" as regions of approximately equal depth:

Each point is displayed at a unique depth; however, the depth values are quantized, and many points within an image may have the same depth. The solution to this problem is to segment the frame-buffer into contiguous regions into which are stored points whose depth approximates that of the average perceived-depth of the region's output pixels. Since the eye cannot perceive small variations in depth, the approximation yields satisfactory results. The regions of approximately equal depth, termed "bins", are chosen to be a number of contigous scan-lines. By filling bins in increasing frame-buffer address, from left to right and top to bottom, occupied and free bin areas can be tracked by maintaining "starting-address" and "next-free-address" pointers for each bin. Furthermore, non-linearity in the mirror-driving waveform can easily be compensated by variation in bin starting location. For our system, each bin is typically 4 scan lines in size, creating 64 distinct depth levels within the upper-half of a 512x512 display. (See Figure 3.)

3. Frame-buffer Use and Configuration:

The size of the frame-buffer, and its allocation for various functions, greatly influences the performance and quality of the varifocal mirror display. The frame-buffer (i.e., pixel memory of the raster graphics system) has as its primary use the aforementioned storage of object points, one per pixel, in bins for scanout to the CRT. Additionally, since it is the only large memory available to the microprocessor, it serves to hold the raw object data (x,y,z,intensity) which is, for each manipulation of the object, transformed and then inserted into the bins. The frame-buffer area available for bin storage can be split vertically in half to enable double-buffering: one half of the bin storage area is used for display on the CRT while the other half is being updated. Clearly storage of raw object-data and double-buffering require a means of limiting framebuffer-scanout to selected areas.

At UNC, the frame-buffer consists of 5 screenfulls of 512 x 512 pixels, each 24 bits deep. This allows double-buffering with full screen buffers, enabling fore and back stroke display; enough room is left over for storage of about 300,000 raw data points. Unfortunately, only about 120,000 points may be displayed at one time from the possible screenful of 256,000 points in a buffer; CRT beam inertia requires the insertion, between any two nonzero points which are adjacent within a bin but which are distant in the xy-plane, of a point with zero intensity and coordinates of the second point. Furthermore, for other systems with smaller memories, the raw data storage may impinge upon areas otherwise used for bins: leaving the raw data on the host for processing sacrifices update time for capacity of display. Double-buffering, for buffers less than one screen in size, may be abandoned to similarly sacrifice update time for number of displayable points.

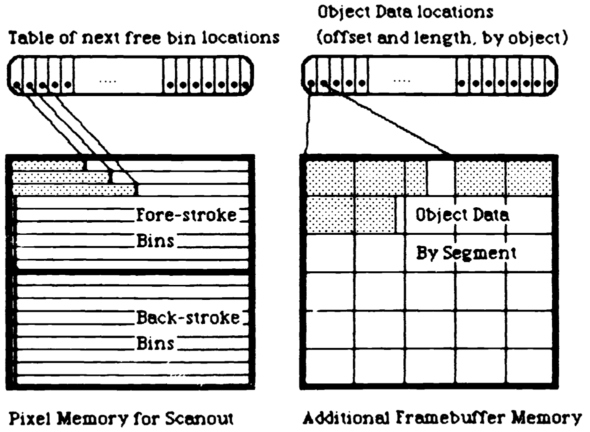


Figure 3. Framebuffer Organization

4. Host and graphics processor functions and coordination:

System functions are achieved by coordinated operation of the host and graphics processors. The host interfaces with the graphics-input devices [e.g., joysticks, sliders] and handles menu commands issued from the terminal. In response to user-actions signaling definition or manipulation of displayed objects, the host interrupts the raster-engine slave and downloads required data to it via asynchronous slave-acknowledged communication. The graphics processor is responsible for, upon interruption, taking the downloaded data, transforming the raw data points for all objects actively displayable, discarding points outside the 3D viewport ("clipping"), properly filling the frame-buffer bins with transformed points, and switching the pixel scanout to come from the newly filled bins.

As mentioned before, in order to support independent manipulation of multiple objects, transform-matrices and class-object links are required. Thus, the downloaded data may include object-data-points, object-attributes, transform-matrices, or the links specifying which transform to use for each object. The object-data is downloaded only once and is stored in a reserved area of frame-buffer memory; other data is stored in microprocessor RAM.

