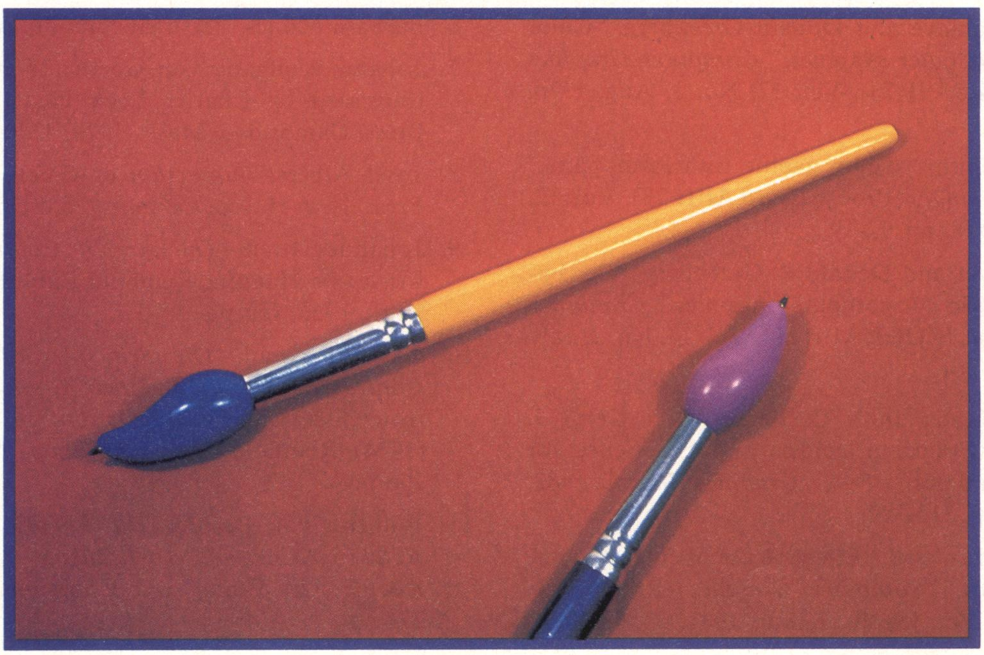
The paint interface: Is it a vehicle to easy electronic art or an obstacle to creative involvement with computer graphics?

The Paint Problem

Copper Giloth and Jane Veeder

Computer artists/programmers



Is this a pen or a brush? Photo credit: Copper Giloth

In recent years the paint interface has been transformed from one of a range of task-optimized tools into a monolithic paradigm for access to creative computer visuals in general. This is distorted. In this article we articulate what we see to be the true nature of this widely distributed interface and place it in proper perspective to art and the medium of computer graphics. We outline some central problems with today's commercial paint systems and offer suggestions for structural improvements.

By describing the role of artists in relation to the technological evolution of their media and by developing a model of the medium of computer and how computer artists interact with it, we aim to stimulate thinking about existing alternative approaches and what new directions interface research and development might take.

The role of artists

As used here, the term *artist* will refer to the fine artist or a person whose interaction with a chosen medium or range of media is driven by personal desires, whether visual, aural, tactile, conceptual, political, etc. We will use preceding modifiers that identify a specific medium used by a fine artist (e.g., computer artist), but will use the term *illustrator* or *graphic designer* to designate those roles centered around the production of visual results specified by an employer or client. Obviously, the artist may perform

0272-1716/85/0700-0066$01.00 © 1985 IEEE

in the role of both illustrator and graphic designer, but those who do, usually make a clear distinction regarding any particular project. The term *visual worker* encompasses all of the above.

The cultural work of artists is to develop and present a model of personal participation in the image life of their culture. Since it is our premise that this image life is based in contemporary media, artists must use the most advanced technology available or they will not operate at equal strength with other model-projectors, e.g., broadcast television.

The preceding concepts may conflict with many readers' notions of artists and computer art. As Gene Youngblood has said, ". . .ironically—most of what is understood as computer art today represents the computer in the service of those very same visual art traditions which the rhetoric of the new technology holds to be obsolete."[1] We are concerned here with how artists interact with the medium of computer as "true artists," attempting, as Root-Bernstein put it, to "invent new tools, problems, and problem solutions"[2] in ways that are intimate and unique to it.

Artists and technology

The "Art and Technology" era of the late 60's and early 70's was not the first meeting of art and technology, but rather a very open, creative period of initial contact between artists and the new medium of computer. Art is one of the many technological activities of human culture. Similar Art and Technology events have occurred throughout history as artists jumped into the new aural/visual technology.

History is rich with examples of artists developing technologies, tools, and design strategies to evolve their perceptual skills and realize their visual goals. Perspective, one of the most powerful technologies for organizing visual information, was originated by 15th-century architect Filippe Brunelleschi.[3] Albrecht Durer, learning the principles of perspective from the Italians, built a device using strings running through a screen for converting a view of 3-D objects into 2-D drawings in perspective (see Figure 1).[4]

The golden section is an ancient design strategy for creating order and harmony as translated from nature to

