DATAMATION®

"SUPERPAINT"...THE DIGITAL ANIMATOR

by Richard G. Shoup

REPRINTED BY
NETWORK ANALYSIS CORPORATION

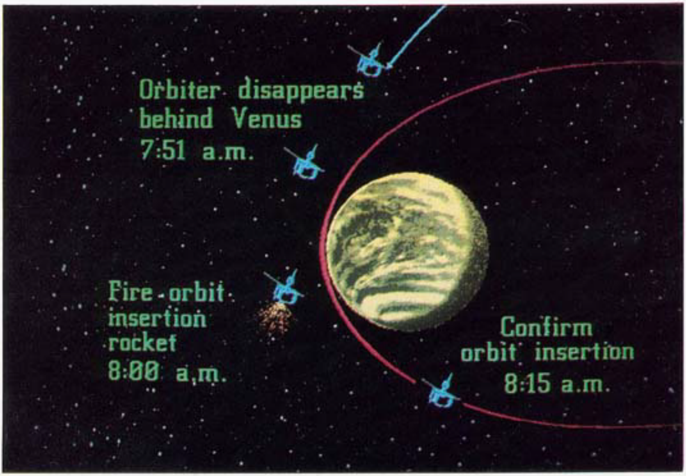
Developed as a computer graphics research tool, this experimental videographics system got nationwide tv exposure with its live, and lively, animation of NASA's Venus mission.

"SUPERPAINT" ... THE DIGITAL ANIMATOR

by Richard G. Shoup

An experimental digital video system which can be used for interactive creation and manipulation of simple, cartoon-like graphics and animated imagery was designed and built at Xerox in 1973 as an experiment in computer imaging and digital picture composition. Since then, the videographics system (known informally as "SuperPaint") has been further developed as a computer graphics research tool and used for a variety of experiments in television graphics and imaging.

Most recently (December 1978), the system was used extensively during the NASA Pioneer Venus mission for visualization of spacecraft maneuvers during



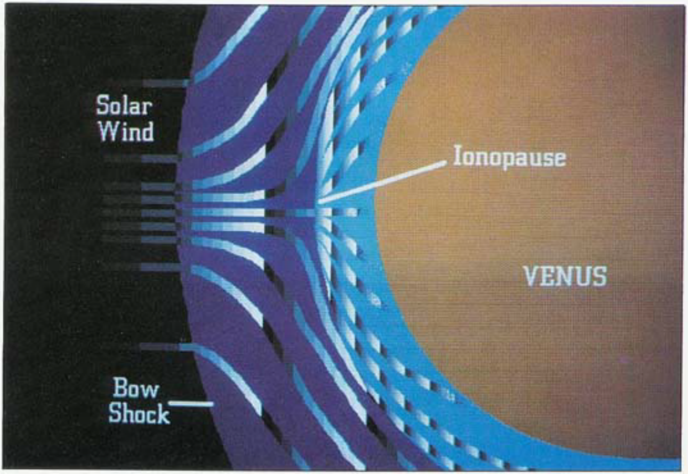
the encounter with Venus, for showing activities of the scientific experiments on board and for illustration of early results obtained. A live video feed from the system was provided during the mission for closed-circuit viewing by the press at NASA Ames Research Center. Graphics and animation created on the system were also used in numerous network and local television news broadcasts.

The system consists of a digital *image memory* (frame buffer) which holds 480x640 pixels or picture elements (8 bits per pixel), a data tablet and pen, a minicomputer and several digital disk drives for picture storage. Fig. 1 shows an overall block diagram. When viewed at this level, the system is quite similar to more recent frame buffer drawing systems. However, a novel image memory architecture and user-oriented software provide considerably more graphical power and flexibility than the typical frame buffer configuration.

Output from the system is via separate red, green and blue (RGB) video components, encoded in standard broadcast video form. Hard copy output on ordinary paper is also available via a laser-driven xerographic color printer.

