FOREWORD

Computer Graphics as a Way of Life

In 1974, Ted Nelson wrote a book called *Computer Lib/ Dream Machines* that successfully intertwined the personalities and technologies of computer graphics. It became a cult classic, the only book on computers anyone ever gave as a present in the 1970s. There have been many books on computer graphics since then, but this book, *The Algorithmic Image: Graphic Visions of the Computer Age,* is the long-overdue successor to Ted's whole-computer catalog. Ted was my roommate in the summer of 1974; I helped paste up the book and glue the signatures together. I am also happy to be part of this celebration of the people and technology of computer graphics.

1974 was a good year for computer graphics. The first SIGGRAPH conference was held in Boulder, Colorado, in July of that year. Ed Catmull and Fred Parke showed their amazing three-dimensional hands and faces done at the University of Utah. Ted Nelson, Dan Sandin, and I showed the first computer-graphics videotape presented at a SIGGRAPH conference; it accompanied a paper entitled "Computer Graphics as a Way of Life." The conference had 500 attendees, and we had to bring all the video gear out in a university station wagon. Today, SIGGRAPH attracts over 25,000 people and spends over $600,000 for audiovisual support. Everything has changed, except that computer graphics is still our way of life.

However, many of the images printed here are beautiful and inspiring simply *because* they represent simulations or are otherwise computational. Many of these images would never have been discovered by an illustrator, and this is the most important part of computer graphics' appeal for artists: It represents new processes for creation. Paint systems yield an increase in speed of experimentation that truly changes the nature of illustration. Collage-type artworks are far easier to put together, and substantial modifications take far less time. Compositing scanned-in and drawn images, changing the size of an image, adding text and layback perspective—all are completely straightforward operations that formerly involved process cameras, stats, glue, and serious attention to workshop cleanliness. The paint system is the word processor of television graphics; you can experiment in less time, so you can experiment more. But the images still have to look good.

Even computer animation is not immune to the "blah" reaction by people outside computerdom. In early 1984, I organized the United States/Canada/Japan portion of the film-and-video show at the Forum des Nouvelles Images in Monaco. It was part of a much larger three-week media extravaganza that included France's top awards for broadcast television shows. I was trying to get into the conference-hotel restaurant, but it had been converted into a temporary television studio to air talk-show interviews with the many celebrities at the event. They wouldn't let me in so, acting a little like George Burns trying to find out what Gracie was up to, I went upstairs to my room and turned on the television. An actress, an actor, and a computer animator, all up for major awards, were on. The animator showed his three-minute piece, after which the excited talk-show host sought the opinion of the daytime soap actor. "Well, it's just animation, after all, isn't it?" he said. The animator replied "No, no! It was done by computer!" whereupon the actress, obviously bored, delivered the coup de grace: "Well, if it was done by computer, what did you have to do with it?" Computer graphics may be our way of life, but it isn't everybody's. Yet.

THE GRAPHICS IN COMPUTER GRAPHICS

To claim that the computer part of computer graphics is more important than the graphics part is equivalent to maintaining that the word processor I'm using now is more interesting than my foreword. Of course, to someone who adopts word processing as a way of life, perhaps it is.

Yet word processing is pretty boring and computer graphics is not, even to the general public. Why? Word processing does not change the expression of written ideas and does not really give any new ways to get to the ideas. Authors just type away on word processors, although debugging and production are significantly simplified (of course, having everything written in computer-readable form will greatly simplify document access and cross-referencing in the future). Except in certain areas of research, people do not write algorithms to produce written material. Computer graphics, on the other hand, is a tool for exploration in which the ever-increasing capabilities for rendering images and animations push at the boundaries of the possible and the probable.

Television, educational and otherwise, has largely degenerated into a medium of talking heads. I attribute much of the excitement and promise of computer graphics to the fact that it is a poor medium in which to do talking heads. Graphics has to communicate more abstractly, and do it well, to survive. The quest for realism in computer graphics is not just for duplicating reality, but for creating any reality with any arbitrary set of rules governing it. That is what draws people into computer programming, and it is exactly what makes computer graphics so exciting. We have had the capability to construct arbitrary rule sets with musical instruments for 10,000 years, and dance has existed as an abstract communication form for even longer. The ability to generate arbitrary animations, however, is new and wonderful.