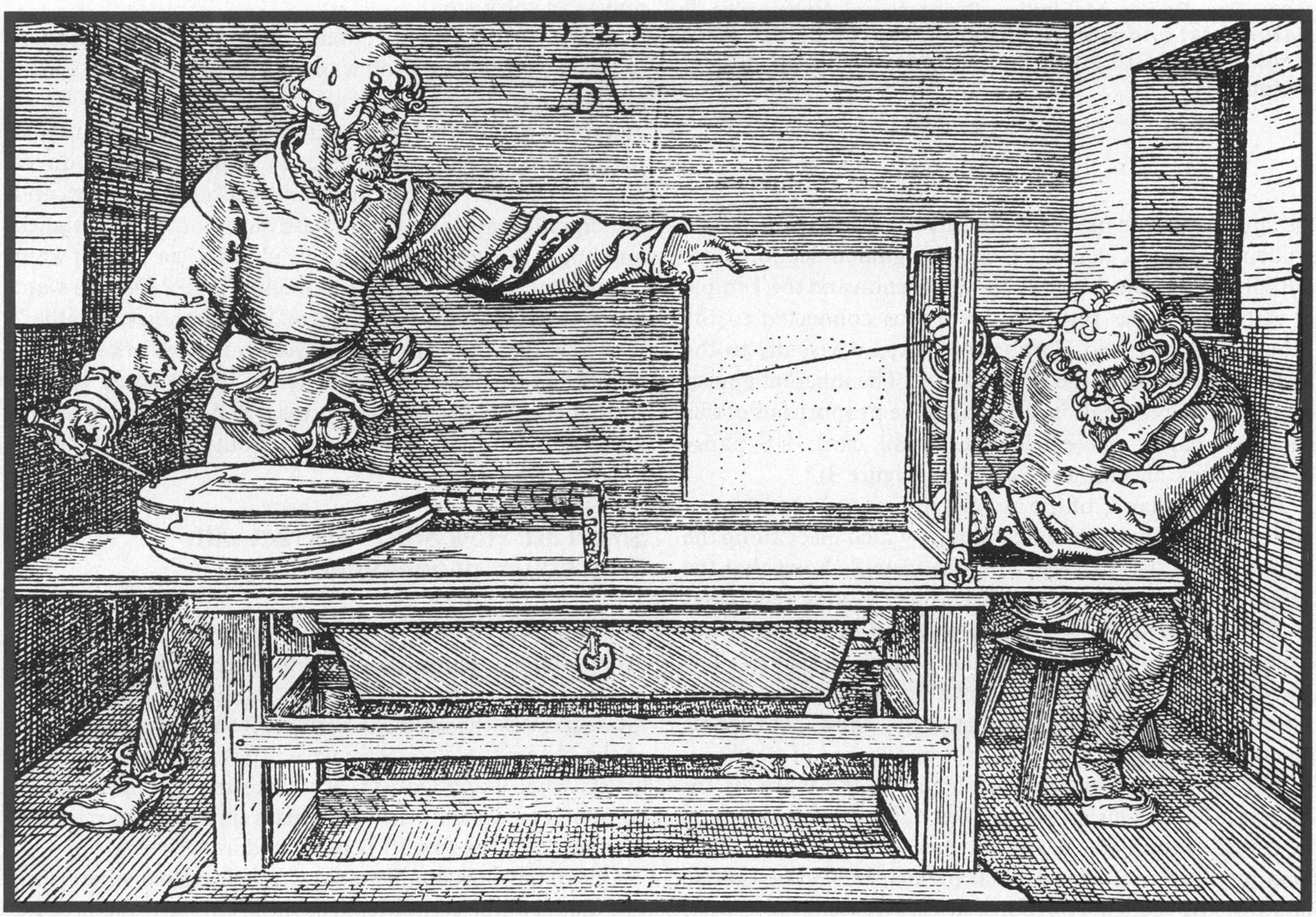


Figure 1. "Artist Painting a Lute," a woodcut by Albrecht Durer, from Durer's work *The Art of Measurement*, fourth book, published 1538.

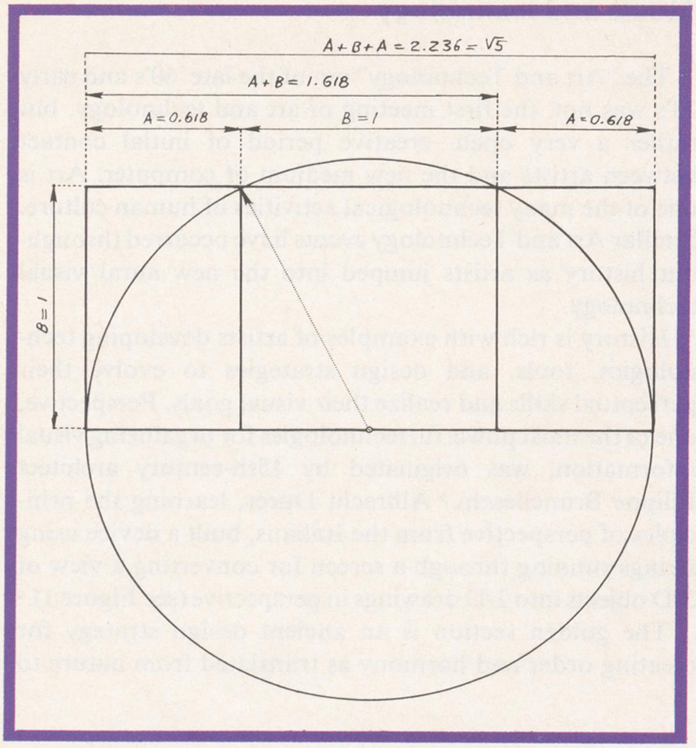


Figure 2. Classical construction of the golden section, from *The Power of Limits—Proportional Harmonies in Nature, Art and Architecture* by Gyorgy Doczi. Rectangles 1 × 0.618 and 1 × 1.618 are reciprocal golden rectangles.