All drawing, editing and animation are done and all commands are given to the system via the pen and tablet. The operator (often, but not necessarily, a graphic artist) need not have any programming or computer experience. On a standard RGB color monitor directly in front of him, the operator sees the picture on which he is currently working (the "canvas"). On another monitor to his left, the operator sees a second picture (the "control panel") showing a palette of available colors, a variety of brush shapes and sizes, and icons representing various picture editing operations he can invoke (Fig. 2). Colors and brushes are selected and operations are initiated by pressing down lightly on the pen when it is positioned over the desired item—much like pressing a button. At the top of the control panel are three slider scales indicating the hue, saturation, and brightness of the currently selected color.

On an adjacent computer terminal screen to his right, the operator is occasionally prompted or advised with messages such as "Please specify a window . . ." or "Touch color to be replaced . . . ." By using these messages, a naive operator can safely explore the system and easily discover many of its features for himself. There are no other buttons or keys and the operator is required to type on the terminal keyboard only when a name is needed to reference a stored picture.



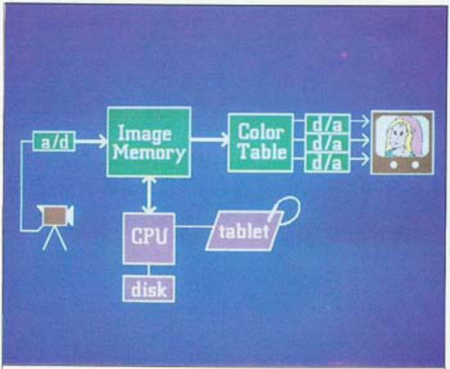


Fig. 1. Overall system block diagram.

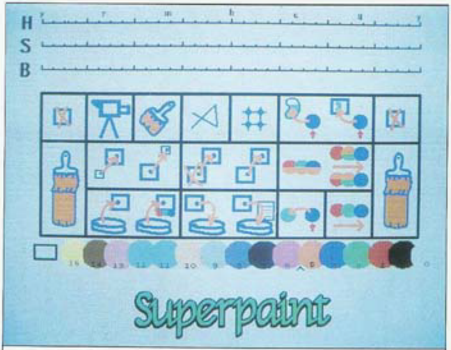


Fig. 2. Control panel.

Graphics and animations may be composed in a variety of ways. Items may be drawn ("videopainted") by the user directly into the canvas picture. Prepared art work or other material may be input to the system via a conventional vidicon camera or other video source. The incoming video can be masked with a rectangular window and automatically reduced to two-colors for a high-contrast version. The resulting subimage can then be used just as if it had been drawn. Text can be added using digitally-stored fonts. Parts of previously created pictures can be recalled and inserted, as can computed or synthetic imagery from other computer programs.

Objects or areas in the picture may be scaled up or down in size, moved, copied, overlaid, combined or changed in color, and saved on disk for future use or erased (see Fig. 3). Also provided are automatic drawing of straight lines of variable width and filling in of closed outlines with a selected color. Line endpoints and positions of moved or copied objects may be automatically constrained to grid points of a specified spacing. This allows easy creation of charts and graphs, etc., and alignment of items in a picture.

Each of the 16 available colors in the palette may be independently adjusted in terms of its hue, saturation, and brightness using the slider scales at the top of the control panel. As the operator adjusts a color, the palette and all areas in the canvas picture containing that color change simultaneously on the displays. By selecting the label to the left of the sliders, the operator can change the meaning of the scales to represent the red, green, and blue components of the color.

HOW IT WORKS

The image memory is arranged in two identical banks (corresponding to the control panel and the canvas pictures), each 480x640 pixels by 4 bits per pixel. MOS shift registers are used and the memory continually recirculates in synchrony with the scanning of the raster. Fig. 4 shows one of these two banks with the recirculation path highlighted. Every memory cycle is a read-modify-write cycle. Thus, the contents of the memory are changed by switching multiplexor 1 at an appropriate time during scanning of the image.

