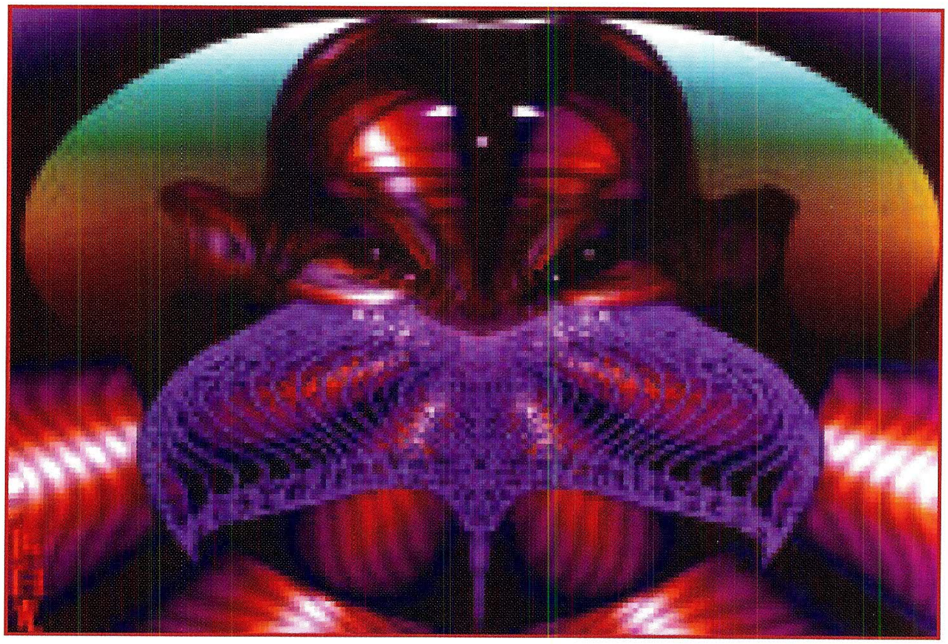
IEEE

Annals

of the History of Computing

Volume 23 Number 2 April–June 2001 http://computer.org



Digital Paint Systems

- Alvy Ray Smith's Overview of Digital Paint Systems
- SuperPaint: An Early Frame Buffer Graphics System
- The Rise and Fall of the Committee on Mathematical Tables and Other Aids to Computation
- The Computer Project at the Naval Ordnance Laboratory

IEEE
COMPUTER
SOCIETY

http://computer.org

IEEE Annals of the History of Computing

Contents

http://computer.org/annals

2 About This Issue/From the Editor's Desk
Tim Bergin, Editor in Chief

Features

4 Digital Paint Systems: An Anecdotal and Historical Overview
Alvy Ray Smith

32 SuperPaint: An Early Frame Buffer Graphics System
Richard Shoup

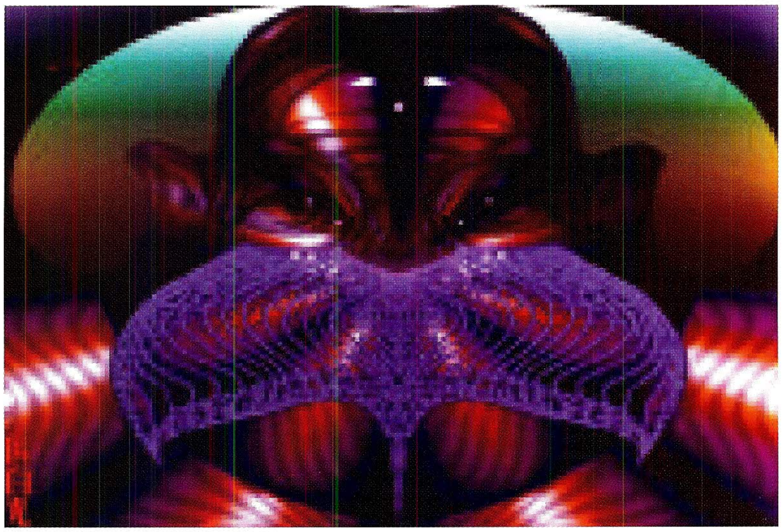
38 The Rise and Fall of the Committee on Mathematical Tables and Other Aids to Computation
David Alan Grier

50 Calvin Mooers, the NOL Computer Project, and John Vincent Atanasoff: An Introduction
Michael R. Williams

51 The Computer Project at the Naval Ordnance Laboratory
Calvin N. Mooers [1919–1994]

Vol. 23, No. 2

April–June 2001



Alvy Ray Smith created *mandarin.tut* by importing an 8-bit painting of King Tut by Paul Xander into 24-bit space using Paint3.

Published by the IEEE Computer Society **ISSN 1058-6180**

IEEE **Annals** of the History of Computing

Departments

68 Events and Sightings
Anne Fitzpatrick and Mary Croarken, editors
Mapping the History of Computing—Software Issues: A report on the International Computer History Conference 2000
Speakers and commentators at the ICHC 2000 Conference
Donald Davies memorial lecture
The John Pinkerton inaugural lecture
Talks at the Computer History Museum
Web Extras

74 Reviews
Peggy Aldrich Kidwell, editor
Systems, Experts, and Computers: The Systems Approach in Management and Engineering, World War II and After
The First Computers—History and Architectures
Robo Sapiens: Evolution of a New Species

78 Anecdotes
David Alan Grier
The First Breach of Computer Security?

80 Biographies
J.A.N. Lee, editor
Donald Davies
Leo Fantl
Homer (Barney) Oldfield
John McPherson
Claude Elwood Shannon
Herbert A. Simon

88 Think Piece
James Cortada
Is the *Annals* at Risk of Falling Behind?

Computer Society Information, p. 31

Editor in Chief
Thomas J. (Tim) Bergin

Associate Editor in Chief, Departments
David A. Grier

Associate Editor in Chief, Special Projects
Paul Ceruzzi

Editorial Board
William Aspray
Werner Buchholz
Martin Campbell-Kelly
Alan Clements
James W. Cortada
Mary Croarken
Anne Fitzpatrick
Bernard A. Galler
Denise Whitson Gürer
Ulf Hashagen
Peggy Aldrich Kidwell
J.A.N. Lee
Michael S. Mahoney
Arthur L. Norberg
Brian Randell
Raul Rojas
Robert F. Rosin
Keith Smillie
Dag Spicer
James E. Tomayko
Eric A. Weiss
Michael R. Williams

Publisher: Angela Burgess
Assistant Publisher: Dick Price
Group Managing Editor: Robin Baldwin
Staff Editor: Cheryl Baltes
Assistant Editor: Tammi Titsworth
Magazine Assistant: Alkenia Winston

Magazine Operations Committee
Sorel Reisman (chair)
James H. Aylor, Jean Bacon
Thomas J. (Tim) Bergin
Wushow Chou, William I. Grosky
Steve McConnell, Daniel E. O'Leary
Ken Sakamura,Munindar P. Singh
Francis Sullivan, James J. Thomas
Yervant Zorian

Publications Board
Rangachar Kasturi (vice president)
Sorel Reisman (MOC chair)
Jake Aggarwal (TOC chair)
Mark Christensen (POC chair)
Laxminarayan Bhuyan, Lori Clarke, Lee Giles
Gabriella Sanniti di Baja, Sallie V. Sheppard

Circulation: IEEE Annals of the History of Computing (ISSN 1058-6180) is published quarterly by the IEEE Computer Society, 10662 Los Vaqueros Circle, P.O. Box 3014, Los Alamitos, CA 90720-1314, USA; phone +1 714 821 8380. IEEE Computer Society Headquarters, 1730 Massachusetts Ave. NW, Washington, D.C. 20036-1992, USA; IEEE Headquarters, 345 East 47th St., New York, NY 10017, USA. Subscription rates: IEEE Computer Society members get the lowest rates and choice of media option—$28/22/36 US print/electronic/combination. Go to http://computer.org/subscribe to order and for more information on other subscription prices. Back issues: members, $10; nonmembers, $20. This magazine is also available in microfiche form.

Postmaster. Send undelivered copies and address to *IEEE Annals of the History of Computing*, IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08855, USA. Periodicals postage paid at New York, NY, and at additional mailing offices. Canadian GST Registration No. 125634188. Canada Post Publications Mail Agreement Number 0487880. Printed in the United States.

Copyright and reprint permission: Copyright © 2001 by the Institute of Electrical and Electronic Engineers, Inc. All rights reserved. Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of US copyright law for private use of patrons articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, USA. For other copying, reprint, or republication permission, write to Permission Editor, *IEEE Annals of History of Computing*, 10662 Los Vaqueros Circle, P.O. Box 3014, Los Alamitos, CA 90720-1314, USA.

Editorial: Unless otherwise stated, bylined articles and reviews reflect the author's opinion. Inclusion in *IEEE Annals of the History of Computing* does not necessarily constitute endorsement by the IEEE Computer Society. All submissions are subject to editing for style, clarity, and space considerations.

About This Issue

Tim Bergin
Editor in Chief

This issue introduces two products of a cooperative effort between the *Annals* and The Computer Museum History Center at Moffett Field, California. As part of a Museum-sponsored lecture series, in January 2000 Alvy Ray Smith and Richard Shoup gave a joint talk entitled "Recollections of Early Paint Systems." Through the efforts of Dag Spicer, an *Annals* Editorial Board member, I invited these pioneers to re-work their lectures into articles for the *Annals*.

In "Digital Paint Systems: An Anecdotal and Historical Overview," Alvy Ray Smith takes us on a tour of digital paint programs from the late 1960s to the early 1980s.

"SuperPaint: An Early Frame Buffer Graphics System" by Richard Shoup discusses his work at the Xerox Palo Alto Research Center (PARC) on one of the first pixel-based frame buffer systems. This system later became known as SuperPaint.

I truly appreciate Alvy Ray's and Dick's efforts in turning their lecture at the Museum into papers for the *Annals* and hope that their efforts will inspire others to do the same.

In "The Storm before the Calm: The Rise and Fall of the Committee on Mathematical Tables and Other Aids to Computation," David Grier reminds us that computing was an important scientific and commercial process before modern computers were developed.

The final article in this issue has a history of its own. In April 1993, the SIGPLAN History of Programming Languages Conference (HOPL-II) took place in Cambridge, Massachusetts. At one of the sessions, I introduced myself during a break to an older gentleman. He said he was Calvin Mooers and that he had worked at the Naval Ordnance Laboratory. When the session ended I asked if he had known John V. Atanasoff. He said that he knew Atanasoff well and that he disagreed with some of Mollenhoff's conclusions.

I urged him to put his thoughts in writing and send them to J.A.N. Lee and Jim Tomayko (then editor in chief and Anecdotes editor, respectively). Mooers' thoughts appeared as "Atanasoff at the Naval Ordnance Laboratory" in Anecdotes, *Annals* vol. 15, no. 2, April–June 1993, pp. 54–55.

You can imagine my surprise when, seven years later, I received a letter from Helen Solorzano, Calvin Mooers' daughter, asking if the *Annals* would be interested in printing some materials from her father's notebooks. I told her about my meeting her father in Cambridge in 1993 and his visit to Washington, D.C. "The Computer Project at the Naval Ordnance Laboratory" was compiled by Helen, a freelance writer, from her father's notebooks. We have also included an introduction in which Helen relates how she and her sister found the notebooks, the editing performed on the material, and their efforts to promote the use of the TRAC language.

Because some of Mooers' comments are unflattering and neither Mooers nor Atanasoff are able to comment, I followed a reviewer's advice suggesting that a senior historian write a preface to the material. Consequently, "Calvin Mooers, the NOL Computer Project, and John Atanasoff: An Introduction" by Michael Williams is an attempt to put the material into an historical context. These materials are now in the Charles Babbage Institute where readers can read for themselves and form their own opinions.

From the Editor's Desk

As I finish my first year as editor in chief, I'd like to review what we've accomplished,look ahead to the future, and acknowledge the people who made this year possible.

First, we had a very successful editorial year in 2000, beginning with James W. Birkenstock's memoir "Pioneering: On the Frontier of Electronic Data Processing, a Personal Memoir" and ending with an issue dedicated to our colleague Allan Bromley. Each issue contained numerous articles, which enlightened us and made our lives more fun.

These issues would not have been possible without the authors' efforts nor without the reviewers who critiqued their efforts and suggested improvement. Special thanks go to David Grier and J.A.N. Lee, both of whom had two articles published in 2000. I also want to thank the editors of our departments. Their efforts to solicit materials makes the departments mandatory reading.

I would be remiss if I did not acknowledge those who stepped forward to assist me as I struggled with these new responsibilities. When I went through the files that Mike Williams sent me, I found a paper by Allan Bromley from 1996. When I sent Allan a note inquiring about the status of the revision, Allan said that his illness would prevent his working on a revision. Recognizing the importance of Bromley's work, I asked for help in revising the paper. Mike Williams volunteered to edit the manuscript. In addition, Mike worked with Martin Campbell-Kelly and Sir Maurice Wilkes to bring Allan's scholarship to our readers.