the current human model of the universe (see Figure 2). The golden rectangle, derived from the golden section, was initially used in the design of the Parthenon and the Temple of Athena. In the 16th century it was connected to the Fibonacci series, and Hambridge linked it visually to the logarithmic spiral in the 20th century. The inherent appeal of the golden rectangle is reflected in the proportions of our standard paper, drivers license, credit card, US paper money, bank checks, and 747's (see Figure 3).[5]

The entire plan C of the aircraft in Figure 3 fits into two pairs of reciprocal golden rectangles, which meet along the center line of the fuselage. The side view B shows that the length of the body is contained in five golden rectangles plus a reciprocal. Both front view A and each wing D are contained within a pair of equal golden rectangles. The wave diagrams show the various interrelationships between the plane's proportions, and the graph shows how these proportions approximate the root harmonies of music.

Another design strategy is shown in the work of the architect Le Corbusier. He worked out an ordering system he called "Modular," based on three points of the human anatomy: top of the head, solar plexus, and tip of a raised hand. Using these proportions, he created an infinite series of combinations and applied them to architectural design. This system influenced German/Swiss graphics designers in the areas of modular systems and grids.[6]

In addition to being technological innovators and popularizers, artists have always been quick to seize on new developments. How to separate the Impressionists from the development of lead-tube paints with screw-tops that suddenly supported the mobile painter?[7]

The medium of computer

"The protean nature of the computer is such that it can act like a machine or like a language to be shaped and exploited," said Alan Kay in a *Scientific American* article. "It is a medium that can dynamically simulate the details of any other medium, including those that cannot exist physically. It is not a tool, although it can act like many tools. It is the first metamedium, and as such it has degrees of freedom for representation and expression never before encountered and as yet barely investigated."[8]

The computer is a complex medium, incorporating many levels of invested intelligence: VLSI manufacturing techniques, chip design, hardware system design, and varying levels of software support. While a number of artists have plugged in at the levels of chip, board, or system design, the vast majority connect at some level of linguistic interface: software.

Each level of software communication offers a different experience. Where do you want to live? Do you want to use a highly mediated software interface comparable to Kay's metaphor about writing an essay using only preconstructed paragraphs,[8] or do you want to experience the direct relation between stark expressions of Boolean algebra and the digital image-processing hardware you have designed and built? Do you want to "peek 'n' poke," or do you want to mind-juggle C's flexible indirection space? Do you want to be compiled or interpreted or have a choice of either? Regardless of the level you choose, you (artist or other) are dependent on the intelligence levels below yours for the formal capabilities at your disposal and dependent on the levels above for the organization of output to the world.

Smart art: How artists interact with the medium of computer

The phrase *smart art*[9] refers to artwork produced by an artist who consciously devises an intelligent strategy or control structure to organize the (visual) information and interacts with the invested hardware/software intelligence of the system to do so. Here we will describe the activity of artists who, like scientists, are using the metamedium of computer to simulate their personal versions of the medium of computer; the spinoff is presented as their art.

One of the most obvious characteristics that distinguishes our model of the computer artist is an emphasis on process rather than artifact. The commercial paint system is a tool optimized for minimal user involvement and maximum production output; by contrast, computer artists treat the