The actions of the graphics processor upon interruption are more specifically as follows. A double-buffering scheme is used to handle simultaneous update of and display from the frame-buffer; the first action is to clear required portions of the frame-buffer area targeted for update. To ensure that initially coarse renderings include representative points from all displayable objects, small segments of points from each object are selected for transformation and then insertion into the bins in the frame-buffer. Thus one set of segments, one from each object, is processed at a time; this is repeated until refinement is complete or until next interruption occurs. To support immediate feedback, scanout to the CRT is switched to the "bin buffer" being updated as soon as one segment-set is processed. To ensure representative coarse sampling from all areas within an object, the placement of points among object segments is randomized prior to usage within the system. Transformation is performed by an MA1024 programmable multiplier-accumulator which runs in parallel with the graphics processor; transformation and bin-insertion are interleaved in units of object-segments. (See Figure 4.)

It should be noted that, for the current version of this point-oriented display system, no provision is made for dynamically scan-converting line specifications. Preprocessing programs are available which convert line and polygon drawing commands to the required object point format.

Hardware Characteristics Supporting High Performance:

Various hardware characteristics within the raster-graphics system support the above features:

Crossbar -- Upon pixel scanout, the user may specify through an array an arbitrary rearrangement of pixel bits. The crossbar, as well as the colormap, is used for spatial clipping, intensity windowing, and blinking.

Colormap -- For each RGB channel, the user may specify through a table a unique output pixel value for each input value. Input and output values range from 0 to 1023.

Window and viewport -- These specify the starting frame-buffer address and size for scanout. Double-buffering techniques, as well as storage of object-data in the frame-buffer, require the use of these functions.

User-programmable Graphics Processor -- This microprocessor performs most of the work. It has been most useful for us to have a cross-compiler from a subset of C to the microcode: the cross-compiler, "GIA2", was developed at UNC by Gary Bishop [Bishop, 1982].

Programmable multiplier-accumulator -- The MA1024 is essential for concurrent high-speed transformation of data.

Features: Principles of Operation

The manipulation capabilites listed as goals rely upon the raster hardware characteristics and underlying display mechanism for their implementation, as described below. It should be noted that windowing, clipping and blinking are real-time operations, as is also dynamic motion of objects under 1000 points.

1. Independent Object Motion:

Each object is individually transformed using the MA1024 in parallel with bin-stuffing. Inter-object sampling of segments as well as intra-object randomization of points among segments ensures that coarse images are representative.

2. Single or Double Buffering:

Single or double buffering mode may be interactively selected. Single buffering mode sacrifices smoothness of image update for an increased number of points in the final refined image: this mode is not justified if more than one screenfull of frame-buffer memory is available. In line with the concept of succesive refinement, double-buffering is switched to single-buffering during periods of non-interaction, and then automatically reinvoked when interaction occurs.

3. Fore-or-Back Stroke:

Each object may be specified to have its points displayed during mirror fore-stroke, during back-stroke, or during both strokes. Both strokes may be used to maximize the number of displayable points by overflowing into back-stroke if fore-stroke bins fill up. Unfortunately, fore-back stroke registration problems slightly degrade image quality if back stroke is used for display.

4. Relative Dynamism:

Dynamic motion of objects against a static background is implementable by segmenting each bin into two or more fixed-area "update-levels". Initially, static objects are transformed and stuffed into bins first, followed by the processing of dynamic objects. Upon later interruption, since the static object-points are invariant, they need not be re-transformed; only the dynamic objects need to be reprocessed and inserted into the bin areas reserved for their update-level. Double-buffering considerations sometimes require copying of static object-pixels. Invocation of relative dynamism is transparent to the user: the system recognizes which objects are being moved. (See Figure 5.)

5. Intensity windowing:

Intensity windowing is easily accomplished by changing the intensity-map (for the blue channel) in the colortable; this effects global windowing. To achieve independently manipulatable windowing maps, we need only section the colortable into two or more parts and ensure that the recorded pixel intensities are looked-up only in that section of the colortable which represents the map associated with the object owning the pixel. By assigning one (or more) pixel bits for the "map id" and routing these bits into the most-significant-bits for colortable lookup, the desired separate intensity windowing is achieved at the