In addition to controlling the disks, the tablet and other peripherals, the cpu (presently a Data General Nova 800 16-bit minicomputer) can also provide a data stream to the image memory. This stream is supplied in synchrony with the raster scanning and usually represents a paintbrush image or a cursor. Data from the cpu is run-length encoded in X (along the scan line) and is expanded by hardware in the image memory. This enables the relatively slow cpu to provide this stream in real time for simple brush shapes and cursors. Overlaying the brush image on the canvas picture is accomplished by simply switching only multiplexor 2 at the appropriate pixel times. Note that since an overlaid cursor is never stored in the memory, no rewriting is necessary when the cursor moves.

Storing (painting) into the picture is done similarly by switching multiplexor 1. In order to accomplish the real-time brush overlay and painting functions, the multiplexor switching must be controlled at every pixel time by the value of the brush pixel. If the incoming brush pixel value is 0 (the background or "transparent" value), then the canvas pixel value is taken. If not, the brush pixel value itself is used. Thus a brush or cursor can have arbitrary shape and will appear correctly over any background.

Digitized incoming video can be entered into the memory by switching multiplexor 1 to input 2. This is also under pixel-by-pixel control via the cpu data path, so that a brush or other image from the cpu can be used to "paint" parts of the incoming video into the canvas picture.

When a picture is loaded from a disk file into memory, it is transferred similarly by the cpu at a rate of about one runcode per scan line each frame time, with (optionally) only nonzero pixels being stored. Thus simple cartoon pictures can be brought into memory in only a few frame times, while very complex ones can take more than 30 seconds.

In the design of the software, considerable attention has been paid to making the system natural and comfortable for the user. Note, for example, that fully half of the image memory is used solely to hold the control picture, thus giving it a similar visual appearance to, and equal stature with, the image being created. A symbolic visual interface is more appropriate to this graphical medium than giving commands by text item selection or by typing or button pushing. The control panel is, of course, itself a picture created and edited on the system. The "buttons" on the control panel can therefore be easily changed to accommodate improvements.

Also, the control picture dims to one half brightness whenever the operator is expected to be directing his attention to the canvas. When selection of a control panel item is expected, a cursor appears and the control picture returns to full brightness as an added cue to the operator. If the operator changes his mind or

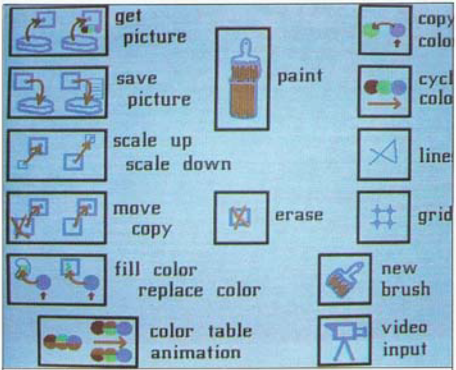


Fig. 3. Control panel icons.

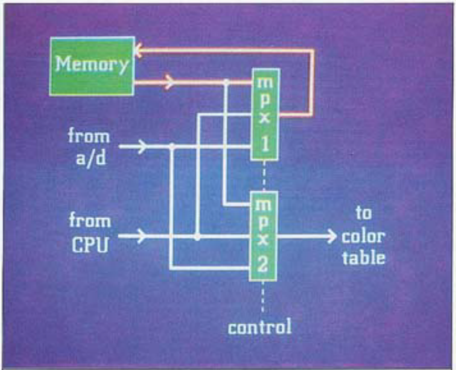


Fig. 4. Image memory block diagram.

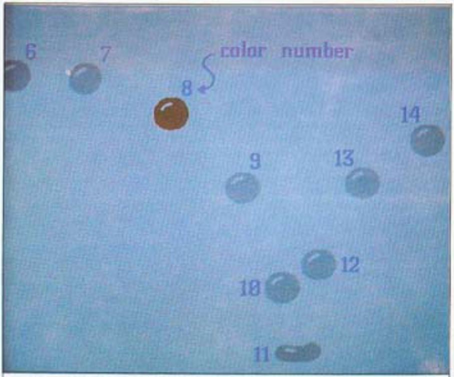


Fig 5. Color table animation.