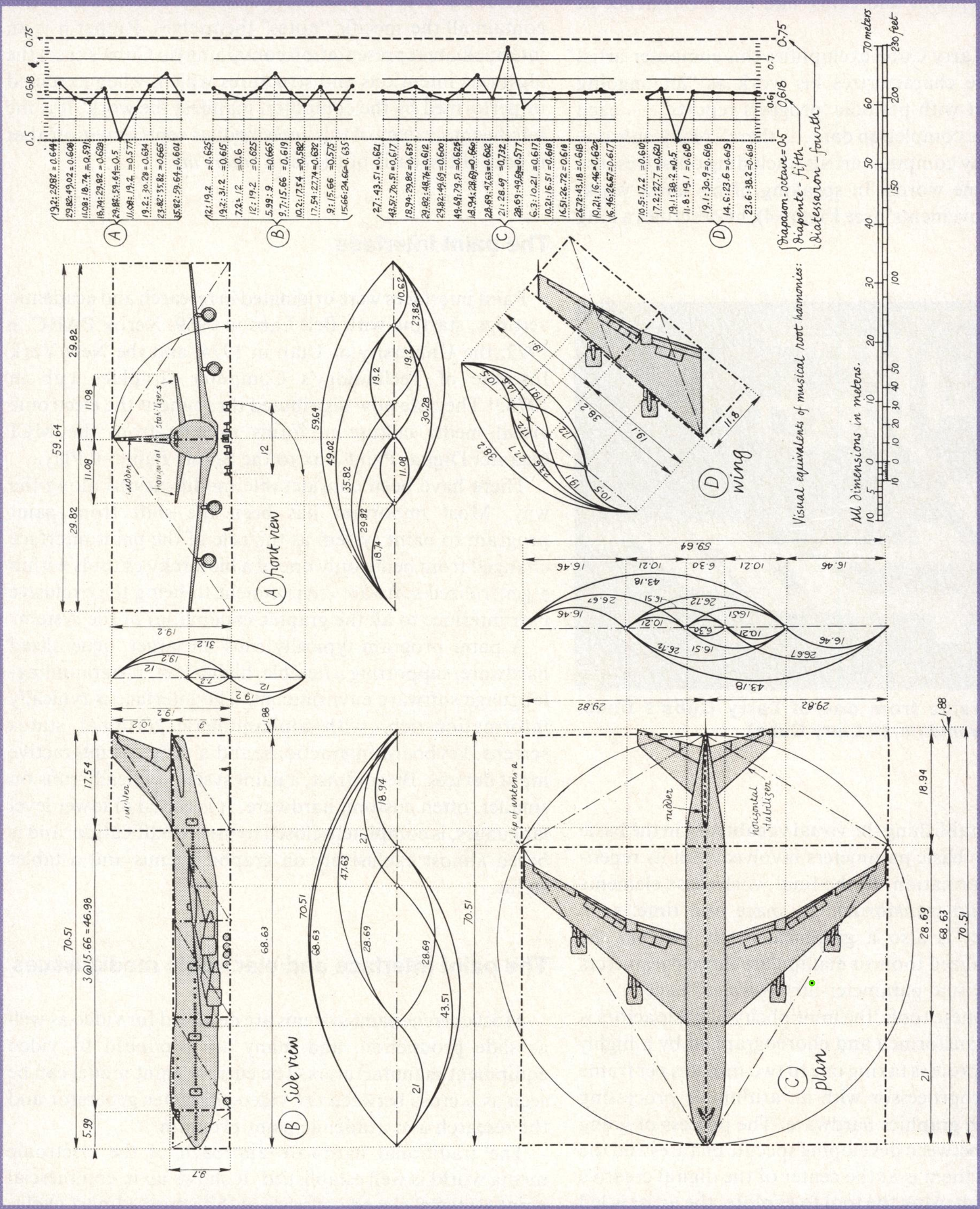


Figure 3. Knowledge of aerodynamics, of the structural strength of materials, of the physics and chemistry of engines, and of business skill would seem the most obvious considerations in the design of an aircraft. The design of the Boeing 747, however, also reveals harmonious proportions strikingly similar to those we have seen in natural forms and works of art.

computer as a generalized medium within which to develop facile tools over which they exercise intimate control. The production goal of these tools, like scientific tools, is the field itself, i.e., artists are developing tools to explore the creative digital process, investigate the medium, evolve their own perceptions and skills, and extend evidence of this to others.

Filmmaker Larry Cuba exemplifies this computer-artist model when he characterizes his work as "an ongoing research project with periodic 'progress reports' . . . each title could be the completion date . . . there's really only one project."[10] Many computer artists, including ourselves, use much these same words. In speaking of his latest work, "Calculated Movements" (see Figure 4), he describes a long

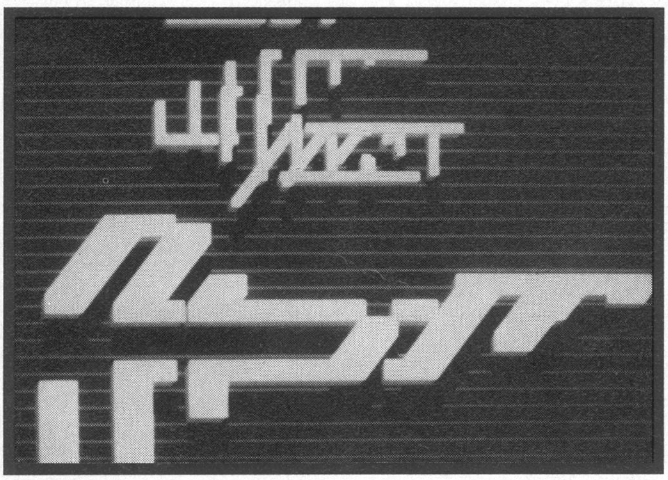


Figure 4. A frame from one of Larry Cuba's films, *Calculated Movements*, (© Larry Cuba).

process: first, establishing the visual vocabulary in the basic element and the basic parameters involved with its repetition and transformation. In the final work these elements are repeated and transformed in space and time, so a graphical object is also a graphical event. Second, he develops generalized tools to manipulate those parameters to explore the visual parameter-dimensioned space.

By means of these tools, the multiplicity of basic actors is algebraically transformed and choreographed by a highly computational process taking one to two minutes per frame on a Z-80 microprocessor with an arithmetic processing unit and custom graphics hardware. The process of going back and forth between developing specific pictures and the tool to develop them is at the center of the digital creative process. Through using the tool to explore, the artist is led to something the tool can't do. Then the tool must be modified. Developing and using explorational tools is not a matter of random playing, but of designing the search approach, knowing where to look, seeing something interesting, and knowing what to do from there.[10]

The computer artist does not simply compose instructions for the computer to produce the previsualized artwork. The computer is not merely a performance/output instrument like a player piano, but is used in development in the same way that most composers play an instrument to hear sounds so they can compose music. The final result is not quite a "score" in the musical sense, because it does not contain all the specific "notes" themselves. Rather it is an intermediate representation containing (in Cuba's case) the algebraic intentions and structures which, when executed or performed by the computer, manifest the work. It is the intelligent strategy of this 'intermediate representation' that is part of the basis of the term *smart art.*

The paint interface

Paint interfaces were originated in research and academic settings, starting with Bell Labs in 1969, Xerox PARC in 1972, the University of Utah in 1974, and the New York Institute of Technology's Computer Graphics Lab in 1975.[11] They are now distributed throughout the electronic visual media culture in forms ranging from the MCI Quantel Digital Paint Box to the Smurf Paint 'n' Play.