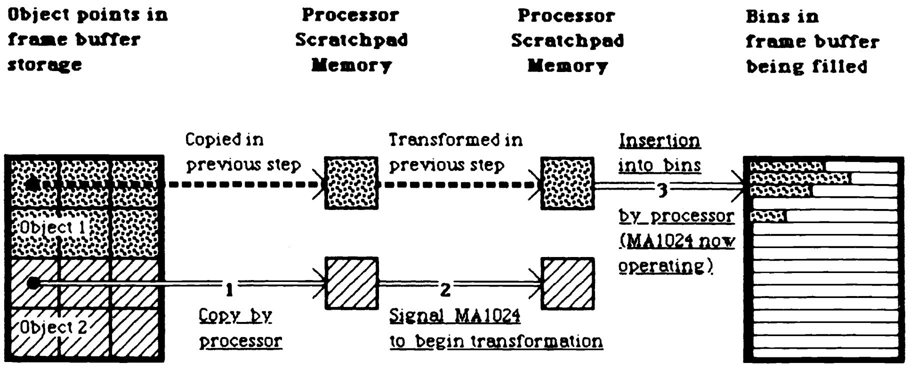
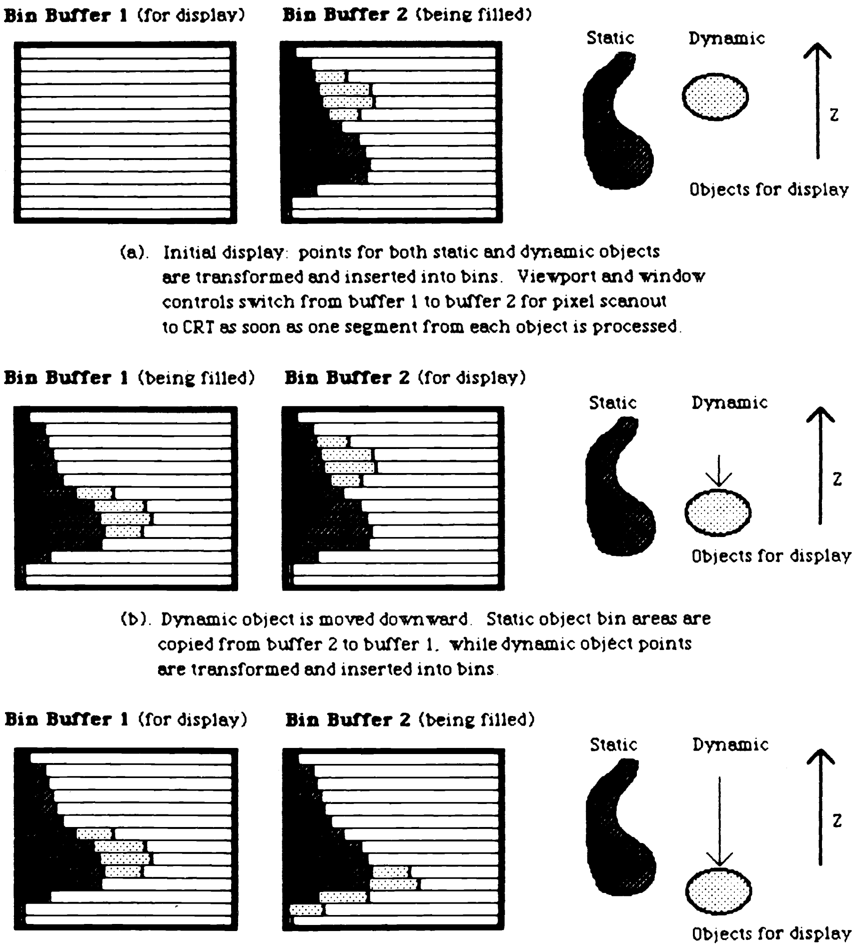

Object points in frame buffer storage
Processor Scratchpad Memory
Processor Scratchpad Memory
Bins in frame buffer being filled
Copied in previous step
Transformed in previous step
Insertion into bins
3
by processor (MA1024 now operating)
Object 1
Object 2
1
Copy by processor
2
Signal MA1024 to begin transformation

Figure 4. Data Flow from Object to Image Space

Each segment of 256 object points is transformed by the MA1024 multiplier-accumulator unit and then inserted into the framebuffer bins being filled for subsequent display. Clipping to the 3D viewport occurs during the bin-insertion step. Transformation and bin-stuffing proceed concurrently: the graphics processor cyclically copies one segment to RAM, signals the MA1024 to begin transformation of the copied segment, and then inserts the previously copied, and by now transformed, segment into bins.


Bin Buffer 1 (for display)
Bin Buffer 2 (being filled)
Static
Dynamic
Z
Objects for display
Bin Buffer 1 (being filled)
Bin Buffer 2 (for display)
Static
Dynamic
Z
Objects for display
Bin Buffer 1 (for display)
Bin Buffer 2 (being filled)
Static
Dynamic
Z
Objects for display

(a). Initial display: points for both static and dynamic objects are transformed and inserted into bins. Viewport and window controls switch from buffer 1 to buffer 2 for pixel scanout to CRT as soon as one segment from each object is processed.