Actually it's not so new. In the 1940s and 1950s, cartoon animators at Warner Brothers and Disney generated quite a bit of alternative reality; in fact, they could almost do whatever they wanted, at costs more or less in line with computer animation of an equivalent quality today, taking inflation into account. What can computer-graphics designers Bob Abel (of Robert Abel and Associates) and John Whitney, Jr. (of Digital Productions), do with computers that suitably funded and inspired animators cannot duplicate with conventional techniques? They certainly give their clients a high-tech image, which by definition requires a computer somewhere. Is it any easier though? Clearly, conventional animation is difficult and labor-intensive, so much so that it is as practical an everyday communication medium for normal humans as are the immense stained-glass rose windows in Gothic cathedrals. Some would say the same for much of computer graphics.

Naturally, we all expect the process of computer animation to get drastically cheaper and easier, so maybe someday it will be as accessible as communicating with embroidery. What the computer really adds to visual communication, however, is the process of discovery. The computer-graphics systems that best aid artists in researching aesthetic spaces are precisely the ones that catch our imagination and hold it. The people Robert Rivlin has chosen to highlight in this book have, all of them, built such systems, the people at Abel and Digital Productions included.

THE QUALITY OF THE MESSAGE

Interaction with three-dimensional moving visuals is not new, of course. Dance, as mentioned before, and sports are quite popular forms of communication that involve elaborate time/space planning and result, often, in abstract beauty. What can we in computer graphics do to compete, using a technology barely 25 years old? How do we maintain viability of the algorithmic image when only three or four computer-animation houses worldwide are producing a return on investment better than bank interest? Rotating logos and sports stadium fly-throughs are a stopgap measure. The answer is reasonably simple: Increase the quality of the message.

Very interactive systems, like ones designed for CAD and even word processing, have, as far as the user is concerned, the highest-quality messages: the user's own ideas. Computer animation rises far above conventional animation when the user is a participant and coauthor of the dialogue. Why have the interactive graphics offered to consumers been limited to Telidon and Ms. PacMan? Nam June Paik, when asked why his video synthesizer couldn't produce trees, said, "Too young." The high-quality graphics systems this book celebrates depend on technology too young to be widely available to creative people in general, and to those who would like to study computer graphics in particular. This stuff is expensive; one of the great unsolved problems of our silicon-technology youth is getting it to our carbon-based youth.

What would it be like if access weren't so tough, if computer graphics weren't so expensive? In 1981, Computer Creations of Indiana produced a television spot complete with a computer-animated logo for a local lumberyard—not exactly the General Motors, Seven-Up or TRW type of client. Suddenly, computer graphics was no longer the exclusive domain of the high priests and their wealthy sponsors. The rose-window folks started looking over their shoulders at the competition coming up fast. I believe this was a turning point of no trivial significance.

A second case in point is an underground piece of computer animation called *Nuke the Duke*, produced in 1983 on a very low budget. It never could have been produced by someone with the military or Hollywood mindset common to the people creating most computer animation. Nor is it cute, cuteness being a safe conceptual space in which to exhibit technical prowess. Charlie Kessler and Jaap Postma's *Nuke the Duke* is an antinuclear war statement. Its soundtrack consists of John Wayne reading an interpretation of the Pledge of Allegiance, backed up by a choral number. The graphics portray a simplistic, video-game-like attitude to death, destruction, and scoring. Indians shoot all sorts of things at the Duke but get him in the end with a nuclear weapon. (Kessler explains that Wayne was doing westerns in the early fifties in Nevada close to nuclear testing sites, a possible cause of his eventual death of cancer.) This is a tape no advertiser would make. It is important because it has a provocative message and because it was made despite its unfundable nature.

Others have transcended the mass of computer graphics and told us that the future is bright. The National Film Board's animated piece entitled *Tony La Peltrie* blew everyone away at SIGGRAPH '85. Jim Blinn's work on "The Mechanical Universe" for PBS actually has taught, not just sold. Artworks by Dean Winkler, Ron Resch, Larry Cuba, Jane Veeder, and Yoichiro Kawaguchi have given us textures of meaning well beyond surface depth. It is not unusual to have artists teach us to see, an educational experience computer scientists and hackers alike should savor. After all, no one expected Gutenberg also to be Shakespeare.

THE SOLVED AND UNSOLVED PROBLEMS

Over the years, the giants of computer graphics have given us their lists of unsolved hardware, software, and algorithmic problems. My list concerns those features of systems that make the difference between real power and formal power, between making computer graphics accessible and keeping them inaccessible.