There have been considerable modifications along the way. Most important has been the shift from paint program to paint system as the role of the paint interface changed from being only one of a hierarchy of tools within a generalized software environment to being the exclusive user interface to all the graphic capabilities of the system.

A paint program typically runs on larger, generalized hardware, supporting a flexible, higher level programming-language software environment. The interface is typically information-rich, with alphanumeric terminal status screens, keyboard interactions, and a range of interactive input devices. By contrast, a paint system typically runs on smaller, often custom, hardware. It is coded in lower level languages, is completely closed to user modification, and is based almost exclusively on graphic menus and a tablet stylus.

The paint interface and electronic media issues

Most current paint systems are designed for video as well as slide production, and many are produced by video equipment manufacturers. The current paint model can be seen as a cross between the video character generator and the research and academic paint program.

The traditional hardware approach of the electronic media world is well established. It shows up in commercial paint systems' closed software architecture, almost exclusive reliance on menu-picking, and the shunning of alphanumeric screen and keyboard interactions.

Just as drama became the content of cinema, and cinema that of television, the pattern we learned from Marshall McLuhan,[12] so traditional 2-D art media have become the initial content of the new medium of computer graphics.

The most obvious evidence of this apparently automatic process is the vocabulary: *paint, rubber stamp, cut and paste.* More fundamental is that the designs of paint interfaces and computer graphics hardware often faithfully reproduce the limitations of the older medium, e.g., no automatic save/recover features and frame buffers with a "frame" of blanking color around the edge of the video raster. Assuming the notion that the computer is a metamedium, the paint interface is actually a simulation of traditional 2-D media constructed within it, just as CAD is the simulation of the traditional medium of engineering drafting.

As computer simulations, all paint interfaces offer considerable gain over traditional media in facile color manipulation, image data-banking, speed in making changes and testing variations, image transformations and combinations, and access to a variety of output devices and hardcopy media. These gains are of great value to the gigantic media world (including broadcast television; advertising; industrial, educational, and institutional media; etc.), and the media appetite for computer-generated visuals is growing rapidly. Fast production time, flexibility, and cost effectiveness are the bases of this desire. The need to have even modest visuals project an "advanced" image and to merge aesthetically with other electronically generated images is also strongly felt.

Paint interface evaluation

The widespread use of the paint system as a computer simulation of traditional 2-D visual media makes it worthy of serious analysis. At this point we will touch on some central issues.

Menu control structures. Even though in theory menus support easy entry, the overhead of menu transactions can become burdensome, especially for the experienced user. Menu control structures often show a lack of attention to task grouping or session flow, and little use is made of graphical means to lead a user through required multiple selections. Icons or buzzwords standing for the same operation vary wildly from one system to another and rarely conform to either the computer graphics term for the operation or the already existing term from the appropriate field, such as graphic design or typography.

In keeping with the hardware bias of their design, the graphic menus of paint systems operate as virtual function-keyboards, which rarely take advantage of their innate software flexibility. A custom menu cannot be user-configured with a collection of task-related functions originally on separate menus. A sequence of existing functions cannot be user-defined into a new single function. By contrast, says Bert Herzog, many current CAD interfaces offer capabilities for user modification of menu groupings, labels, and the ability to record, name, and replay graphic procedures in a variety of ways.[13]

Restricted interaction. The range of task-optimized interactive devices described repeatedly in such human factors literature as the article by Foley, Warner, and Chan,[14] does not seem to have touched the commercial paint system. Where the stylus is used exclusively to model the interaction of a variety of other devices, says William Buxton, efficiency is lost via the "null" factor, as the user must (consciously) shift gears from one to the next, and no advantage is taken of (unconscious) muscle memory, learned by using specific devices for specific tasks.[15]

Quality of the simulation: loss of traditional control and skills. Much of the control, skill, and access to the wide variety of tools and design strategies traditionally enjoyed by painters, illustrators, and graphic designers is lost in the translation to anything but the most advanced, expensive, and recent computer graphic design systems. Bigelow mentions a fact that extends beyond the subject of digital font design when he says that ". . .the designs of such fonts are often the work of people untrained in letterform design, while traditional lettering artists have rejected computers because the tools and output media of digital typography have been so clumsy and crude."[16] One of the most obvious and pervasive examples is that the two-handed user is reduced to single-handedness. A universal feature of the paint interface is the *airbrush,* but without another hand to position screens, stencils, and masks, the electronic simulation does not begin to approach the versatility and power of the mechanical tool.

Information and the visual worker. It seems obvious that our access to the power of digital techniques for originating and manipulating visual information can only be as good as the communications interfaces we design. Unfortunately, the great extent to which paint systems fail to take advantage of the computer is related to a major misconception of the creative process practiced by all visual workers as a noninformational process. This results in a decided lack of precision and control for the user, because the interface does not provide methods for inputting specific information (draw a box *n*-pixels wide) or for outputting specific information (how wide *is* the box I'm drawing?). A common and blatant example is user difficulty in locating the center of the screen (where am I?).

General design guidelines: We offer the following for improvement of the paint interface more or less within the current model but supporting greater power, efficiency, and precision for illustrators and graphic designers.

1. **Provide an information-rich environment.**
   - user-accessible parameter structure
   - both interactive and keyboard input methods for specifying parameters
   - terminal screen as concurrent parameter status output
   - graphic screen interrogation capability
   - sophisticated image-data storage and retrieval features

2. **Develop a clear and minimal menu structure.**
   - generalized tools that integrate options, variations, and modes to reduce menu interactions
   - concurrent parameter controls over generalized tools via a range of interactive input devices
   - features that take advantage of user concurrency
3. **Support user-customization of the environment.**
   - customize, save, and retrieve menus
   - new function sequences: name, save, retrieve, and add them to menus
   - environmental or global controls, e.g., user-specified graphic windows, tablet rescaling capability to facilitate tracing pictures, etc.