(b). Dynamic object is moved downward. Static object bin areas are copied from buffer 2 to buffer 1, while dynamic object points are transformed and inserted into bins.

(c). Dynamic object is moved downward. For the dynamic object, old bin areas are erased in buffer 2 and points again are transformed and inserted into bins. Static object bin areas are not altered.

Figure 5. Relative Dynamism and Double Buffering

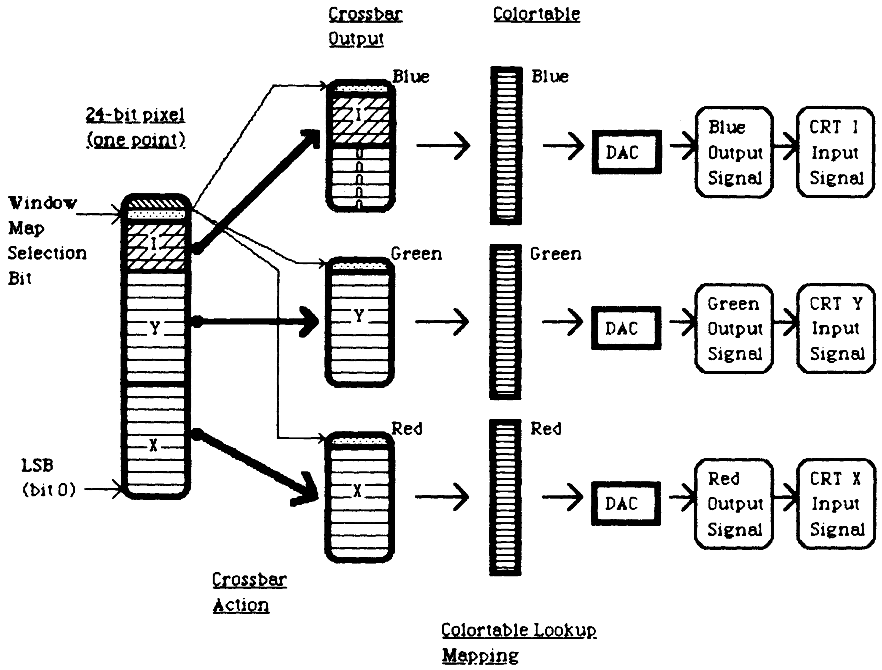


Figure 6. Crossbar and Colortable Actions

Intensity windowing and xy-spatial clipping are implemented by changing the blue, red, and green colortables used in lookup. The window mapping bit is routed into the MSB of each channel by using the crossbar, and determines which half of the the colortable to use for mappings. Blinking is achieved by alternatingly selecting between the two window mapping bits for crossbar routing.

expense of a reduced number of distinct intensity values. The UNC system halves the colortable to give two windowing maps, requiring reservation of one "map id" bit. (See Figure 6.)

6. Spatial clipping:

X and Y spatial clipping is performed in a manner similar to that for intensity windowing; the red and green channels of the colortable (corresponding to x and y values) may be altered to have 0 output for selected input ranges. This projects points outside the clipped region onto the x=0 and y=0 planes, respectively. Again, the colortable may be halved to give two separate clipping maps; however, to accomodate pixel size constraints, a single value (0 or 1) specifies both windowing and clipping map selection.

Z-clipping is performed by altering the Ikonas window and viewport control values, limiting or expanding the size of the framebuffer-area to be scanned out. (See Figure 7.)

7. Blinking:

Blinking, in simplest form, should be an alternation between dark and light, selectable by individual object. Since support for multiple windowing maps already exists, this effect is most easily accomplished by allowing the user to set one of two intensity maps to black, and then alternating between the two intensity maps. This idea can be generalized to alternating between spatial-clipping maps: cycling between clipped and non-clipped areas can prove useful. Yet it turns out that any real-time implementation of blinking requires the reservation of a bit or two from an already well-used pixel. Forcing one group of bits to serve as "map id" and "blinking on/off" specification, which is then applied to both intensity windowing and spatial clipping actions, provides all functions while costing a minimal number of pixel bits. (See Figure 6.)

This is realizable by encoding 01 in two reserved bit positions in each pixel owned by a blinking object, encoding 00 or 11 in the reserved positions for nonblinking objects with "map ids" of 0 and 1 respectively, and then using the crossbar to alternatively choose the first or second bit for routing to the most-significant-bit of all colortable lookups.