Other help came from William Aspray and Jim

1058-6180/00/$10.00 © 2001 IEEE

Digital Paint Systems: An Anecdotal and Historical Overview

Alvy Ray Smith

The history of digital paint systems derives from many things—chance meetings, coincidences and boredom, artistic license, brilliant researchers, a wealthy benefactor, and, of course, lawsuits. Alvy Ray Smith tells the fascinating story—facts first, then anecdotes—in his own words.

This article is based on a talk presented in January 2000 at an evening hosted by the Computer History Museum on Moffatt Field, near Palo Alto, California. I shared the floor with longtime colleague Richard G. "Dick" Shoup who figures highly in what follows. It is also based on a document I submitted to the Academy of Motion Picture Arts and Sciences (AMPAS) in 1997 in answer to a call from them for information about early paint programs and their contribution to the film business.[1]

The time frame dates from the late 1960s to the early 1980s, from the beginnings of the technology of digital painting up to the first consumer products that implemented it. I include only a little information about major developments in the later 1980s. Two surveys that cover this later period fairly well—when the emergence of the personal computer completely changed the software universe—were both published in the magazine *Computer Graphics World*.[2,3] My emphasis then and now, of course, is on those systems I knew firsthand. A description of the legal battles about paint patents in the 1990s adds a certain modern perspective; however, these cases center on systems from the earlier period.

After definitions (see also the "Digital Paint Glossary" sidebar), I present a simple timeline of programs and systems in Table 1 (on p. 6), and then attempt a weighting and a "genealogy" of these. I particularly emphasize those painting systems that have directly affected the movie industry, important to AMPAS, of course.

Having cast the systems into perspective in this rather dry way, I follow with brief histories and then a full anecdotal history of several of them, tying in the strong personalities and strange coincidences of this exciting time. I do not cover the many nonpaint developments proceeding simultaneously—for example, digital 2D (two-dimensional) and 3D modeling and animation, film recording, video editing, and audio synthesis. An excellent rendering of my time with Dick Shoup (sounds like "shout," not like "hoop") in the early days at Xerox PARC (Palo Alto Research Center) can be found in the recent book *Dealers of Lightning: Xerox PARC and the Dawn of the Computer Age*.[4] For other PARC references, see Lavendel,[5] Pake,[6] Perry,[7] and Smith.[8]

Definitions

A digital paint program and a digital paint system are distinguished by their functions. A digital paint *program* essentially does no more than simulate painting of a brush on a canvas. A digital paint *system* does much more, using the "simulation of painting" as a familiar metaphor to seduce artists into the new, perhaps forbidding, digital domain. Of course, they are both programs, but the term "system" will imply many more features; it will be more "complete." In fact, a system might even use several well-integrated programs.

The world of computer-assisted picture making can be divided cleanly and simply into two distinct worlds: geometry-based graphics and pixel-based graphics (also known as imaging), depending on how the original data is stored.[9] Standard computer-generated imagery—as in Pixar's *Toy Story*, for example—is geometry based. The digital paint system captures the pixel-based half of graphics.[10]

A digital paint system is a set of tools for dealing with pixels. Certainly, a digital paint program is one of these tools. I widen the definition of a digital paint program—from simply a simulation of traditional painting—to include any image processing (that is, pixel processing) function applied to pixels under *hand control*—for example, with a mouse or a stylus on a tablet. To support this encompassing definition

1058-6180/01/$10.00 © 2001 IEEE

Digital Paint Glossary

Antialiasing

Representing a continuous entity with a set of discrete samples results, when done incorrectly, in annoying visual artifacts called *aliasing*. For example, a straight line, if naïvely represented by the nearest pixels in a rectangular array of them, will look like a jagged staircase. In fact, this aliasing is called *jaggies* in computer graphics jargon. Antialiasing, of course, removes these sampling artifacts. The idea used in all techniques is to spread out a line or edge and round it off so that the sampling grid has something partial to sample rather than a simple binary hit or miss. In sampling theory terms, its high frequencies are removed, leaving only those consistent with the frequency of the sampling grid. For a black line on a white background, one simple solution is to lay down a ramp of partial grays along each stair step, varying the black of the line smoothly to the white of the background. From a normal viewing distance, this will look like a smooth straight line.

Cellular automata (CA)

Think of an infinite chessboard where each square is a digital computer. Suppose that each of these computers, a *cell*, can take input from only its four nearest-neighbor cells, and can give only them its output. Assume that each cell computes one step at the same time as all the other cells compute one step. The theory of CA explores the capabilities of such highly uniform, but highly parallel, metacomputers. John von Neumann, who originated the theory with Stanislau Ulam, proved that there was a group of cells in one particular cellular automaton that could reproduce itself and also compute any computable function, a nontrivial (because it's computation-universal) self-reproducing machine. The "Game of Life" (M. Gardner, *Wheels, Life and Other Mathematical Amusements,* W.H. Freeman, New York, 1983) is a popular example of a computation-universal CA.

Color map

A nifty trick for creating lots of colors when you have only a little memory is the color map, also called a color lookup table. Suppose you have only 8 bits per pixel, instead of the desirable 24, for color. Then there can be only $2^8 = 256$ values per pixel. Rather than hardwiring these 256 values to fixed colors, they are assigned, via a color map, to any set of 256 colors from a possible choice of over 16 million.

The color map is a table with 256 entries, with each entry being a triple of red, green, and blue (RGB) values, each with 8 bits. When one of the values is read from a pixel, it is looked up in the color map table. The corresponding triple of RGB values, the full color for that value, is sent on to the display device for that pixel. But each RGB triple can have one of $2^{3*8} = 16{,}777{,}216$ possible colors. Although any one picture can have only 256 colors in it, those colors can be completely different from the 256 colors in another picture. Furthermore, since only 256 table entries have to be changed to completely change the color of a picture, this can be done extremely fast, at interactive speeds. This is important, for example, on the Web.

The drawback of color-mapped pictures is that there is no notion of mixing colors available. Color A mixed with color B is just some other color in the color map and most probably this third color has nothing to do with A and B; it is not their mixture, in other words. For mixture, you need all 16 megacolors. Think of a color map as 1D (one-dimensional) color and full RGB as 3D color to understand the difference.

Compositing

To composite two images is to place one over the other to produce a third. For example, in a frame of animation, a foreground character is composited over a background painting. Where the foreground object is transparent, the background shows through in the result. Where opaque, only the foreground object shows in the result.

In the digital equivalent of this process, one must be careful to antialias the edges of the foreground object to avoid jaggies in the composite image. This is accomplished by partial mixing of the foreground object with the underlying portion of the background scene. In the case of the black line, the gray ramp along each stair step would weight the amount of mixing with the background: Black is foreground only; white is background only; and gray is a mixture of the two, depending on the value of the gray.

Frame buffer

A frame buffer is nothing more than a piece of computer memory with a means for viewing what it holds. Originally it buffered an output device from a computer. The computer would write the next frame of a video, say, into the buffer—hence, frame buffer. Then it would begin to compute the next frame while independently the display device read the current frame out of the frame buffer and displayed it.

In 1975, a full-color frame buffer for television resolution frames occupied three large racks of equipment, about three kitchen refrigerators in size, and cost over a million 1999 dollars. Today, this same device is called a graphics card and fits in one small slot in a personal computer. It costs on the order of ten dollars, and every PC has one.

---

of digital paint system, we should consider modern consumer products that are heirs to the technology discussed here: Adobe Photoshop, Microsoft PhotoDraw 2000, the Corel Draw suite, and so forth. Suffice to say that "system" implies much more capability than "program."

A word about bit depth: There are five bit depths in the programs and systems discussed

Table 1. Timeline for color paint programs and systems.

Time frame	Event
1969–1970	Joan Miller implements a crude paint program on a 3-bit frame buffer at Bells Labs.[12]
1969–1970	W.J. Kubitz and W.J. Poppelbaum implement the "tricolor cartograph" with eight fixed colors, a crude (and possibly partially or wholly analog) paint program.[13]
1972–1973	Dick Shoup creates SuperPaint, the first complete 8-bit paint system, including hardware and software, at Xerox Palo Alto Research Center (PARC).[14-16]
1974	Jim Blinn writes 8-bit paint program Crayon[17] for the new Evans & Sutherland (E&S) frame buffer[18] at the University of Utah in December 1974.[19-21]
1975–1976	Garland Stern writes 8-bit paint program for the E&S frame buffer at the University of Utah.[19] He brings it with him to the New York Institute of Technology (NYIT) in 1975.[22,23]
1975–1976	Alvy Ray Smith creates 8-bit paint system Paint at the NYIT, also for an E&S frame buffer, later for the Genisco frame buffer. Paint is sold to Ampex in late 1976.[19,24]
1976	Jules Bloomenthal creates an 8-bit paint program at the University of Utah, E&S frame buffer, after seeing Blinn's and Stern's programs.[25]
1976	Marc Levoy programs an 8-bit paint system for the E&S frame buffer at Cornell University.[19,26]
1976	Alvy Ray Smith creates an 8-bit paint system BigPaint, the first for pictures larger than video resolution, on E&S and Genisco frame buffers.[19,27]
1976	Massachusetts Institute of Technology students implement an 8-bit paint program on MIT's own frame buffer.[19,27,28]
1977	Alvy Ray Smith implements first 24-bit red, green, blue (RGB) paint system Paint3 at the NYIT, for three E&S or Genisco frame buffers in parallel.[19]
1977	Alvy Ray Smith and Ed Catmull invent the concept of the integral alpha channel, a fourth channel, in addition to RGB, carrying transparency information at each pixel.[29] Alvy calls it the "alpha" channel then.[30]
1977–1978	Jim Blinn adapts Bloomenthal's paint program to his needs at the Jet Propulsion Laboratory (JPL) in Pasadena. In particular, in 1977 he adds airbrushing that is later used to generate texture maps for his Voyager flyby simulation movies in 1978.[21]
1978	Alvy Ray Smith publishes the first documentation of Paint, BigPaint, and Paint3, plus history of digital painting up until then, as tutorial notes for ACM's Siggraph [Association for Computing Machinery's Special Interest Group on Computer Graphics and Interactive Techniques].[19] These notes plus others[31] are used for several years for successive Siggraph tutorials.
1978	Marc Levoy implements a specialized 8-bit program for painting (opaquing) animation cels, Opaque. It preserves soft edges, using a color map technique,[32] and is used in Hanna-Barbera animation production.
1978–1979	Alvy Ray Smith creates BigPaint3, the 24-bit version of BigPaint, and hence the first 24-bit paint program for higher-than-video resolution.
1978–1979	Marc Levoy implements a 24-bit paint system Paint at Cornell, on a 24-bit Grinnell frame buffer.
1978–1979	Ephraim Cohen implements ept, an 8-bit or 24-bit paint program (it could do either), at the NYIT, for E&S and Genisco frame buffers.
1979	Ed Emshwiller creates *Sunstone* video using Paint (with assistance from Alvy Ray Smith, Lance Williams, and Garland Stern) at the NYIT. This video is now part of many museum collections, notably the Museum of Modern Art in New York.

here: 1-bit, 3-bit, 8-bit, 24-bit, and 32-bit. These correspond, respectively, to 2 colors (black and white), 8 colors, 256 colors, 16.7 million colors, and 16.7 million colors plus 256 levels of transparency. I give the 1-bit and 3-bit systems short shrift. The 8-bit, or 256-color, systems made digital painting a real tool in video. The 24-bit and 32-bit paint systems are required for film use. The difference between 24-bit and 32-bit systems is the availability of an extra channel—the "alpha channel"—carrying transparency information, for all images at all times.[11]

In both kinds of systems, transparency effects such as airbrushing and soft-edged compositing are possible, but only 32-bit systems as a matter of course always generate an alpha channel with every image. In other words, the alpha channel is painted when the color channels are painted, not as a separate task. Systems with even higher bit depth—for example, 64 bits—are a modern development due to cheap memory and will not be further discussed.