Among the problems that have been solved is interactive painting into a frame buffer. Systems priced as low as $1500 have good paint programs; most with some type of special effects, such as color-map animation, zooming, and panning. The ability to generate oodles of rotating logos in three dimensions is available to you for moderate to substantial cost in systems such as the Cubicomp and Bosch FGS-4000. Rendering objects the geometry of which is well known is also routine these days. Shadows, surface texturing, highlights, reflections, and refractions are now done by undergraduates on PCs. The communication of the algorithms developed by the 30 or so top researchers has been very good.

We still understand very little about how to describe anything but the simplest objects and motions, however. Paint systems do not describe objects any better than television cameras do. Nor do they allow the user to construct algorithms or procedures—the construction of arbitrary rule sets is thus not available. That level of excitement and creativity is consigned to the system designers who preset the control paths at the factory.

We also have computer languages, which, of course, predate paint-style and icon-based interfaces. Most computer graphics are done as extensions to FORTRAN or C, which allow the user access to the best idea of the twentieth century: creating algorithms with stored programs, looping, and recursion. It turns out that most people who want to use computer graphics to communicate come from backgrounds in which scripts and written proposals precede the start of work, so some form of linguistic interface is arguably natural. Yet we have only the most primitive means of going from script to finished animation, except in cases involving such things as rotating logos and sports stadium fly-throughs. Although it makes sense to argue that a graphical medium should have a graphical specification mechanism, we shouldn't be purists and eliminate access to language. Of course, computer languages are ridiculously hard to create, document, and debug, and they gain acceptance slowly. We get a new one about every five years. Specifying, implementing, and marketing a good script language for manipulating graphical objects, even two-dimensional ones, has yet to be done.

A cheap way of recording computer graphics on videotape is needed. You can now buy a board with sixteen million colors at a resolution of 512 by 512 pixels and put it in a fast PC with a mouse, a hard disk, and several megabytes of memory for $10,000, the price of a three-year-old Buick. You can then generate single frames to your heart's content. Putting them onto videotape, however, destroys conventional video recorders; it costs $75,000 for a tape deck that records frames reliably. (Producing a ten-minute animation takes 30 times 60 times 10 edits. [Video is recorded at 30 frames per second.] Look inside a videotape recorder sometime and think of that mechanism rewinding, starting, editing a frame, and then stopping 18,000 times without flaking out.) Finding $10,000 does not fundamentally change your lifestyle, but finding $100,000 will. A reasonable solution is a cheap, multiframe graphics buffer that can be loaded from any source and flipped through at will. A buffer with 30 frames allows one-second edits, certainly possible with the most inexpensive videotape-editing equipment. Sixty frames or more makes a much nicer visual chunk. Such a buffer would, of course, also allow visualization of the effect before recording, something largely missing from raster graphics. At a few hundred dollars per megabyte, the memory costs are reasonable. You could use the thing as a big RAM disk, you could carry it around and share it with others, and it has no moving parts. Any solution to this problem will revolutionize media production.

Another particularly nasty problem is protection and funding of software. I lump these together because they are completely intertwined. Financial interests will fund products if and only if some form of protection is assured, normally in the form of patents. Software, however, lives at the cutting edge of the legal system, which means that any effective legal remedy involves writing new legislation, an exciting but ridiculously expensive way to go. Copy protection is annoying to honest people and no real challenge to the bad guys. Amusingly enough, software houses have reverted to making systems unusable without 500-page manuals printed on odd-sized paper just to encourage people to buy their own copies to get the documentation. Much of the work on self-teaching systems has been abandoned, ironically. In analyzing possible solutions to this problem, including distribution on execute-only memory or compact laser audio disks, one might want to look at the example set by newspapers, magazines, and television. Putting advertisements on the screen while waiting for compiles, printouts, and so on could pay for the bulk of production costs. Users could copy programs freely, and producers would be interested in updating their software frequently so they could sell more ad space. I can imagine an advertisement like the following: "If you bought our new Model 100 hard disk, you wouldn't have to see this message for so long!" Think of the children's educational software that would be funded by cereal and toy companies. Ads could even be sinister enough to ask questions about the product and its desirable qualities. Give a wrong answer and you get to watch it again. Think of how the industry would develop!

HOW TO PLUG YOURSELF IN

People often ask me how they can learn to produce computer graphics, how they can start programming algorithms, even what system they should buy for their kids. The answer is simple: Go out and buy the most expensive computer-graphics system you can afford to throw out or give away in six months. You can do some form of graphics in BASIC for under $100. You can amaze your friends with a $500 system. After six months, you'll know exactly what you want to get next.

So go do it! It's the best thing you can do for yourself and the computer-graphics industry.

Tom DeFanti,
University of Illinois at Chicago