8. Highlighting:

Highlighting is achieved by increasing the intensity component of an object's pixels; for extreme highlighting, pixel replication may be necessary. Dimming may similarly be achieved by decreasing the intensity values or by sampling the object's points.

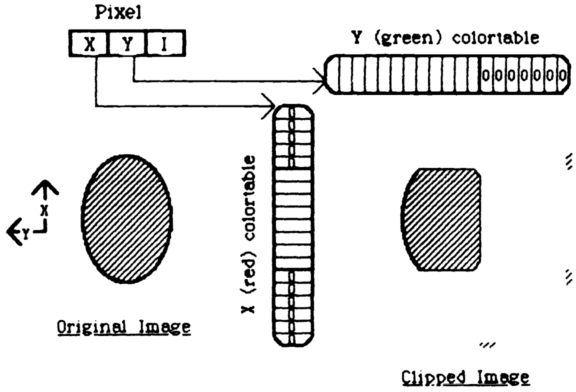


Figure 7.a. Example of XY-Spatial Clipping

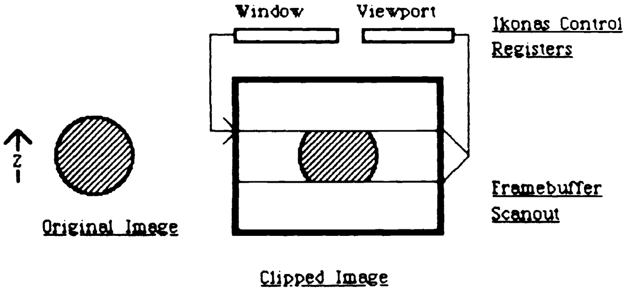


Figure 7.b. Example of Z-Spatial Clipping
The Ikonas window register controls the location in the framebuffer for start of scanout, while the viewport register defines the area for non-black scanout.

Limitations

Some practical as well as inherent limitations in our implementation present themselves. The number of points that can be displayed is a function of frame-buffer size, resolution and size of raster-display, and beam-inertia of the point-plotting CRT; we need to be able to display more points. While an increased number of points enhances visibility of complex objects, it also results in "clouding" of the image; although the ability to segment an image into individually displayable objects helps combat this, a need arises for selective area highlighting that is interactive. Displaying objects with different colors might also ameliorate confusion caused by obscuration. Color display might be achieved by placing electo-optic fast-shutter filters for color selection in front of the CRT [Lippman, 1984]. Enhanced functions that are interactive need to be designed, such as dynamic object creation using procedural modeling. Lastly, the speed of interaction needs to be further increased; a faster graphics processor is needed.

Conclusion

The described implementation of a varifocal mirror display-system supports near real-time interaction with multiple-object manipulation. It relies on an expensive but standard raster graphics frame-buffer system: no specialized processor hardware is required. A tradeoff exists between the varifocal-mirror display capability versus the power and complexity, and thus cost, of the raster graphics device, the host processor, and the supporting software. This tradeoff creates a continuum of varifocal-mirror display capabilities, from strictly static display of single object to those described here.

Acknowledgement

The work reported here was supported by NIH grant NS/HL 16759-01.

References

B. Baxter (1978, 1984), Personal communications, University of Utah, Salt Lake City, Utah.

B. Baxter (1981), "Three-dimensional Display Viewing Device for Examining Internal Structure", *Proceedings of the SPIE,* Vol.283, pp.111-115.

S.A. Benton (1977), ed., "Three-Dimensional Imaging", *Proceedings of the SPIE,* Vol.120.

G. Bishop (1982), *Gary's Ikonas Assembler, Version 2; Differences between GIA2 and C,* Technical Report Number TR82-010, Department of Computer Science, University of North Carolina.

J.S. Cohen (1979), *Three-Dimensional Display of Parallel Sliced Images,* Master's Thesis, University of North Carolina, Chapel Hill.

R.L. de Montebello (1977a), "Wide-Angle Integral Photography - the Integral System", *Proceedings of the SPIE,* Vol.120, pp.73-79.

R.L. de Montebello (1977b), "The Synthalyzer for Three-Dimensional Synthesis and Analysis by Optical Dissection", *Proceedings of the SPIE,* Vol.120, pp.184-189.

J. Fajans (1979), "Three-Dimensional Display", *Proceedings of the SPIE,* Vol.199, pp.23-28.

H. Fuchs, S.M. Pizer, J.S. Cohen, F.P. Brooks, Jr. (1979), "A Three-Dimensional Display for Medical Images from Slices" *Information Processing in Medical Imaging, INSERM,* Vol.88, pp.581-602.