Genealogy of digital paint systems

The principal stream, or mainstream, so far as movie production is concerned, consists of

Time frame	Event
1980	Dick Shoup and Damon Rarey start the company Aurora Systems; they build and sell the Aurora/100 digital videographic system, an 8-bit hardware and software paint system.
1979–1980	Marc Levoy, Bruce Wallace, and Chris Odgers implement a digital imaging system for Hanna-Barbera based on the 24-bit Grinnell frame buffer. The system is still in use at Wang Studios in Taiwan, but the 24-bit paint program never made it into production use.[33]
1979–1980	Thomas Porter, Rodney Stock, Larry Evans, Ken Turkowski, and Junaid Sheikh create Ampex Video Art (AVA), a commercial 8-bit paint program and hardware.[34] It took special care not to create aliased edges, using special color-map techniques (compare to Levoy's Opaque, above).
1980–1981	Jeff Burton adapts Ephraim Cohen's program to implement IMAGES (Image Manipulation and Generation Electronic System),[35] an 8-bit or 24-bit (depending on configuration) commercial paint program product for CGL Inc. (named for the Computer Graphics laboratory at NYIT), a subsidiary of the NYIT.
1980s	Richard Taylor and others implement Paintbox, a 24-bit commercial video paint product for the English company Quantel, the first system to employ special-purpose hardware for acceleration of digital painting. They later extend the resolution in the product Graphic Paintbox for use in print and film.[3,35]
1981	Gene Miller writes a paint system at the New York company Digital Effects, founded by Judson Rosebush, for the ad agency J.W. Thompson. It ran on the Lexidata frame buffer and output to film.
1981–1982	Thomas Porter implements first complete 32-bit (RGBA) paint system Paint at Lucasfilm, using a 32-bit Ikonas frame buffer. It is used in *Star Trek II: The Wrath of Khan*[36]—the first use of digital paint systems in a film released to commercial theaters.[37-39]
1985–1986	Mark Leather implements LayerPaint, a 32-bit paint program, on the Pixar Image Computer at the Computer Division of Lucasfilm. He is awarded a technical Academy Award for its use in "wire removal" in 1994.
1985–1986	The Knoll brothers, Thomas and John,[40] working at Lucasfilm, adapt the Computer Division imaging functions, including painting, to create Photoshop, a 24-bit commercial imaging product (originally for the Apple Macintosh, which was not 24-bit-capable until about 1989–1990), later purchased by Adobe.[41]
1986–1988	Tom Hahn (formerly with Dick Shoup at Aurora), Michael Shantzis, and Peter Nye implement CAPS (Computer Animation Production System), a digital animation system including cel painting, on the Pixar Image Computer at Pixar on a contract with Disney. They (and Disney personnel) are awarded a technical Academy Award for its use in animated feature films in 1992.
1989	Alvy Ray Smith is star witness for the defense in the *Quantel* v. *Spaceward* paint (airbrushing and matting) patent infringement trial, London, February 1989. Spaceward loses despite his testimony and that of colleagues Jim Blinn and Lance Williams, so Quantel's English patents are upheld.
1990–1991	Alvy Ray Smith writes Composer, a 32-bit imaging system, with digital painting, at Pixar. Technology is spun off into start-up company Altamira Software, which produces Altamira Composer. This is merged into Microsoft in 1994 where the product was repackaged and improved as Microsoft Image Composer, a 32-bit product with painting. A closely related sister product (based on the same principles) for photos is Microsoft's Picture It! The most recent incarnation of the technology is in Microsoft's PhotoDraw 2000, which integrates geometry with imaging in one product. Version 2 of this product shipped in late 1999.
1997	Alvy Ray Smith and Marc Levoy are star witnesses for the defense, *Quantel* v. *Adobe* paint (airbrushing and matting) patent infringement trial, Wilmington, Delaware, September 1997.[42] All five of Quantel's US patents in the case are invalidated. Adobe and Photoshop are exonerated. Also testifying for the defense were David Em, Christine Barton, and others.

the following seminal events:

- Dick Shoup's SuperPaint (1972–1973)
- Alvy Ray Smith's Paint and Paint3 (1975–1977)
- Tom Porter's Paint (1981–1982)
- Mark Leather's LayerPaint (1985–1986)
- The Knoll brothers' Photoshop (1985–1986)

To establish context—before discussing the many side streams that influenced movie production—I mention the developments that had little or no effect on this mainstream.

The Miller and Kubitz and Poppelbaum programs have not influenced the programs or systems just cited. They both had only eight colors and were not systems but merely programs, using the terminology introduced earlier.

I have not included any paint programs that painted only in black or white (1 bit) since they had no influence on the mainstream, either. For example, I recall one such program at Xerox PARC while I was employed there in 1974.

The MIT system was not influential, at least not on the systems with which I am familiar.

Jim Blinn's program had no direct effect on me other than the influence, if any, it had on the program Garland Stern brought to the New York Institute of Technology (NYIT) in 1975.[43] The Stern program goaded me, from dissatisfaction, into writing my first system. For example, it had a strange stroking technique that I knew not to use: A stroke was started, and stopped, by clicking the tablet stylus once (down and up). This had the effect of feeling good while stroking, there being no pressure on the fingertips via the stylus, but awkward while starting and stopping—causing jerks in the painted stroke. The stroking technique I used—and used by all subsequent programs—was the one I learned from Dick Shoup's system: Push down the stylus to start a stroke, drag it (under pressure) to paint, then release pressure to stop the stroke.

Jim Blinn's influence took a separate path. In 1977 he took the Jules Bloomenthal paint program from Utah to the Jet Propulsion Laboratory. Then, at the request of Evans & Sutherland, he took all his frame buffer software and installed it at the National Institutes of Health (NIH) in Washington, D.C. (on an E&S frame buffer, of course). This was in 1977. The principal system engineer at NIH with whom he worked was Tom Porter, who was probably seeing his first paint program. Jim continued at JPL for many years. He provided a home there for early computer artist David Em. The proximity to Hollywood of JPL in Pasadena made Jim's lab the first place where many film people were exposed to digital imaging technology.[21]

The Quantel stream had no influence on the mainstream. It is not clear if Quantel was influenced by the NYIT developments. Quantel people made trips to NYIT, attended tutorials by NYIT personnel, acknowledged having my paper,[19] and had an abortive deal with NYIT but legal precedence matters have obfuscated this part of the history. (The legal battles will be discussed later.)

I don't know of any connection between the Digital Effects paint program and the mainstream. Founder Judson Rosebush does figure in the history, however.[44]

One definite side stream to the main paint system lineage is of course the Aurora Imaging system Aurora/100. Later, Tom Hahn left Aurora to join Lucasfilm and write the CAPS system there under contract to Disney (a deal and a hire I negotiated).

Another important side stream is that started by Marc Levoy at Cornell, which led to Hanna-Barbera. Marc and his colleagues were some of the first and frequent visitors to our lab at NYIT. I believe it is accurate to say they were inspired by what they saw there, although their stream was essentially a separate development after that with many creative contributions.

Another stream, which comes back into the mainstream, is the one that began with the sale of my Paint program to Ampex and its influence on Ampex Video Art and particularly on Tom Porter.[45] As the new director of Computer Graphics Research at Lucasfilm, one of my first hires was Tom Porter to implement our Lucasfilm paint program, knowing that he understood my style. Mark Leather, later hired from Ampex as a hardware engineer at Pixar, surprised us by writing the first paint program, LayerPaint, on the new Pixar Image Computer (that he helped design) and that Lucasfilm used for movie production. He had worked on the hardware for AVA at Ampex.

My colleague Ephraim Cohen took a lot of the NYIT ideas, including those from my Paint and Paint3 (see Figure 1) and brewed them a different way[46] to produce ept, which begat Images and later (early 1990s) a system, Cricket Paint, that Ephraim wrote under arrangement with Computer Associates. In particular, he noted that painting was just a hand-driven form of image composition and used this as an organizing principle.[47]

The Photoshop connection is not well known. I recall that my Lucasfilm colleague David DiFrancesco kept urging me to visit another Lucasfilm building to see what the Knoll brothers were doing with "our stuff," putting it onto Apple's Macintosh computer. He told me it was impressive, but I couldn't be bothered with tiny little machines at the time—a mistake I corrected many years later by adopting the Windows platform. The Knoll brothers shopped their creation for a while, finally getting some support from a start-up company called BarneyScan. Then John Warnock saw it and arranged to have Adobe market the system, grandfathering (presumably) BarneyScan. The dates I give in Table 1 (1985–1986) are approximate, based on the fact that we, Ed Catmull and I, spun out Pixar from Lucasfilm in 1986.

Brief factual histories

Before relating the anecdotal history, I sketch the background of the paint programs involved.

Dick Shoup: SuperPaint

Dick Shoup's SuperPaint[48] was a revolutionary program—simple and intuitive, the parent of all modern paint programs. It had 256 colors selectable from 16.7 million, a palette, a color map, video in and video out, a tablet and stylus,

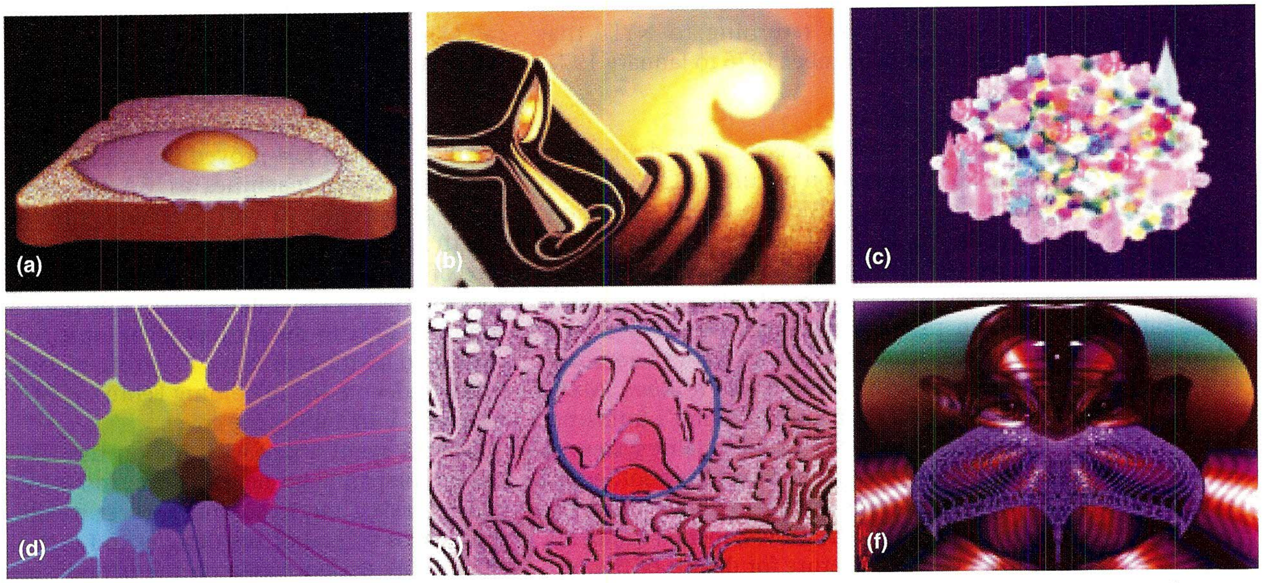


Figure 1. These six pieces, all created by the author, represent several developmental stages in the paint program series at the New York Institute of Technology. (a) I created *egg.on.toast* by parts with Paint, making it the first high-resolution image at NYIT and inspiring the creation of BigPaint, used to create the tubular (b) *darth.vader* in 1978. (c) The colored blob, is the first test pattern for Paint3 in 1977, followed by (d) another early test called *colorweb*. Both these images figured in the *Quantel* vs. *Spaceward* patent trial. (e) *bleu.drop* was created from the 1975 Paint piece *bleu*, by adding a dropped shadow in 24-bit space to the paint strokes. (f) To create *mandarin.tut*, I imported an 8-bit painting of King Tut by Paul Xander into 24-bit space and modified it highly, including the addition of a moustache using Paint3. The last four pieces are at video resolution, and the first two at twice that.

variable paintbrush size, animation, video magnification, image transformations, image file input and output—all the basics of a modern paint program. The program eventually earned Dick (and Xerox) an Emmy award in 1983.

I began using SuperPaint in February 1974. My small contribution to it was the RGB to HSV (hue, saturation, and value) also known as HSB (hue, saturation, and brightness) color transform, for more intuitive color choice by artists. I used the system to create my first digital animation videotape, *Vidbits*, in 1974.

Alvy Ray Smith: Paint, Paint3, BigPaint, and BigPaint3

I've clearly described Paint (and its high-resolution extension, BigPaint) elsewhere.[19] Of more pertinence here is the full-color, 24-bit version, Paint3, briefly described in Appendix B of that documentation.[19] It was the first paint system to have 16.7 million colors. This permitted airbrushing and full compositing of any image over any other, and these were both implemented. In fact, using the notion introduced in Paint—that any image can be used as a "brush" to paint on any other image—Paint3 allowed brushes of any shape, including the "shape" of its transparency. Airbrushing fell out of this observation by easy default: Simply choose a brush that is opaque in the center and drops off gradually to transparent around its edges. BigPaint3 was the high-resolution extension of Paint3.

Paint3 featured 24 bits of color, airbrushing, tablet and stylus control, variable paintbrush size and shape (any shape, with any pixel-by-pixel opacity variation), image save and restore (of either 8-bit or 24-bit images), a disappearing palette (it disappeared after color selection), color selection from anywhere on the screen (not necessarily the palette), video magnification, palette selection (for convenience only) from palettes of arbitrary colors, tint and value adjustment of colors, color mixing (or smearing, as I called it), 24-bit color fill, and other functions. There was a button that allowed any other 24-bit program in my system to be run from within the paint program (hence my definition of system). This included a full-featured image restoration program that handled (after 1977) alpha channels (called Getpa) and an antialiased geometric rendering program (by Malcolm Blanchard) called Sketch (extended by me to 24 bits as Sketch3).