For artists who have focused on computer graphics as their "medium of obsession," nonextensibility of the usual paint interface offers trivial access to the field.

Is paint an artist interface?

For a painter wanting to connect with the distribution and exposure of electronic media, but still operate within at least some semblance of traditional media, a paint system would seem to present the perfect vehicle. For artists whose central orientation is videotape production and who collage visuals from various sources, the paint system would simply be another piece of hardware, and accepting its high level mediation would be no more troublesome than our acceptance of the fixed interfaces of our videotape recorders.

For artists who have focused on computer graphics as their "medium of obsession," the nonextensibility of the usual paint interface offers trivial access to the field, allowing neither state-of-the-art computer graphic techniques nor the opportunity to develop new ones.

For such artists as Veeder, who feel that digital works of art must be interactive, the paint interface is irrelevant, because it is not an interface to a software development environment.

High level mediation. One of the main criteria with which artists are evaluated is the depth of involvement with their medium or media and how innovative they have been at organizing that involvement. Heavily mediated environments (e.g., paint-by-the-numbers kits) are not generally thought to be meant for artists, and their use of same is not valued by the art world unless used with startling originality for the purpose of conceptual comment. Given the high level mediation of the paint system, to what extent can the artist do anything "new"? Given the reduced communication bandwidth, how much of an artist's intelligence can be encoded in the resulting work?

System nonextensibility. Innovative artists throughout history have experimented with the lowest accessible levels of their medium, inventing new application and construction techniques, modifying traditional tools, bulldozing huge areas of land around, etc. The closed nature of paint systems prevents user-customization to the point that frequently an entirely different and more powerful computer is required for software development.

One of the best and most well-known artist's works produced at the New York Institute of Technology's Computer Graphics Lab, Ed Emschwiller's "Sunstone," used the NYIT Paint Program. In addition the artist had access to a collection of user-friendly, flexible, intelligent tools with which he interacted to develop custom image transformation, texture mapping, and animation routines—namely, the software development staff.[17] This additional tool collection does not generally accompany the commercial paint system.

Different goals. It is important to remember that the early paint interfaces were developed for very specific applications, e.g., NYIT's goal of computerizing feature-film animation. The computer graphic "Scan 'n' Paint" process is five to eight times as efficient as traditional means for painting solid-color areas on animation cels, reports Doris Kochanek.[18] Paint and other highly mediated interfaces work best when they are designed for a process that is reducible to a well-defined algorithm. This is fine as a task-optimized tool, but it is not what either art or science is about.

Though the computer's ability to process large amounts of information rapidly is of great value to anyone who uses it, artists are not using the computer simply because it makes art more efficient. In fact a late 1950's series of automatic painting machines, French sculptor Jean Tinguely's "Metamatics," made wry comment on this subject.[19] The paint system's public image of extreme mediation extends the unfortunate suspicions of both the art world and the public that computer art is the product of a fully automated process.

User extensibility. The related but neglected concept of user extensibility, as described by Thomas A. DeFanti,[20] fits perfectly the artist's concern with self-evolution, and this can apply to other types of users as well. In addition to beginner-to-adept evolution, this concept encompasses the phenomenon of user-evolving-into-developer, e.g., the weaver using a weaving-design software package, becoming increasingly dissatisfied with its limitations, learns to program, modifies the package, develops new software for personal work, and ultimately offers packages or tool sets to other weavers. If language interfaces were more sophisticated, this would be common. Owens comments in *The Design Journal,* "The designer who is also a programmer can combine both functions; seeing new possibilities as he begins to develop the program, and incorporating these ideas into more extensive and refined design goals."[21]

The ultimate product of in-depth human-computer interaction is the human transformed. How many interfaces design for this product? As we have mentioned, there are many levels at which to interact, but to access the simulation power of the computer, to enter the creative development loop Cuba and other computer artists describe, one must program.

The linguistic interface

"Almost all paint/animation systems are menu driven," say Hutzel and Peters, "thereby allowing the artist to concentrate more on the creative process."[22]

What evidence supports the notion that programming is an obstacle to creativity? Since when do creative workers not think about the medium, tools, etc. (at some level), while creating? When artists (or others) are programming, they are always focusing on the (creative) goal; they are in intense contact with their medium's general-purpose interface. When Hancock says, "Thinking in musical terms can help develop your technique for programming computers,"[23] he is in effect saying "connect your creative process to conscious thought." The computer is a medium of the mind, and thinking is certainly central to its manipulation. The paint system model, with its emphasis on the input of information via drawing, has convinced many people that the mind's control of the medium can only be as sophisticated as the user's hand.

There has been a variety of approaches to interfacing with graphics that offer alternatives to the closed, monolithic paint model. Graphic procedural and command language interfaces offer greater control, but have been unforgiving and unwieldly. As we've discussed, the current CAD model offers a combination of the more flexible menu system with limited procedural language access. Generalized languages that support rich graphic capabilities enjoyed a flurry of research activity in the late 1970's, says Sylvia Carol Lea,[24] but very few were actually released as products. One that did not take Kay's requisite 10 years to release was "ZGRASS," a language interpreter designed specifically for graphics, real-time animation, user evolution, and optimized for custom videogame hardware. This affordable, Z-80 based sports car of a system spawned a subculture of creative users who developed software toward a diverse range of goals, including custom language commands. All software packages were written in the resident language and so were frequently customized by users. Even though it is no longer commercially supported, the subculture continues to evolve through this system because of its software generality and formal graphics capabilities, which in other than resolution and bit depth have yet to be equaled.