H. Fuchs, S.M. Pizer, L.C. Tsai, S.H. Bloomberg, E.R. Heinz (1982a), "Adding a True 3-D Display to a Raster Graphics System", *IEEE Computer Graphics and Applications,* Vol.2, pp.73-78.

H. Fuchs, S.M. Pizer, E.R. Heinz, S.H. Bloomberg, L.C. Tsai, D.C. Strickland (1982b), "Design of and Image Editing with a Space-Filling Three-Dimensional Display Based on a Standard Raster Graphics System", *Proceedings of the SPIE,* Vol.367, pp.117-127.

D. Gabor (1972), "Holography, 1948-1971", *Science,* Vol.177, No.4046, pp.299-313, July 28, 1972.

W.S. Hobgood (1969), *A Three-Dimensional Computer Graphics Display Using a Varifocal Mirror,* Master's Thesis, University of North Carolina, Chapel Hill.

T.S. Huang (1971), "Digital Holography", *Proceedings of the IEEE,* Vol.59, No.9, pp.1335-1346.

S.A. Johnson, R. Anderson, "Clinical Verifocal Mirror Display System at the University of Utah", *Proceedings of the SPIE,* Vol. 367, pp.145-148.

L.B. Lesem, P.M. Hirsch (1968), "Computer Synthesis of Holograms for 3-D Display", *Communications of the ACM,* Vol.11, No.10, pp.661-674.

A.B. Lippman (1984), Personal communications, M.I.T., Cambridge, Mass.

G. Lippmann (1908), *Comptes-Rendus,* Vol.146, Gauthier-Villars, Paris, pp.446-451.

J.C. Muirhead (1961), "Variable Focal Length Mirrors", *Review of Scientific Instrumentation,* Vol.32, pp.210-211.

A.L. Olins, D.E. Olins, H.A. Levy, R.C. Durfee, S.M. Margle, E.P. Tinnel, B.E. Hingerty, S.D. Dover, H. Fuchs (1984), "Modeling Balbiani Ring Gene Transcription with Electron Microscope Tomography", submitted to *European Journal of Cell Biology.*

R.J. Perlmutter (1982), "Digital Holographic Display of Medical CT Images", *Proceedings of the SPIE,* Vol.367, pp.109-114.

S.M. Pizer, H. Fuchs, E.R. Heinz, S.H. Bloomberg, L.C. Tsai (1982c), "Varifocal Mirror Display of Organ Surfaces from CT Scans", *Proceedings of the World Federation of Nuclear Medicine and Biology,* Paris, France, August 1982.

S.M. Pizer, H. Fuchs, E.R. Heinz, E.V. Staab, E.L. Chaney, J.G. Rosenman, J.D. Austin, S.H. Bloomberg, E.T. MacHardy, P.H. Mills, D.C. Strickland (1983), "Interactive 3d Display of Medical Images by Varifocal Mirror", *Proceedings of the VIIIth International Conference on Information Processing in Medical Imaging,* Brussels, Belgium; Sept.,1983.

E.G. Rawson (1968), "3-D Computer-Generated Movies Using a Varifocal Mirror", *Applied Optics,* Vol.7, No.8, pp.1505-1511.

E.G. Rawson (1969), "Vibrating Varifocal Mirrors for 3-D Imaging", *IEEE Spectrum,* Vol.6, pp.37-43.

L.D. Sher (1979), "3-D Display", 1979 Conference on Computer Graphics and CAD/CAM, MIT, Cambridge, Mass., April, 1979.

L.D. Sher, U.S. Patent No. 4,130,832, for vibrating plate varifocal mirror.

H.S. Stover (1981), "Terminal Puts Three-Dimensional Graphics on Solid Ground", *Electronics,* July 27, 1981.

H.S. Stover (1982), "True Three-Dimensional Display of Computer Data", *Proceedings of the SPIE,* Vol.367, pp.141-144.

A.C. Traub (1967), "Stereoscopic Display Using Rapid Varifocal Mirror Oscillations", *Applied Optics,* Vol.6, No.6, pp.1085-1087.

A.C. Traub (Feb. 3, 1970), "Three Dimensional Display", U.S. Patent No. 3,493,290, The Mitre Corporation Assignee, Bedford, Mass.

E.L. Withey (1958), "Cathode-Ray Tube Adds Third Dimension", *Electronics,* Vol.31, No.21, pp.81-83.