I conceived of doing Paint3 while lying bored in a motel room in Redwood City, California, after completing installation of Paint at Ampex in December 1976. I wrote it immediately upon returning to NYIT that month, from December 1976 to January 1977.

Tom Porter: Paint

The Lucasfilm Paint system was designed to be suitable for motion picture use, so Tom paid a great deal of attention to image resolution, color fidelity, and antialiasing problems.[37-39] He began the system development during 1981 and completed it in January 1982, in time to be used in production of the "Genesis Effect" sequence (which I directed) for the Paramount movie *Star Trek II: The Wrath of Khan.* Matte painter Chris Evans of Lucasfilm's Industrial Light & Magic (ILM) used the paint system in early 1982 to create the ground and soft-edged clouds that were texture mapped onto the receding planet. The movie opened in June of 1982. See the literature for details of the production, including reference to Tom's system.[36]

This was the first use of digital paint systems for movie production at Lucasfilm. I believe it was also their first use in theatrical release motion pictures. Paint was also used in the production of the special effects for "the stained-glass man" in the 1985 Amblin Productions' film *The Young Sherlock Holmes.*

Major features of the Lucasfilm Paint system were arbitrary resolution, creation and manipulation of the alpha channel with every stroke (that is, it was a full 32-bit system—the first, I believe—with 16.7 million colors and 256 levels of transparency), and careful attention to antialiasing and compositing. Tom Porter reports the existence of a March 1981 Lucasfilm technical memo about the proposed architecture.[49]

Anecdotal history of PARC and NYIT Paint

Facts are facts, but it's the details—the stories and the characters—that bring these histories to life.

Xerox PARC, Dick Shoup, and SuperPaint

A 1970 *Scientific American* issue made a splash when Martin Gardner devoted his famous column to John Conway's "Game of Life."[50] "Life" brought Dick Shoup and me together—before paint programs—while we both were involved in our early academic careers. I was a specialist in cellular automata (CA) theory, having written my PhD dissertation on it at Stanford University in 1969, and pursuing my first professorship at New York University while I did CA research.

Having been a Gardner devotee since boyhood, I saw the column as soon as it was published and leapt, on reading it, to the phone to call Martin in upstate New York. Excited, I told him that the "Game of Life" was just a special case of CA, that John von Neumann and Stanislau Ulam had been involved in their creation, that I had just proved the existence of self-reproducing machines as CA (in eight pages as opposed to von Neumann's book-length proof), and that I had settled a conjecture of Conway reported by Gardner in the column. He was very interested because the CA column had been the most popular topic ever published by the magazine, inspiring its publisher to devote the upcoming February 1971 issue to CA, including the cover.

Martin spent a day with me on the NYU campus preparing his next column, and subsequently asked me to submit a cover design, along with several of his designs and one of Ulam's. My design, based on the palindrome "too hot to hoot" (that I learned from a *New York Times* crossword puzzle), was a stylized proof of palindrome recognition by CA I had recently published. Luckily for me, the publisher of *Scientific American* happened to be a palindrome aficionado! My cover design was chosen, I gained my first inklings of fame, and the event led directly to my meeting Dick Shoup.

Having thus become a "known" entity as a result of my cover design, I was asked to chair a panel on parallel logic (CA is a special case) at the 1971 International Computer Society Conference in Boston.[51] I needed someone with hardware expertise to round out my panel. In asking around, all pointers led to Dick Shoup, just finishing his PhD dissertation at Carnegie Mellon University on a related topic. He accepted my offer. On meeting in Boston, we discovered that we liked each other enormously, but it took a broken leg to force the next connection in our careers.

While schussing down an icy slope in New Hampshire, my cap slipped over my eyes just long enough that I failed to spot the skier barrel-rolling toward me, completely out of control. The stumblebum skied away but I didn't. A nasty spiral fracture of my right femur put me away—a helpless, immobile invalid—for three months in a full-body cost, nipples to toes.

This time turned out to be one of the most wonderful of my life. Empowered by the vast mental capacity available when the brain doesn't have to move the body about, I rethought my whole life and came to the conclusion that I was on the wrong track, not using my artistic talent, not enjoying the fact that only a few dozen people in the world could talk CA with me. I made the decision to leave academia—"drop out" was the term then—go to Berkeley, and wait for something good to happen. Looking back, I am astonished at my certainty that something would—and it did.

Just about the time I ran out of money in Berkeley, the Munich publisher Rogner & Bernhard Gmbh asked me to write the introduction to the German edition of von Neumann's *Theory of Self-Reproducing Automata,*[52] containing the proof that I had shortened.[53] I decided to make this my swan song to academia, a survey of the entire field of CA (which I was then calling *polyautomata*). I did this off the top of my head at the time, except for the many bibliographic details. The Berkeley library wasn't sufficient for this task, but I knew that the Stanford library was. So I called up my buddy Dick Shoup in Palo Alto, across San Francisco Bay, to get a room for the night near Stanford.

Dick had come to California in 1970 and joined the Berkeley Computer Corporation. BCC received early funding from the ARPA (Advanced Research Projects Agency, US Department of Defense) group that funded computer projects, the Information Processing Techniques Office. Bob Taylor, as former head of IPTO, had been instrumental in the BCC funding. He had just joined a new place called Xerox PARC. When BCC ran into trouble, Taylor moved quickly to scoop up several of its key folks: Butler Lampson, Peter Deutsch, Charles Simonyi, Jim Mitchell, Chuck Thacker—and Dick Shoup. So, not nearly so well known as the first personal computer, the first windowing system, and the mouse, PARC also sheltered the first computer painting system, SuperPaint, midwifed by Shoup in 1972–1973.

I am embarrassed to admit that I didn't really "get" the notion of a paint program at first. In a series of earlier phone calls from PARC to NYU during the early 1970s, Dick had tried to explain to me his idea for a computer to help artists, knowing that I was a painter. (I had painted in oils and acrylics for years, even exhibiting at the Stanford Coffee House once.) But I, like everybody else at the time, could not grasp the idea from a verbal description, which is hard to believe today when a paint program comes with every PC and children teach adults how to use its brushes and palettes. For several years, however—until the arrival of the commercial PC—I had similar difficulty explaining that it really does work to move your hand "down here" on a mouse or tablet while watching the result "up there" on a video display. The question simply disappeared with widespread personal computing in the 1980s, but at the time I didn't know what Dick meant.

My call to Dick was in February 1974. The entry for May 8 in my on-again, off-again journal tells it all:

> Another exciting event ... was a visit to Dick Shoup in Palo Alto. I went down to Stanford to check some references for my polyautomata survey and spent the night at Dick's apartment. After lunch next day he persuaded me to visit his project at Xerox. I was reluctant to go because I had visited before several times and had listened to his dreams for 2-3 years. but had never seen anything tangible. I went because of his hospitality—and was greatly and very pleasantly surprised! His machine finally exists: a color TV "paintbrush" hooked up to a computer. It is dazzling. I had to wait a few days to return to Palo Alto (had to meet a February 28 deadline on the survey) but spent 12 hrs. on the machine next visit. It's such an incredible invention [that] I've decided to record this chronicle of my excitement and involvement with it.

I stumbled out of Dick's lab finally knowing what a paint program is; I'd found in California that "something good" I'd envisioned while recuperating immobile from that New Hampshire ski accident. I created my first several animations on his system in May and June, and Dick made noises about hiring me in September. I presented a proposal based on CA to him, but he told me his higher-up, Bob Taylor, wouldn't buy it. We loosened up over some beers at the Dutch Goose, a famous Stanford beer joint, and came up with an approach, a film demonstrating the capabilities of SuperPaint.[54] Next day, I talked with Bob Flegal, Dave Liddle, and Ron Baecker, and gained support for the film proposal subsequently presented to Taylor. On 31 July 1974, Dick Shoup and I formally joined forces at PARC. We were both just turning 31, birthdays only a month and a half apart.

I found living quarters in a house in Los Altos Hills above PARC, renting a room from Richard and Sandra Gilbert. This seemingly innocuous detail would figure in an unbelievable way in my future.

I wasn't exactly hired by PARC. No employee slots existed when I arrived, so my several supporters there, including Alan Kay, Dave Liddle, and Bob Flegal, managed with Dick to contract me with a purchase order (PO) instead—like a piece of equipment! I didn't care. I was an artist and just wanted access to Dick's program. A product of the 1960s, I was having trouble with corporate America at the time—surprising perhaps in light of my future career—so I probably wouldn't have been a good employee anyway. I'm sure this attitude didn't help in subsequent events at PARC.

Dick is one of those rare persons proficient in both the fleeting mind of modern computation and its enduring body. He not only wrote the software for SuperPaint, he created its hardware too. He built the crucial *frame buffer*, a computer memory specialized for holding pictures. Ordinary computer memory, the well-known RAM of personal computers, stores everything in discrete bits and pieces—actually called bits at the lowest level. But pictures don't fit well in one-dimensional lists, so a frame buffer memory stores a picture divided up naturally into rows and columns of tiny colored points, each called a *pixel*,[55] in a 2D array of memory locations.[56] Two dimensions make a frame buffer memory special, but that's not all—you can see it, too.

A paint program needs a canvas. The frame buffer provides one by delivering its memory contents to a video display. Dick added circuitry to his frame buffer design so that each pixel, in succession, lights up color phosphors, in the same order, on a television monitor.[57] So the 2D picture stored as bits of electricity in the memory becomes a 2D array of colored light emitted from the face of the television—a picture! SuperPaint, or any paint program, works by letting an artist change the bits in a frame buffer at locations specified with a simulated paintbrush, hand-driven by a mouse.[58] Because of the direct and immediate correspondence between the electrical bits in the memory and the physical phosphors on the TV screen, this feels like virtually painting on the screen instead of fiddling bits in the memory.[59]

Dick called his memory a "picture memory," but "frame buffer" stuck. It held one video frame in a buffer zone between the high-speed uptake of the television eye—well, the inverse of eye actually, with the picture going out, not in—and the slower, blind computer brain. The SuperPaint frame buffer eventually evolved—crossbreeding with the crude text displays of early machines—into the color graphics display card, an essential part of every modern PC. The early stages of this history of painting can be construed as a search for that rare beast, the next frame buffer, with Dick's being the first.[60]

The color map innovation did not yet exist when I began at PARC, although Dick explained to me how it would work when he finished it. The color map inspired my first formal contribution to PARC and computer graphics, namely, a more natural way to select a palette of colors in SuperPaint, informed by my artistic experience.

SuperPaint created colors by mixing red, green, and blue (RGB) lights, just like home TV. In fact, both used the same electronic technology, that of broadcast video. Thus RGB color space controls are the natural ones for computer graphics and video, but I had trouble mixing, say, pink or brown with them. I asked Dick for the algorithm that converts RGB to and from the more intuitive hue-based world that I knew from painting: Mix a color by choosing a base with the desired hue; add white or black paint to lighten or darken it. So pink and brown are just red lightened with white and darkened with black, respectively. Surprised when Dick told me that no such algorithm existed, I sat down and invented one overnight—the first simple hue-based color mixing system for computers, the HSV algorithm—and coded it into Dick's paint program.[61]

But my main PARC contribution, in my opinion, was a videotape called *Vidbits* that showed, for the first time, the kinds of imagery and animation that could be accomplished in the new medium of the digital frame buffer—and I was convinced it was a new creative medium. To help perform my job at PARC, demonstrating the artistic dimensions of SuperPaint, Dick and video whiz Jim Mayer taught me video editing. I recorded the work in short pieces of several seconds each, edited the pieces together, and added a soundtrack, excerpts from Holst's *The Planets* in this case. *Vidbits* became a strong part of the sales pitch I used later for talking my way into NYIT, or New York Tech, and my entrée into New York City's video art scene.[62] It was shown briefly at the Museum of Modern Art in New York, thanks to Barbara London, the adventuresome video curator. Nevertheless, I am not unhappy that my personal copy of it ceased to replay years ago.

Meanwhile, I cast my lot with San Francisco film and video artist David DiFrancesco and the National Endowment for the Arts (NEA). David, originally of Nutley, New Jersey, was to become one of my closest lifelong friends. He had already been on a worldwide quest to find ways to use the computer in his art when I met him. He had worked, under an NEA grant, with Lee Harrison and Dennis Kolemainen at Computer Image Corporation in 1971–1972. David then lived briefly in Japan using the Scanimate machine, built by Computer Image, at Kodak's Far Eastern Laboratories in Tokyo. He had just returned from Japan when he saw a demo of SuperPaint that Dick and I gave to a group of artists in San Francisco. From the audience, he asked Dick what SuperPaint had in common with Harrison's machines. Dick knew, because Lee was a friend, and encouraged David to call Xerox PARC.