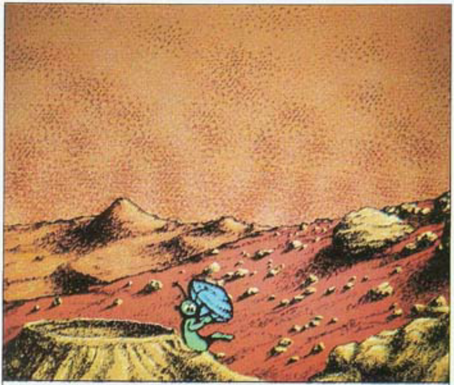


Fig 6. Pioneer Venus videographics.

acts inadvertently, any operation which has been invoked can be aborted and control returned immediately to the control panel by a single tablet stroke.

The value of movement in visual communications is great. Fortunately, even very simple motion in an image can produce a vastly more effective visual communication than a still image. If we do not require elaborate or complex motion in our images, then a simple, highly interactive form of animation can be effected using the color table hardware often included in frame buffer systems.

The *color table* is a small fast memory (usually a bipolar RAM) that holds the red, green and blue values associated with each possible pixel value (Fig. 1). During scanning, each pixel value is used to address the color table and the resulting color component values are passed to the digital/analog converters. The cpu can change the color definitions stored in the color table during vertical or horizontal blanking times.

In the present system, a form of limited but very effective animation is provided which relies on changing the colors of objects hidden within a single picture. Several views of an object are placed at successive positions along its path of motion, each in a different color number or pixel value (see Fig. 5). Initially, all the views are hidden by setting the color table so that each of these pixel values displays a color identical to the background color. The animation effect is then created by manipulating the color table definitions so as to turn on or reveal the hidden objects one at a time in sequence. Notice that successive images can be different in shape and size so that much more than just simple translation of the object is possible. Furthermore, several objects or areas can be in apparent motion simultaneously. Successive images cannot overlap, however.

The operator can manually step through the animation or he can set a speed via the tablet and allow the cycling to proceed continuously. Instead of suddenly changing from background to foreground color, smooth transitions are made between steps in the animation by interpolating in RGB color space over several frame times. The degree of interpolation depends on the speed of the animation. This softening is essential for a pleasing visual effect at all but the fastest speeds. The software also makes it possible to paint into the picture while the animation is running.

Currently, only 10 colors are used for cycling and animation and 6 are static or background colors. (It is worth noting that 10 animating colors are quite sufficient for a wide variety of simple cartoon-like animated graphics.) This means that motion in the image must be effected in 10 or fewer steps. Our application here is not continuous, story-telling animation, but rather simple self-contained graphics such as shown below.

A GRAPHIC AMPLIFIER

Artists and nonartists alike have responded positively to using the system. Most people quickly adapt to the videographic medium and are able to create interesting drawings and graphics with only a few minutes of experience. The medium acts somewhat like an amplifier—it expands greatly the range and scope of both the trained graphic artist and the nonartist.

The illustrations on the opening pages of this article and below show examples of animated graphics by artist Damon Rarey for use during the Pioneer Venus mission. Like all the figures shown here, these pictures were created on the system and can be printed by a laser xerographic printer.

Limitations of the system are numerous and apparent after some use. Most annoying, of course, are the jagged edges often present in the picture due to the limited resolution and lack of softness of the digital medium. Techniques now exist for eliminating these quantization effects, but using them fully in a highly interactive system is beyond the capabilities of this hardware. Other desirable features not present in the current system include: multiple overlays (like a cartoonist's cels), the ability to deal with full-color natural images as well as flat color, smooth scaling (zooming), and a more general animation capability. ✪

AUTHOR'S ACKNOWLEDGEMENT

The author would like to thank Damon Rarey for his enthusiastic use of the system and for many important comments and suggestions. Thanks are also due to Bob Flegal and Alvy Ray Smith for their early work on this system and for numerous useful ideas and discussions.

RICHARD G. SHOUP

Dr. Shoup is a member of the research staff at the Xerox Palo Alto Research Center. He has been with Xerox since 1970 and was previously at Berkeley Computer Corp. He has a PhD in computer science and a BSEE, both from Carnegie-Mellon Univ.