ZGRASS, a sports car of a system, spawned a subculture of creative users who developed software toward a diverse range of goals. . .

The development of high-level and very high-level languages presents the most global approach to the problem of future access to computing. In the course of our research, we have not discovered evidence that the academic and research world is currently doing much in this area with regard to graphics. There is activity in the specialized area of animation, which we hope will offer some evolved interfaces for future animation hardware.

Artist as a user model

We propose the consideration of computer artists as informative models of the future computer graphics user. We have described both the flexible, generalized environments, with rich informational feedback, in which artists like to function and their propensity for progressive tool development and customization. These come closer to a description of academic and research computer graphics environments than to the commercial systems supposedly designed to make art easy.

Future development of graphics interfaces can benefit not only from using computer artists as models for future users, but from direct participation as well. We have participated in the development of two graphics language environments and have a clear idea of the contribution evolved and energetic visual desires can make to the process. Likewise, the acclaimed quality of the MacPaint package is directly attributable to the participation of such an artist as half the development team.[25] At the same time, artists must become truly involved with computer graphics (i.e., jump in, learn programming, etc.) to have an informed perspective to lend to the development process.

Conclusion

We have evaluated the current paint system interface design and offered guidelines for improvement. Further, much can and should be done to evolve the simulation of 2-D graphic media beyond the monolithic paint interface to a collection of focused tools and tool sets strategically designed for specific tasks and set in a global graphic development environment. As discussed, this involves not only the generation of enlightened alternatives from the research world but also major shifts in the design approach of commercial system manufacturers.

One of the most encouraging trends is in custom VLSI development for graphics and, for the first time since the video-game chip designs of the mid-1970's, real-time animation. We hope that these hardware advances can be incorporated into systems that will offer more open, global interfaces.

Recent successes of such products as Thinktank and Sidekick, which provides users random access to a ROM resident tool set of frequently used capabilities, might loosen up the commercial notion of a viable product, suggest Bartimo and Mace.[26] More generally, it indicates that users value self-directed access to their computers' capability.

The domination of the paint interface emphasizes the absence of available interfaces to computer graphics that are not in the service of a simulation of past media. The quality of future graphics interfaces will depend to a high degree on our ability to recognize in the medium of computer the power to simulate an advanced version of ourselves and to develop tools worthy of that user. ■

Acknowledgements

We are grateful to many people for advice, information, and sources, in particular Laura Scholl and Stephen Joyce for their encouragement and criticism.

References

1. Gene Youngblood, "A Medium Matures: The Myth of Computer Art," SIGGRAPH 83 *Exhibition of Computer Art Catalog,* July 1983.
2. Robert Scott Root-Bernstein, "On Paradigms and Revolutions in Science and Art: The Challenge of Interpretation," *Art Journal,* College Art Association of America, Vol. 44, No. 2, Summer 1984, pp. 109-118.
3. Georges Coppel, "Old Perspective and New Vision," *Leonardo,* Vol. 15, No. 4, Autumn 1982, pp. 269-276.
4. R.E. Schofield, "The Eye of the Beholder: A Critical Examination of Some Cultural Aspects of Scientific Creativity," *Leonardo,* Vol. 16, No. 2, Spring 1983, pp. 133-137.
5. Gyorgy Doczi, *The Power of Limits: Proportional Harmonies in Nature, Art, and Architecture,"* Shambhala, Boulder, Col., 1981, 150 pp.
6. A. Hurlburt, "the grid," Van Nostrand Reinhold Co., New York, 1978, 96 pp.
7. A. Smith, "The New Media as Contexts for Creativity," *Leonardo,* Vol. 17, No. 1, 1984, pp. 46-48.
8. Alan Kay, "Computer Software," *Scientific American,* Vol. 251, No. 3, Sept. 1984, pp. 52-69.
9. Personal communication, Phil Morton, 1976-1983.
10. Personal communication, Larry Cuba, Dec. 1984.
11. Alvy Ray Smith, "Paint," Technical Memo No. 7, Computer Graphics Lab, New York Institute of Technology, July 20, 1978.
12. Marshall McLuhan, *Understanding Media: The Extensions of Man,* McGraw-Hill, New York, 1964, 364 pp.
13. Personal communication, Bert Herzog, Computer Graphics Consultants, Jan. 1985.
14. J. Foley, V. Wallace, P. Chan, "The Human Factors of Computer Graphics Interaction Techniques," *IEEE Computer Graphics and Applications,* Vol. 16, No. 11, Nov. 1984, pp. 13-48.
15. W. Buxton, "Lexical and Pragmatic Considerations of Input Structures," *Computer Graphics,* Jan. 1983, pp. 31-37.
16. C. Bigelow, "Font Design for Personal Workstations," *Byte,* Jan. 1985, pp. 255-270.
17. Personal communication, Ed Emschwiller, Jan. 1985.
18. Doris Kochanek, "Computers and Animation," a lecture at the University of Waterloo, Waterloo, Canada, Jan. 23, 1985.
19. J. Benthal, *Science and Technology in Art Today,* Praeger, New York, 1972, 180 pp.
20. Thomas A. DeFanti, "Language Control Structures for Easy Electronic Visualization," *Byte,* Nov. 1980, pp. 90-106.
21. C. Owens, "Should Designers Learn to Program?" *The Design Journal,* Vol. 1, No. 1, 1983.
22. I. Hutzel and C. Peters, "Paint/Animation System Survey," *Computer Graphics World,* Vol. 7, No. 8, Aug. 1984, pp. 11-28.
23. Herbie Hancock, "The Musician/Machine Connection," *Digital Deli,* Workman Publishing, New York, 1984, pp. 221-222.
24. Sylvia Carol Lea, "Fecit—A Structured Language for Describing Computer Generated Scenes," Masters Thesis, University of Waterloo, Waterloo, Ontario, Canada, 1983, pp. 4-24.
25. Bill Atkinson and Susan Kare, Microcomputer Graphics Panel, *Computer Graphics* (Proc. SIGGRAPH 84), Vol. 18, No. 3, pp. 175-186.
26. Jim Bartimo and Scott Mace, "Winning Big by Thinking Small," *Info World,* Vol. 7, Nos. 1 & 2, Jan. 7 & 14, 1985, pp. 30-41.