Dick asked me to handle this caller who, so far as I was concerned, was just another of many artists trying to gain access to SuperPaint. But I was wrong. This guy was different. He was very persistent. I finally relented only after he had brought ceaseless beseeching, natural charm, and unusual humor to bear. I suggested that we share an evening "jamming" on the paint program. This meant time-sequential jamming, where I would paint for a while, then he would take the result and add to it, then I to his, and so forth.

We hit it off tremendously. David's irreverence and mine meshed, as did our off-key interests. When I mentioned my 1950s collection of "pictures of the future" clipped from such magazines as *Popular Science*, *Mechanics Illustrated*, and *Ford Times*, he knew exactly what I'd done and why. In fact, he had worked with the Ant Farm, an art group I admired,[63] on this very subject. Under his insightful humor, he had a sense of dignity I'd not encountered before. But the essential factor to which I responded was his love for machines as much as for art. Because he liked all machines, he could put up with the inherent pain of working with the early digital ones, could drive right through it for the results he could get—part of the adventure!

This tolerance of pain—indifference to it, really—was characteristic of early computer artists. Another David, David Miller, jammed with us at PARC and went on—as David Em—to join his accomplished technological cohort Jim Blinn at Pasadena's JPL and suffer for years in the fluorescent lights, roaring and freezing air conditioners, and institutional green walls there to create his sunset-beautiful digital paintings and strangely tiled 3D landscapes. You had to believe the technology would become less harsh in a couple of years.

Other visitors to PARC were my housemates Richard and Sandy Gilbert. I loved showing our work because nearly everyone was entranced by it—quite a change from CA theory. I thought it was particularly important that the Gilberts see what so excited me—so they would understand why I hardly ever came home and when I did, stayed only long enough to get a few winks before heading back to PARC.

Around Christmas 1974, Richard took a vacation to New York. Upon return he tried to interest me in his uncle's activities there: "He's doing what you're doing, Alvy, I think." I dismissed this talk, since Richard was an economist with only a single glimpse at my world. Besides, I thought I was completely in touch with the gossip of this particular underground. I assumed that I knew every related activity in the world. Richard's uncle certainly did not figure in that universe. This arrogance backfired soon.

David DiFrancesco, familiar with the ways of "art biz," soon suggested that he and I ask the NEA for a grant to exploit the new artistic medium, the frame buffer. He needed the money since he was at PARC only unofficially. And I wanted to make art. My experience with government grants had been several National Science Foundation grants obtained as an NYU professor. These were 40 pages thick and submitted in "20-plicate," so I listened in disbelief as David told me that a single page qualified for an NEA proposal. We submitted our work instead; he had obtained NEA grants before—was living on his second one then, in fact—and knew the ropes. We made a video piece using the paint program and its frame buffer and submitted it with a one-page grant proposal, thus cementing our relationship.

Bad timing. Xerox chose then to pull the rug out—deciding not to do color! I was dumbfounded. This was like a major film studio deciding, in the 1930s, not to do sound. I was convinced that color was the future of computer pictures, that Xerox had the lead on the world. I tried to argue this vision to my managers, Jerome Elkind and Bob Taylor, but they informed me that it was a "corporate decision" to go with black and white—silent movies were good enough.

I shouldn't have been surprised. The rumblings had begun almost at once. I was dismayed, just after my arrival at PARC, to have Dick's boss, Bob Taylor, ask, "Don't you think Dick's program is hard to use?" No! It's impressively intuitive! I shouted inside, trying quickly to take stock. He pointed approvingly at William Newman's black-and-white project in the corner of the SuperPaint lab room as the future of Xerox.[64] I should have recognized my risk from that flabbergasting moment, but I counted on the outright obviousness of our work, soon to come.

Taylor's bosses, at corporate Xerox in upstate New York, were having trouble too—all of PARC was at fault in this case—with the public perception of its shaggy-haired, bicycle-riding, sandal-wearing think tank, way out—far out—in California, their fear heightened by Stewart Brand's hip article[65] about the place in a 1973 issue of *Rolling Stone* rock'n'roll magazine—not exactly button-down Webster, New York, reading material. A skunk works really stunk to them. And things were a bit extraordinary, such as

Dick's riding his bike literally into the building to work. Beanbag chairs served as office furniture. Fritz Fisher, a mathematician with a master's degree, got himself hired as a night watchman so he could hang out with SuperPaint in the wee hours (and read for fun a massive book on the general theory of relativity).

My position at PARC was extremely vulnerable, having been secured with the purchase-order "hire." So when Xerox decided "not to do color," it simply canceled my PO (which I still have). Elkind and Taylor gave me the final No on 16 January 1975. My journal entry that day: "End of an era: It's been wonderful. I believe extension of my access was not granted in a decision based on fear." Probably not. Probably just corporate blindness—that I would better understand later, from the other side of the table—but before I left, Taylor made me edit out every reference to Xerox in the hours of videotape I'd made at PARC—tape later used to help Xerox win the Emmy.[66]

New York Tech and Paint, Paint3, BigPaint, BigPaint3

I lost my job, but the more important loss was access to Dick's frame buffer. The NEA grant proposal depended on it. David and I heard that another frame buffer—the next frame buffer—was being built in Salt Lake City. The Evans & Sutherland Corporation was building a commercial frame buffer, and the University of Utah Computer Graphics Department would garner the first one. So we made a mad dash in my big, white Ford Torino—my Turin machine, I fancied it—out across the dazzling, snow-covered Nevada desert on a mission. Our goal was access to that frame buffer. Our naïveté was amazing.

First, we visited E&S where, magically, in the parking lot on the way in, we met Jim Kajiya for the first time—a striking combination of Japanese features, black waist-length ponytail, and a yellow-and-black Lotus—a designer of the E&S frame buffer, the grail we sought.[67] Soft-spoken, well read, and extremely intelligent, he would become one of the outstanding academic contributors to computer graphics as a professor at Caltech, the California Institute of Technology in Pasadena, and, decades later, my still-longhaired colleague at Microsoft. He had stopped cutting his hair in 1968 (and still hasn't cut it).

E&S reeked of the Defense Department, so we didn't even attempt to scale its walls. Instead, we tried to talk our way into the university. But David and I didn't work out: We were a mismatch with Utah. We could see but not touch the new frame buffer. We soon learned that the Computer Science Department, too, was Defense Department funded so avoided the "art" word, but one look at us said that art was the theme, explicit or not, of our presentation. We were told that the Defense Department was threatening to cease funding the CS department (it did) and that their quota on artists was soon to be filled with an extended visit from Judson Rosebush, then with the Everson Museum in Syracuse (this never happened, either).

Someone at Utah finally spoke the crucial words that softened our disappointment: "You want to talk to Dr. Alexander Schure, a wealthy madman from New York who came through here recently and bought one of everything in sight." Yes, including the not-yet-delivered frame buffer. He had animators from Hollywood making an animated film. "You can talk art with him," we were told.

Very likely Martin Newell first mentioned Schure. Newell, the most life-changing Utah person we met that trip, was a calm, deliberate, highly respected Englishman teaching computer science. Newell's personal teapot would become the most famous icon in computer graphics: Many graphics researchers would use it—actually a 3D database that he measured directly from it—to show off their latest rendering tricks at conferences. The original teapot itself now resides honorably in the Computer Museum on Moffatt Field, along with the Shoup SuperPaint system.

Newell informed us of his impending visit to consult at Schure's NYIT. Where? Despite four years in the academic sphere of New York City, I'd never heard of this place. Newell told us he would call, upon his return, with a report. He did. His message was simple: "If I were you guys, I would jump on the next plane." We did. I spent the last money I had for the plane fare.

I was ecstatic with this turning point, bubbling with the news to housemates Richard and Sandy about my future—my new plans, if they worked out, would entail leaving them soon. I chatted energetically about the animated movie project at New York Tech, about the report that the New York Tech campus among the wealthy North Shore estates of Long Island was like a movie set, and, of course, about Alexander Schure who was the key to the next step in my, and David's, life. That's when Richard floored me: "But, Alvy, he's the uncle I've been trying to tell you about!" This coincidence continues to baffle me, decades later. Of all the people I could have rented a room from … .

So right then I began to call Schure "Uncle

Figure 2. Clockwise from upper right: Garage of the Gerry House, home of the Computer Graphics Lab at NYIT; DeSeversky Mansion, held the video studio and gave visitors their first impression of NYIT; the carriage house on the McGrath estate, home of several Lab members; and Holloway House, home of several other Lab members.

Alex." The name stuck. My colleagues and I at New York Tech called him that for the duration, although it was "Alex" or "Dr. Schure" to his face. Uncle Alex was to be the first of several wealthy and fascinating patrons in my career. George Lucas, Steve Jobs, and Bill Gates would be the next three.

We initially stayed with David's father in Nutley, New Jersey, eating scrumptious homemade apple pies in his gadget-filled home. We waited a day for a February 1975 blizzard to subside. Next day, we forced our way through the snow across lower Manhattan, out onto Long Island a few miles beyond Queens to the exclusive North Shore community of Old Westbury, home of NYIT, or New York Tech.

We found our way onto the campus, a collection of former estates, and to one of the mansions, Gerry House. Schure formed the campus from several adjoining estates (see Figure 2). The Gerry, Whitney, Goodyear, and Holloway estates, and several others, figure into the campus itself or into nearby homes for faculty or staff. The "mansion next door," the former Whitney home renamed the DeSeversky mansion, had played the role of baronial estate in several films—for example, *Arthur* (1981). Schure himself played a cameo role in one of them, *Three Days of the Condor* (1975), greeting a helicopter then opening the door of a limo. There are even "spare parts" for the mansion in the woods nearby—extra columns, capitals, and friezes.

We had been told to find Dr. Ed Catmull—only recently graduated from the University of Utah and just hired by Schure to manage his new E&S toys—at Gerry House. Schure had found his way to Utah by way of an aggressive salesman, Pete Ferentinos. The eastern US representative for E&S, cold-calling on all the universities in his territory, hit pay dirt with New York Tech and Alexander Schure. Pete's vision of making movies with computers excited Schure so intensely that he made the trip to Utah to meet Ivan Sutherland (half of Evans & Sutherland). On his shopping spree there, David Evans (the other half of E&S) asked him who would run the new equipment for him. "Who should it be?" asked Schure. "Well, you just missed the right guy," responded Evans. "Ed Catmull has just taken another job out of desperation." Myth has it that Schure said, "Money has a way of changing people's minds." True or not, it wasn't money that changed Ed's mind. It was the opportunity he had been waiting for.

Ed had always wanted to make animated movies with computers. One of his PhD advisors, Sutherland, had gone to Hollywood to start a company to do just that. Ed was to join it when funded, but needing to feed his wife Lorraine and child David, he finally had to give up on Sutherland. The company, called Picture/Design Group, formed with Gary Demos (who also worked on hardware design of the E&S frame buffer), John Whitney Jr., Glenn Fleck, and Barry Wessler, never got funding—a victim of the 1974 recession. Whitney and Demos teamed up to make movies with computers—and to become the loyal opposition to Ed and me in that endeavor in the coming years. Ed took a job in Boston at Applicon, a major computer-aided design (CAD) company. But Schure's call was what he'd awaited—if in an unexpected form. He asked Malcolm Blanchard, his Applicon officemate and fellow University of Utah alumnus, to join him, since Malcolm had some systems programming experience, which was certainly required to get a nascent computer center off the ground. So, in November 1974, Ed began as head of what we would call "the Lab" for Dr. Schure, and Malcolm joined him in January.

The Lab resided in a converted garage of the Gerry House that David and I approached through the snow on that first NYIT day. We still say that we got started in a garage—it just happened to be a four-car garage—with chauffeurs' quarters above. In it were two people on that fateful day. I addressed the first person as Ed, but it was Malcolm. Unexpectedly, Malcolm sported shoulder-length hair; Ed, a beard. Good signs!

Soon Ed, Malcolm, David, and I jumped into a waiting limousine that escorted us to the DeSeversky mansion on the adjoining estate—completely unaware that we would spend most of the next twenty years together. The limo wheeled us past a gatehouse, up a winding drive past a pond and manicured lawn adorned with flocks of Canada geese. We crossed the floor mosaic of the grand foyer and entered the dining room, complete with gilded mirrors, silver service, and liveried waiters. "Welcome, California!" boomed a voice from a table at the far wall, the only occupied table in the spectacular room. Thus Alex Schure introduced himself, a highly staged and very impressive piece of theater—theater which was to become quite familiar to us over the next four years.

Fast and furious conversation ensued, although Schure talked in what I call "Casey-Stengel-speak" and David calls "word salad." "Our vision will speed up time, eventually deleting it," he would declare to a reporter in a particularly infamous example.[68] Linear rational speech, as usually practiced in human communication, instead became from him an engulfing torrent of words. Somehow, miraculously, thoughts were transferred, but one was never clear how or exactly when it happened—more as if coming from poetry than speech. Hearing one's own words from Schure's lips signaled a successful idea transfer. Strange as it seems, this system worked.

Within minutes we were in the limo again, being delivered this time to the Schure home, on another nearby estate. We had brought videos, *Vidbits*, and the NEA submission tape. He claimed to have the necessary 3/4-inch Sony U-Matic tape player in his house, for playback of the video lingua franca of the time. Actually, he had more players, one in each of several rooms. Unfortunately, most of them didn't work. We had to move from room to room in concert with his wife Dorothy who moved the "killer dogs" out of each room ahead of us so they could not spot us and—presumably—tear us to shreds.

Finally, we located a working player, watched the tapes, and talked at high speed for about an hour. We discovered he had been hard at work on a feature-length animated film, *Tubby the Tuba*, being realized with conventional cel animation à la Disney, having hired dozens of Hollywood and New York animators, producers, inbetweeners, checkers, cameramen, and so on—all currently on campus generating the movie. We didn't have to hide the word "art." We were more than welcome in this mix, the upshot being our having obtained permission—not jobs, just artistic access—to use his soon-to-be-delivered E&S frame buffer. We promised to close up in California and return in one month. We had found the next frame buffer!

Before leaving for California, I talked further with Ed Catmull. David and I were two hippie artist guys, so concerned Utah friends had warned us that he was a very straight, Mormon missionary type, or at least that's how they painted him. Having grown up as a Southern Baptist, I was used to the religious, and I fancied my family to be Mormon several generations back (subsequently I have proved there are indeed Mormon branches of the family). So I assumed this probably wouldn't be a problem. In our talk, he was as excited as we about computer graphics technology and its promise and said all the right things to us about art and movies. He was clearly unconcerned with my long hair and David's electric frizz. Then we found out why. A prolonged bout with the Mormon Church had "radicalized" him. Ed also refused to buy the Vietnam War and fought for Conscientious Objector status. Good patriotic Mormon boys just didn't do that. Utah officials made it rough for him.

My most important discovery was that he had much too much to do. Only two people setting up an entire animation studio? "You need some help, don't you?" I observed. He seemed relieved and quickly agreed. I told him I had a PhD in computer science from Stanford, had recently been part of the NYU computer science faculty, and could program well. Voilà! A talk with Alex Schure and I was on. A job—and access! It would take another year to get David hired, too—Schure was unsure of an artistic type without even an ameliorating PhD—but we didn't let that bother us since Schure tolerated his unpaid presence. We had solved the NEA grant proposal problem.

I spent the first week of April 1975 in the Netherlands at what I think of as the 0th Artificial Life Conference, and then officially joined New York Tech the second week. I was the old man at 31; Ed and David were just barely 30 and 26, respectively, and Malcolm would turn 25 a few days later, all three of their birthdays within three weeks of one another. Thus began the NYIT-Lucasfilm-Pixar computer graphics dynasty, a marriage of the house of Xerox and the house of Utah, pixels and geometry, art and technology. The movie we dreamed of then—completely generated on computers—was first shown in November 1995, 20 years later. *Toy Story* was that movie.

We fantasized and played at the fabulous Great Gatsby campus of New York Tech, but we also worked—maniacally. We had waded ashore a new continent—The Work—and didn't mean to waste a moment before claiming territories. The frame buffer hadn't arrived from Utah, so we began to master the E&S Picture System and the 3D line geometry it could display, and there was a new computer to learn: a PDP 11/45 from DEC, the Digital Equipment Corporation. It was only natural that Ed, as one of his first acts at the Lab, ordered a DEC computer rather than an IBM machine. This would have surprised corporate America at the time. The International Business Machines Corporation was the dominant force in computers. IBM was Snow White to the Seven Dwarfs, including DEC, a very distant second. But computer science students, including those of us just freshly minted, knew and loved DEC machines. IBM computers, the mainframes, resided at the university computer centers. Students handed their decks of punched cards (IBM cards, with rectangular holes punched into them on IBM keypunch machines) to staffs that ran the big machines. DEC, however, made small machines, "minicomputers," and placed them in the hands of students. Naturally, now that our time had come to buy machines, we did business with the company that had paid attention to us, a lesson not lost on Ed and me when later we entered the hardware business as Pixar.

IBM lost again when it came to choosing a programming language for the Lab. Ed and I both hated Fortran, the predominant programming language of the time, from IBM of course. We had both taught it and knew it very well. We believed that the hegemony of IBM had foisted an inferior language on the world when much better ones existed, like Algol and BCPL.[69] (Tough words from someone whose most recent job was with Microsoft!) We elected to program in assembly language rather than use Fortran, while we shopped for a high-level language nicer to humans than Fortran was.

Our forbearance paid off. Ron Baecker, from the University of Toronto and briefly a housemate of mine in Redwood City, California, soon informed us of a bright new computer system exciting the universities. It was called Unix, from AT&T's prestigious Bell Labs. It came complete with a high-level programming language simply called C and could be bought for a song (about a hundred dollars) by educational institutions like NYIT. We purchased Unix and immediately became enamored of it and its language C. They were logical, natural, and simple. Twenty years later, Unix is the system and C—or one of its immediate descendants, C++ and Java—the language of choice among most academicians and scientists. Unix and its variant Linux are currently being widely used on World Wide Web servers.

We were thus among the earliest users and proponents of Unix and C. Programs written in C are not only structurally more elegant, they look better than Fortran, too. More importantly, though, C greatly increased our productivity. The Lab would churn out code prodigiously for the next several years, due in part to the efficient C programming tool. We were rewarded again for sticking to our beliefs. We believed in color, and we didn't believe in Fortran.

But we didn't believe in writing things down. We were too busy. None of us recorded anything, failures or successes, a practice that would haunt us for years—still does, in fact. We saw each step we made as "obvious to those

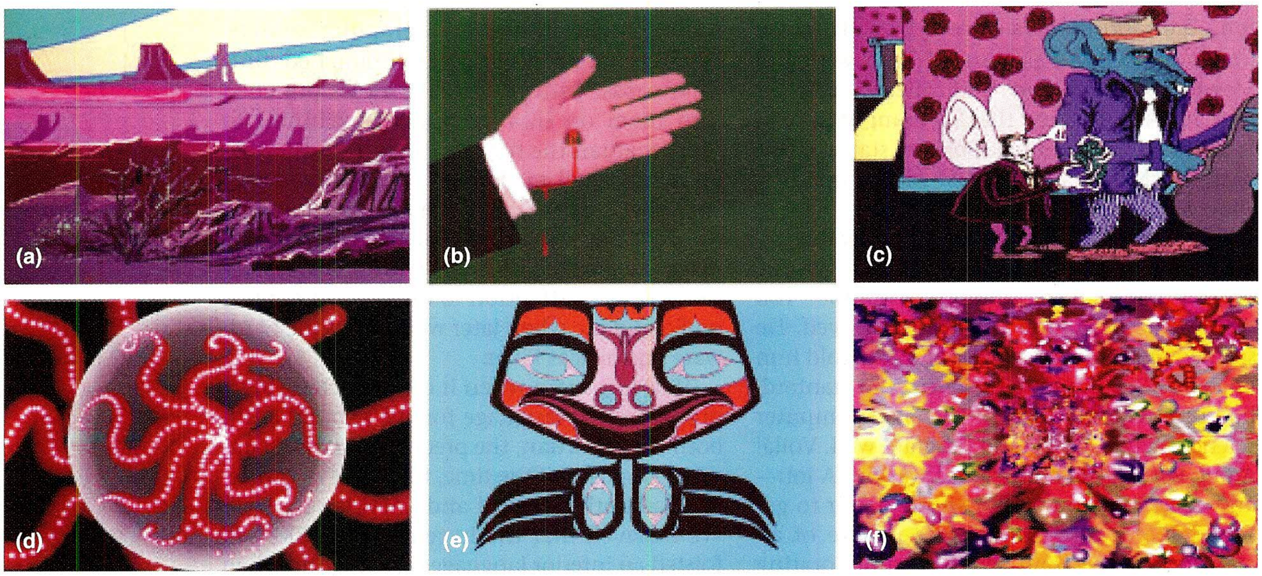


Figure 3. A sampling of the hundreds of pictures made by members of the NYIT Computer Graphics Lab in the period 1975–1979, using the programs Paint, BigPaint, and Paint3. (a) The desert landscape is thought to be the first painting by Paul Xander with Paint, in 1975. (b) The stigmata (Ephraim Cohen), (c) gangster rats (Lance Williams), (d) worms (David DiFrancesco), and (e) Haida motif (Christie Barton) were created using Paint features in the period approximately 1975-1977. (f) The Lance Williams explosion, one of the first serious artistic uses of Paint3 in 1977, began as a scan of the Mona Lisa, whose eyes still peer (mid upper center) from within the chaos. All images are video resolution.

practiced in the art," in the US patent terminology unknown to us at the time. We were 1960s rebels creating for the greater good of mankind; we showed everything we did freely to anyone who asked. We would be legally denied our priority for years by johnny-come-latelys who patented our techniques a decade after we "invented" them in the 1970s—a term much too grand, by our understanding of scientific priority, to be appropriate. We had much to learn about the US patent system—for example, that it honors claims that would have the claimant laughed out of scientific meetings in shame. But the patent battles happened much later.

The E&S frame buffer finally arrived, and I was back in my element. My urge to make art on the NEA grant David and I believed we were sure to get and the need for a tool to create animation backgrounds for the Lab meshed to drive me into writing the Lab's paint program. This pixel packing was natural to me. After all, I'd just graduated from the tutelage of Dick Shoup, the original master. But I had an advantage over Dick. I had the $100,000 Picture System as a menu monitor rather than making double use of the video monitor as Dick was constrained to do.

I quickly introduced several innovations. In Dick's SuperPaint, a paintbrush was selected from a predefined set of brushes. In my Paint, any image of any shape could be used as a brush. I generalized painting to types of painting. Instead of just simulating painting a stroke of constant color, I extended the notion to mean "perform any image manipulation you want under the pixels of the paintbrush." For example, I added "not paint" that reversed the color of every pixel under the paintbrush to its color complement, "smear paint" that averaged the colors in the neighborhood of each pixel under the brush and wrote the result back into the pixel, and "z paint" that treated each pixel value as depth in the third, or z, dimension and only overwrote a pixel if the depth value in the corresponding brush pixel exceeded that in the pixel underneath. I extended Paint to BigPaint, which could paint on images larger than the frame buffer screen.

The train of visitors to see the wonders, such as those in Figure 3, being generated by my colleagues and me at NYIT began early and never ceased for the entire NYIT era. There were the daytime visitors and the nighttime ones. Ed and Malcolm both had families, so worked traditional times. David and I were night people, a trait that worked well because there was only one computer system for the four of us to share. In actual fact, David and I freewheeled, meaning that we worked as long as our bodies would take it, slept at the Compound[70] as little as possible, then hit the Lab running for the next stint. I found that I had a 26-hour cycle, so my days started completely out of phase with Ed's and Malcolm's, then shifted slowly into phase every couple of weeks and then back out again. This also worked because I needed to interact with the two of them a great deal for Lab work. David and I got out of sync, too, but less seriously because we were working together a lot.

The daytime visitors included a steady stream of computer graphics people who were learning of the new mecca on Long Island. Over time, anybody who was anybody in computer graphics or art would find their way to the Lab. Hardly a day passed without visitors, a principal joy of New York Tech. Years later, in a patent squabble, we would be accused of being secretive with our work. An impossible thought! Hundreds of visitors, if not thousands, had trooped past our workstations, day and night, and seen The Work. Of course, with our lack of records, we had no proof other than verbal testimony for the courts.

One regular group of visitors was from Cornell University, the computer graphics students under Don Greenberg. We quickly noticed a pattern. These bright students, like Marc Levoy, would visit, then the next Siggraph would feature a paper about work similar to ours on equipment similar to ours. They were buying the same equipment just after we did. For example, not long after we had an E&S frame buffer and Picture System, they had the same combo and had written similar software for it. Eventually, this would matter in the patent wars.

I began getting many speaking requests and doing lots of traveling. After one particularly long trip away, I came back to the news that Alex had sold Paint to Ampex. I was shocked because, even then, we knew that one didn't sell a program, one licensed it. Recall that this was before the PC and the notion of commodity software. I was dismayed but agreed to go to Redwood City, near Palo Alto, to install Paint at Ampex and train people to use and modify it.

Meanwhile, Uncle Alex kept improving the Lab, making it the wonder that it was. He passed through every morning and every evening about 5 a.m. and 5 p.m. His hours were as strange as ours. He sounded a constant theme: "What do we have to do to stay ahead?" I explained to him, in the laborious communication method already described, that two more frame buffers would be an immense improvement. If we ganged three frame buffers together we would get an RGB frame buffer, capable of true color—not just 256 colors but 16 megacolors. Why is that interesting? Well, 256 different hues seems like a lot but to do antialiasing at every edge requires the mixing of each hue with every other hue. For example, at a slanted edge between a red line and a blue fill, we needed about 256 different mixtures of red with blue. For us to have enough colors so that this issue simply disappears between all possible pairs of hues, I explained to Alex, give us two more frame buffers. Sixteen million colors is so many that, for all practical purposes, it is a color continuum for human beings.