Additional Readings

1. DeFanti, Thomas A., "The Mass Impact of Videogame Technology," *Advances in Computing,* Academic Press, New York, Vol. 23, 1984, pp. 93-140.
2. Edwards, Edward B., *Pattern and Design with Dynamic Symmetry,* Dover Publications, New York, 1932, 132 pp.
3. Tufte, Edward R., *The Visual Display of Quantitative Information,* Graphics Press, Cheshire, Conn., 1983, 197 pp.
4. Richmond, Sheldon, "The Interaction of Art and Science," *Leonardo,* Vol. 17, No. 2, 1984, pp. 81-86.
5. Pollack, M., editor, *Common Denominators in Art and Science,* Aberdeen University Press, Great Britain, 1983, 191 pp.
6. Kurth, W., *The Complete Woodcuts of Albrecht Durer,* Dover, New York, 1963, 346 pp.
7. Di San Lazarro, G., *Klee, a Study of His Life and Work,* Praeger, New York, 1964.
8. Vasari, *Lives of Painters, Sculptors & Architects,* Vol. 1 and 2, Dent Press, London, 1965, 364 pp.
9. Kay, Alan, "Learning vs. Teaching with Educational Technologies," *EDUCOM Bulletin,* Fall/Winter 1983, pp. 16-20.
10. Meggs, P.B., *A History of Graphic Design,* Van Nostrand Reinhold, New York, 1983, 511 pp.
11. Bigelow, C., and D. Day, "Digital Typography," *Scientific American,* Vol. 249, No. 2, Aug. 1983, pp. 106-119.
12. Dondis, D.A., *A Primer of Visual Literacy,* MIT Press, Cambridge, Mass., 1973, 194 pp.
13. Reichart, Jasia, *The Computer in Art,* Van Nostrand Reinhold, New York, 1971, 96 pp.
14. Shneiderman, Ben, "Direct Manipulation: A Step Beyond Programming Languages," *Computer,* Vol. 16, No. 8, Aug. 1982, pp. 57-69.
15. Kay, Alan, and Adele Goldberg, "Personal Dynamic Media," *Computer,* Vol. 10, No. 3, Mar. 1977, pp. 31-41.
16. Glinert, E., and S. Tanimoto, "PICT: An Interactive Graphical Programming Environment," *Computer,* Vol. 17, No. 11, Nov. 1984, pp. 7-25.
17. Feiner, S., D. Salesin, T. Banchoff, "Dial: A Diagrammatic Animation Language," *IEEE Computer Graphics and Applications,* Vol. 2, No. 7, Sept. 1982, pp. 43-54.

Copper Giloth is a computer graphics artist/programmer who will be an Associate Professor in the Art Department at the University of Massachusetts, Amherst, teaching computer graphics and video. Her research interests include real-time animation, software tools for art making, graphics languages, and user interfaces for plotters and interactive installations. She has taught at the School of the Art Institute, Chicago, and the University of Michigan, Ann Arbor, and was with Real Time Design from 1981 to 1984 as a software engineer and vice president in charge of application software. She was chair of the SIGGRAPH 82 and 83 Exhibitions of Computer Art. Her computer-animated videotapes, installations, and drawings have been exhibited worldwide.

Giloth received a BFA in sculpture from Boston University and an MFA in Electronic Visualization from the University of Illinois, Chicago. She is a member of IEEE, SIGGRAPH, and CAA.

Giloth can be contacted after September 1, 1985, at the University of Massachusetts, Art Department, Fine Arts Center, Amherst, MA 01003.

Jane Veeder is an independent electronic media artist/programmer living in Chicago. Her personal work is in real-time computer graphics/animation and sound synthesis for interactive installations and videotape. Her work, WARPITOUT, is a permanent installation at the Ontario Science Centre, and her videotapes have been exhibited internationally. Her commercial work has included research/design/development in graphic language environments, user interfaces to application software, and animation. From 1981 to 1984 she was an application software consultant for Real Time Design.

Veeder received her MFA in video from the School of the Art Institute of Chicago and her BFA from the California College of Arts and Crafts. She is a member of SIGGRAPH, served as Audio/Visual Chair of SIGGRAPH 83, and comanaged A/V for SIGGRAPH 79, 81, and 82 conferences.

Veeder can be contacted at 600 South Dearborn Street (#509), Chicago, IL 60605.