Several weeks later Uncle Alex happened to mention amidst his usual poetic ramblings, "Oh, I just bought you five more frame buffers. Now you can have two RGB frame buffers." Exultant, we chorused, "Thanks, Alex!" E&S now charged $60,000 each for these. With the $80,000 price tag on the first one, this came to an expenditure—on an almost casual request—of $380,000, or about $2,000,000 in 1999 dollars—just for the frame buffers! The first two commercial RGB frame buffers in the world! And perhaps the first ones at all, but we didn't know what the government had in its mysterious Cold War labs and still don't. We did know that Alex Schure was funding us magnanimously, kick-starting the industry.

Ed, Malcolm, and I took advantage of the big order to E&S. We requested that a new hardware device be attached to each, and retrofitted to our first frame buffer, that would make programming even easier. Since the frame buffers were to attach to the DEC computer through its component called the Unibus, the box diagram we sent to E&S engineers featured a box labeled FBUNI, for frame buffer-to-Unibus interface. The Utah engineers read this as "eff-bunny" rather than our "eff-bee-you-nee". The campus of NYIT was populated with dozens of little bunny rabbits, and the novel *Watership Down*, featuring a warren of adventuresome bunnies, had just been published to much acclaim.[71] As denizens of our own Watership Down, we were delighted by the Utah reading of our label and instantly dropped the F. We called the new device a "buni" henceforth, adding six more bunis to the local warren.

I went nuts. I had written by now a complete suite of raster graphics tools: Paint, BigPaint, Fill, Clr (clear a window in the frame buffer to a constant color), ColrMakr (to design color palettes), Flip (the frame buffer horizontally or vertically), Savpa and Getpa (save and restore a frame buffer image to and from a disk file), and so forth, plus dozens of *hacks*, simple programs of only short-term interest. Immediately on arrival of the new frame buffers, I set about converting all my tools to the new world of RGB. Paint became Paint3; Fill, Fill3; Savpa, Savpa3; and so on. Every adapted program was a first. Nobody had had RGB, *true color*, before. To put this in perspective, each of these programs had to run in 32 kilobytes of RAM; the PC I wrote this article on has 768 megabytes.

I remember clearly the moment the notion of RGB paint came to mind. I was in Redwood City for the installation of Paint at Ampex. It was night and I was alone in a mediocre motel room and bored. So I turned my thoughts to the new frame buffers and realized that it would be a snap to rewrite Paint to make it RGB savvy. Essentially, all I had to do was take each line of code and triplicate it, while keeping the control structures (*if* statements, *for* loops) constant. This I did immediately upon return to Long Island. It was indeed as easy as I imagined, so you can see my puzzlement at calling this an "invention" as opponents were later to do. I claimed then and still do that anybody with an RGB frame buffer would have done the same thing. I just happened to be first.

The most important consequence of RGB was that Paint3 could be antialiased. With only 256 colors, Paint, like SuperPaint, used brushes with jagged edges. There simply weren't enough colors to form all the mixtures at the edges with the other colors that might fall under the brush. But Paint3 allowed "airbrushing," using brushes with very soft edges.

This term, airbrushing, came to haunt me several years later so let me explain it. The notion was that a brush is partially transparent. For an "airbrush," the center of the brush is more opaque than the edges and there is a gentle increase in transparency going from the center of the brush to its edges. Where the brush is opaque, solid color is applied to a painting. Where it is transparent, no paint is applied, of course. Partial transparencies in the brush cause the brush color to be mixed with the background, the amount of mixing proportional to the amount of transparency. Low transparency (high opacity) means that the brush color hides or almost hides the background colors; high transparency means high amounts of background show through by mixing only slightly with the brush color.

Now here's the problem. I didn't call the process above "airbrushing," as it's called today in all RGB painting programs. I called it "wetpainting." I didn't call it airbrushing because the process described above is not a simulation of real airbrushing. A real airbrush works like this: Compressed air is forced to pass over the top of a bottle of paint—real paint: particles of colored pigment held in suspension in some liquid such as oil or water. The compressed air rushing over the bottle of paint causes some of it to rise out of the bottle and be mixed with the air stream. The result is a steady stream of pigment particles being deposited on the surface being painted. The tiny particles build up faster near the center of the air jet and less so near the edges of the air pattern. The particles are randomly placed within this pattern. The "airbrushing" procedure, described in the preceding paragraph, has no randomness; the true airbrushing procedure does.[72] So I claimed that "airbrushing" was the wrong metaphor for what we did in an RGB paint program. Obviously I lost this argument, but more importantly, at least one company was almost driven out of business about a decade later partly because I bothered to draw this distinction in print.

One of the few developments I actually bothered to write down also came back to haunt me—to haunt the industry, really. The story of the airbrushing patent follows shortly. Suffice to say at this point that whether one calls it airbrushing or wetpainting, the concept of the weighted brush was so obvious that it never occurred to my colleagues or me to patent it. Like everything else we did at NYIT, we showed it freely and gladly to hundreds of guests. I liked to describe it to them as "painting with ice cream," because the colors melted so nicely together. I also presented it to the annual computer graphics show Siggraph in a tutorial. The tutorial needed notes, so that's why I bothered to write this particular development down. Thank goodness for that little bit of writing.[2,19]

And thank goodness for Marc Levoy at Cornell, now a Stanford professor. Soon after our RGB frame buffer and Paint3 went into play, Marc Levoy was writing his true-color paint program for Don Greenberg. What we didn't know then, and didn't know until the 1990s, was that Marc had been keeping very careful records of every visit of the Cornell team to NYIT, including lists of equipment, and even the floor layout for it. Eventually he wrote papers on all his work, but didn't mention his extensive NYIT notes.

The Lab started to grow almost immediately. Lance Williams and Garland Stern arrived from Utah to spend the first summer, 1975, with us. Garland brought a paint program with him from Utah! It was one of several 256-color paint programs that emerged suddenly with the arrival of the E&S frame buffer at Utah. Jim Blinn, Jules Bloomenthal, and Garland had all written different ones. Garland's felt very good once a stroke of painting was initiated. It didn't require you to apply pressure while stroking, like Dick's SuperPaint, my Paint, and all modern paint programs do. However, one had to click once to start a stroke and once again to stop it. This is extremely unnatural and didn't catch on.

Figure 4. An early (1975) painting on the 8-bit Paint program by Paul Xander (signature at lower right), a professional background artist for whom the program was largely designed. The original was 512 × 486 pixels at the 4:3 aspect ratio of ordinary television.

Figure 5. A 1977 still life by Paul Xander, one of his first uses of the 8-bit program BigPaint for high-resolution paintings. The original resolution of this painting was 1024 × 972 pixels with a 4:3 aspect ratio.

Much of the ease of use of my NYIT paint programs resulted directly from requirements of Paint's principal first user, Paul Xander. Paul painted backgrounds for conventional animation in Hollywood before coming to Long Island to work on *Tubby the Tuba*, Uncle Alex's conventional animation effort. Schure assigned him to master the digital form of background painting, which meant he worked closely with me. Not being the least bit technically minded, Xander simply could not master difficult menu or keyboard sequences. Consequently, I had to seek the simplest user interfaces I could find. This, of course, was an excellent discipline for an interface designer that served me well for subsequent decades. And Paul was able to paint hundreds of dazzling pictures, as Figures 4 and 5 illustrate, many of which were eventually used for backgrounds in an educational video on the metric system produced by the Lab called *Measure for Measure*.

The first new permanent employee from the outside world, in addition to the original four musketeers, was Christine Barton. A woman! Who knew computer graphics! Christie came from Utah too, but not the university. She had been working at E&S. Ed put her to work designing a computer network for the Lab. This was before local area networks existed as a commonplace as they are today.

We now had the six stations in the Lab, each with a computer (DEC PDP 11/33s this time except for the original 11/45), a tablet (Summagraphics), a video monitor (Barco), a menu monitor (Three Rivers Graphic Wonders, except for the E&S Picture System at one station), and a frame buffer (E&S). The idea was to offload the work of listening to the tablets to a separate machine (a DEC PDP 11/34). The computer cycles in 1976 were so precious that it was a crime to have many of them used simply to listen to the tablet, which is seldom used as seen from the computer's viewpoint. So all of the station computers were to talk to Christie's central computer that would handle all the tablet traffic and pass only pertinent information to the stations. We would call her machine a tablet "server" today.

Another hire of future paint history importance was Ephraim Cohen. David and I first heard of Ephraim on our original trip to the University of Utah. While talking with Robert McDermott, another longtime computer graphics colleague then working on his PhD there, I noticed a computer terminal with Barbie Doll-like arms protruding from it like flesh rabbit ears—as if she jumped feet first into the puddle of beige plastic atop the display. He explained, "That's Ephraim Cohen. You'll understand when you meet him."

We would spend years with Ephraim eventually and come to appreciate Robert's description. He indeed turned out to be a zany one, but brilliantly so. He made himself memorable—as if the doll's arms hadn't already—by sleeping under the desks at the Lab, apparently never going home (understandably so since home was in New Jersey). He could sketch like a Rembrandt, especially groups of people and, when not up to that standard, could at least pillory you with a telling caricature rendered effortlessly—or charm you if you were a waitress and he needed a free meal. He would later write yet another NYIT paint program that was commercialized after I left, by a subsidiary of NYIT called CGL (named for Computer Graphics Lab, of course).

Two things that we talked about in those early days but didn't do were subpixel painting and pressure-sensitive painting. Early at NYIT we talked of pressure-sensitive styluses. It made natural sense if you were a painter. We tried to talk the three tablet manufacturers of the time—Summagraphics, GTCO, and Talos—into making a pressure-sensitive pen for us. The closest we got to it was a Summagraphics prototype, built specially for us. I hooked it up to my paint program and soon noticed two fatal flaws. The most serious problem was that the tablet dropped points. That is, while painting a smooth stroke, suddenly a single copy of the paintbrush would appear at some random place on the screen, not part of the stroke being painted. Clearly the tablet position-sensing hardware was losing bits at random times. This made the tablet and its pressure-sensitive stylus unusable. The second problem was that it was highly nonlinear in its response to pressure. This would have been tolerable, however, if the positioning mechanism had worked. As usual, we failed to write any of this down. It failed, so what was the point?

One of our first observations about the Quantel Paintbox demonstration, when we saw it in the 1980s, was that Quantel had talked someone into building a pressure-sensitive stylus for them. We puzzled about who they had found to build one that we hadn't. We never believed, as they would later claim, that it was an original idea with Quantel. We suspected, without ever knowing, that Summagraphics had built it, but our contact there had died in the meantime. We had noticed only that they had ceased making overtures to us, as opposed to the other two tablet companies.

Several times Ed Catmull suggested that I implement subpixel painting in my paint programs. The notion is simple: Paint into a frame buffer of much higher resolution than can be displayed—say, eight times higher resolution in each dimension—then average down to display—for example, each display pixel is the average of an 8×8 array of "subpixels." I always refused to do it because of speed. How to do it was obvious. Subpixel computations were being used in several places in computer graphics of that early time. Ed, for example, had a subpixel hidden-surface algorithm (implementation of which had led to our inventing the alpha channel one day). And the Sketch code that I adapted from Malcolm for my paint programs was realized with subpixel resolution. Neither of these programs had to stay up with the smooth motion of the artist's hand. But to add subpixel code to my paint, which did have to keep up with the artist, would have slowed it down intolerably on the computers that we had then.

Quantel had implemented subpixel painting in their hardware. They had to. As with other claims to "invention" that Quantel would later make, this too was an easy idea to us in the preceding decade, not implemented for practical or technical reasons, not for lack of understanding how to do it nor for lack of thinking about it. We certainly appreciated Quantel for their doing it well, but again it never crossed our minds to think they originated the idea.

One of the best uses of my paint programs at NYIT before I departed in late 1979 was by artist Ed Emshwiller. Those of us in the Lab one night had watched a TV special about this guy. He had been an abstract artist in Paris in his youth with an esoteric fame as a painter of 1950s science fiction magazine covers, signed "Emsh." Then he had made some of the first avant-garde 16-mm films. Along came early video art, and he plunged into that with early notable contributions. The documentary informed us that he lived in nearby Levittown! One of my colleagues suggested we call him and invite him over. I confidently claimed, "If he's who I think he is, he'll find us." Sure enough, Emsh showed up one day to explore the next new artistic medium, computer graphics. He announced that he had a Guggenheim Fellowship and wished to work with us for six months to make a three-hour movie. We burst into laughter, greatly unsettling him. "You'll be lucky to finish a piece of three minutes in that time," we explained to him. Thus began the most important artistic collaboration of my life and a mentorship I cherish.

It worked, as usual, because Emsh loved technology—explosives, physics (his brother was a physicist), The Bomb. He had made a film documentary of the Nevada Test Site, where he picked up the cancer that affected his face for years and probably the cancer that killed him. He and I talked about everything—children (why have 'em?), marriage (why bother?), war, personalities, getting old (what's it like?)—but mostly we talked art.

Figure 6. Ed Emshwiller (right) and the author working on *Sunstone* at the NYIT Computer Graphics Lab in 1979. Two of the five paint stations are visible. Note the NYIT labels on two of the monitors. The painting above the author's head is detailed in Figure 2. The monitor face appears to float in space because the camera was jostled between two exposures used to make this shot, lights on and lights off. The result captured our mood precisely.

Emsh would propose a scene he wanted to make. For example, he wanted to push a 3D face through a wall. I'd explain we couldn't do that yet, in the late 1970s, too hard to compute. But if he would alter the design like so, I could write a hack to implement it. Then he would take my idea and push on it, and eventually we would ping pong into a workable and artistically interesting shot. I would write the code and make it happen. He used the paint programs a lot, but he surprised me by eschewing the color at first. "Too overwhelming," he said. He only slowly started adding color to the piece, and very carefully. Colleagues Lance Williams and Garland Stern added parts to the final piece too. The result was *Sunstone*, in many museum video collections around the world today, and my proudest artistic achievement.[73]

Sunstone was the last major event for me at NYIT (see Figure 6). Emsh moved to Southern California to become provost at Cal Arts—the California Institute of the Arts—founded by Walt Disney. Malcolm Blanchard had already departed, a couple of years before, mostly because his wife didn't like New York, saying that if we ever got back to California, to give him a call. Ed Catmull, David DiFrancesco, and I left NYIT for California in 1979. We did give Malcolm a call shortly thereafter, and he heeded it, when the next major chapter in our lives unfolded at Lucasfilm in Marin County.

The trials

The trials began with an approach at a Siggraph in the 1990s, probably led by Robin Forrest, a Scot living and teaching in East Anglia, England, and a longtime colleague from early computer graphics. Robin is one of the very earliest professors of the science. An Englishman, a solicitor for his client Spaceward of London, accompanied him. With great earnestness they asked me to aid them in a patent challenge from Quantel, also of England, but a company I had obviously watched for years. Spaceward, on the other hand, I had not heard of, but I did understand their problem and immediately sympathized with them, annoyed by Quantel's claims on our technology.

Quantel had for years built and sold a beautiful realization of a paint program. When we had first seen the Paintbox at a National Association of Broadcasters convention, we had known immediately what we were seeing: paint done in hardware. There was no other way to get the speed they were showing at that time, in the early 1980s. The software-driven, general-purpose computers that we had used throughout the 1970s and would continue to use for decades were simply not yet fast enough to allow someone to sweep a large brush of paint across a screen and have the computation keep pace with the human. (They easily do this now, but not then.) And this was part of the Quantel demonstration of Paintbox.

What we knew—what all computer science

students knew—but what the patent offices of the world apparently didn't know then, was that hardware and software are interchangeable. This is a fundamental idea of computation. One can take any program and convert it into a special-purpose piece of hardware that does the same computation, but much faster—since that's all it does. A general-purpose computer can, on the other hand, compute anything, can execute any computation, but at the expense of running slower than a special-purpose machine. This is simply because the general-purpose machine has to handle all situations. One wins big with speed in hardware but loses all versatility. A piece of software can be rewritten, sometimes in minutes; a piece of hardware can take the good part of a year or more to redesign and reimplement.

The point is that we knew immediately that Quantel had "dropped it [painting] into hardware." We never for a moment thought they had "invented" painting. Paintbox appeared in the 1980s, after all, and we had done painting in the 1970s. This, of course, was the point of the upcoming trial *Quantel* v. *Spaceward*. Quantel claimed to have invented "airbrushing," or soft-edged painting—or "wetpainting" as I had called it in 1977. This was why Spaceward had approached me: to establish my priority and get them off a patent infringement hook. The Spaceward product was an airbrushing paint program aimed at the video market, directly competitive with the Quantel Paintbox, and cheaper.

Quantel also claimed, via British patents, to have invented digital compositing. To my mind, airbrushing and digital compositing are the same thing: One combines one image with another, using a third for transparency control. In the case of airbrushing (I'm using the modern meaning of this term, of course), one image is simply of constant color, representing the paint on the brush; another is the image to be painted on; and the third is the shape of the brush, which effectively controls how much of the paint color is to be laid down over the background image. The shape of a brush is not only its footprint but also the weights of its pixels. For example, the brush "cone9" in Paint3 had a circular footprint and weights at each pixel in the brush to approximate a cone—high in the center and sloping linearly off to zero at the circular edge.

Digital compositing is the same thing but the images all tend to be relatively large. A foreground image is combined with a background image, using a "matte" image—what we called an "alpha channel"—to control the amount of combination, pixel by pixel. Now digital compositing is simply a digital realization of the old technology of matting that came from the video and film worlds. As with airbrushing, we did not, and still do not, see the digitization of a well-known process as "invention." It is too obvious and too simple to warrant such glory. But Quantel had been able to bamboozle the patent office of the UK and eventually of the US into believing just that. I accepted the pro bono job with the Spaceward team with the blessing and support of my colleagues at Pixar and the computer graphics world in general. This was clearly an example of misuse of technology. Not only that, but they were claiming priority over my work, however simple it was.

The trial—the first of the two described here—took place in London in 1989. I was put up at a nice hotel on Aldwych Circle in the theater district and not far from the courts. Since I was in over my head legally, I took along our Pixar patent attorney, Gary Hecker, as an advisor. He came in handy. At one point, before I was called to the witness stand, Quantel asked for a private meeting with me. I asked Gary what that could possibly mean. He thought perhaps they were going to suggest some kind of deal to keep me off the stand, because we both thought I had the killer argument: I had done it first, had paper to prove it, and the testimony of colleagues to hammer that in.

The meeting turned out to be an attempt by Richard Taylor of Quantel and his attorneys to find a weakness in my testimony that they could exploit. This went nowhere so the meeting broke up. Richard and I were the last two to leave the room. Just before doing so, I turned to him, looked him in the eyes, and said, "I did this first, you know, of course." I just couldn't believe a technical person, a scientist or an engineer, which I thought he was, would stoop to the Quantel claims. It had to be just a legal move to wipe out an opponent, so I gave him a private chance to prove his personal scientific integrity.

But Taylor didn't budge: "No, you didn't." So perhaps he really believed he had done it first? Hard to believe since Quantel had visited New York Tech and had my paper on paint.[19] One of my surprises at the later *Quantel* v. *Adobe* trial was the discovery that Quantel had even been in a negotiation with NYIT at one time, after my departure in 1979 but prior to their developing Paintbox.

We were naïve, Yanks in an English court. Taylor and Quantel waved around their British honors in the face of the old judge, His Lordship Falconer, presiding over his last case.

And we didn't have really solid proof. True, we had my testimony, my papers, my reputation, the corroborating testimony of Jim Blinn and Lance Williams, both colleagues of mine at NYIT. But our word, American word, apparently was insufficient. Spaceward lost the case, lost their product, and I was officially branded a liar in Falconer's decision.

The problem was more than just insufficiently convincing evidence. The legal team for Spaceward was not up to the task. They too thought they had a slam-dunk case. They were too young to know how to play the court system as their opponents did. Spaceward did have a master old barrister to argue its case, a very respected man who was of the same generation as the judge and had argued before him many times. I met the old gentleman before my time on the stand. He looked directly at me and said, "This is just the old matting technology from the early film business, isn't it?" Right. I knew we had the right man. He had argued cases in the old film technology and knew it well. But the clever Quantel legal team simply stretched out the trial until this accomplished barrister ran out of time. In England, the barristers reserve their time in advance. The Spaceward team arranged for him for one month. The month came and went, before I went on the stand. The young barrister who took his place for Spaceward was arguing one of his first cases.

This trial affected us in different ways. Jim Blinn was so scorched by the process that he refuses still to help anyone else in patent battles. I was furious at having my word brought into question and my priority stolen. But I learned a lot about the patent process and was therefore much better prepared, and much less naïve, when the call came a second time.

Adobe contacted me directly, in late 1974 or early 1975. John Warnock, cofounder of Adobe and a former colleague of mine from Xerox PARC (and another University of Utah graduate), personally asked for my help. I had just become Graphics Fellow at Microsoft, so I had to get permission from my new employers to help Adobe. This was not perceived as a problem. If anything, protecting Adobe might prove to protect Microsoft in the future. I believe it did. Microsoft provided Tom Burt, one of the company's bright litigating attorneys, to help me and protect Microsoft's interests.

Because of the unpleasant outcome of the Spaceward trial, I insisted that I would help only if Warnock promised not to settle and to have so accomplished a legal team that this seemingly simple case would not fail. He came through on both these fronts. To settle would have meant that the validity of the patent claims would not be decided, and my stature as a liar would not be overturned. As it turned out, Adobe took the case all the way, and all five patents that Quantel accused Adobe of infringing were judged invalid. The legal team that accomplished this feat—proving only non-infringement would have been a success for Adobe—was first rate. The firm was Fish & Richardson. My two principal contacts on the team were David Barkan and John Gartman.

I also urged them to find proof, if possible. This could only be the original source code, which NYIT might just possibly still have. The team found the code. So, as opposed to the Spaceward case, we had a "smoking gun" this time. The team went even further, and this shows their level of commitment to making the argument stick: They found a third party who took the original C code obtained from New York Tech (in a deal with terms I do not know) and had him recompile it to run on a modern PC running under the Windows operating system. This was possible because the original code was written in C, still a very popular programming language. All he had to do was to change the display code section to display from a simulated frame buffer in the PC memory onto the standard monitor of the PC—that is, into a window in the Windows system. This means he had to modify only some of what I call the "plumbing" code that displays the result of the serious code, which continued to work in the trial just as it did in the 1970s at NYIT. Same code!

The trial itself took place in Wilmington, Delaware, in September 1997.[74] The Quantel team waved around their Emmy, and I my technical Academy Award for "pioneering inventions in digital image compositing," awarded in 1996 to Tom Porter, Tom Duff, Ed Catmull, and me.[75] The Adobe team also had me show the jury the 1990 Siggraph Achievement Award that honored Dick Shoup and me for "seminal contributions to computer painting systems."[76] But the star of the show was Paint3.

When agreeing to testify in the trial for Adobe, I had one caveat: The month of September was out because I was going to take my wife, Zu, to Italy that month for her 50th-birthday present. (I had finally succumbed to the institution of marriage too, in 1984.) Of course, that was exactly when they slotted me to testify. So the legal team made special arrangements to accommodate Zu and me. They arranged for my testimony earlier than was originally reserved, and I pushed off my trip by three days to go to Wilmington first. So all I saw of the trial was my part, of course, and that of old colleagues Christie Barton and David Em, whose testimonies were next to mine and supportive, of course, since they were there at the beginning.

I did not get to see Dick Phillips testify as an expert. He had prepared the Adobe team exceptionally well with his background report,[42] which should be consulted as part of this history, as should other court documents for the trial. I also did not get to enjoy the Marc Levoy evidence. He had not come forward with his detailed notes about NYIT for the Spaceward case, but he had this time around. His recollections were another key ingredient in Adobe's argument. His true-color paint program preceded the Quantel claims, too.

The jury was technically unsophisticated. In fact, a high school educational level was the highest represented. This was an excellent constraint on the attorneys. They could not obfuscate as those in the Spaceward case had done (for which there was no jury, only old Judge Falconer in his red robes and white wig—quite a contrast in court scenes). The Adobe attorneys took me through a demonstration of Paint3. It was a thrill to show the original demos again, 20 years later. I was able to show soft-edged painting and digital compositing with 1970s code, which Quantel claimed to have invented in the 1980s. I even used the old line "feels like painting with ice cream" for them.

The Quantel side of course tried to imply that the code the jury was watching had all been written recently, but we successfully convinced them that this was not true. The clincher perhaps was my proof to them that the person who recompiled the code for Windows didn't even know what it did. For example, he was unaware of a secret button in the menu that I had put there when I originally wrote the code for it! I demonstrated this button (I used it to test new features then) and also explained to him what other features, "intensity painting" and "constant magnify on," did.

The point is that, although the third-party programmer had seen the code, he did not know what it did. This is consistent with recompiling as contrasted to rewriting. He had simply recompiled blindly except for the display code module. I was also able to pinpoint what time in the 1977–1979 time period that this particular version of the program was written. It was missing some of the menu items that appeared in the 1978 paper,[19] so therefore was a version predating July 1978.

The other point Quantel attorneys tried to score, since I had also testified against them in the Spaceward case, was that I pursued a personal vendetta against Quantel. This I easily refuted by telling the court how much I admired the Paintbox, that it was a beautiful implementation and that I believed it deserved its Emmy and the financial rewards from the marketplace. I just didn't buy the proposal that Quantel people had invented the underlying concepts.

Then I left for Italy, not having heard most of the arguments. The only clues that perhaps I had scored were the grins that the jurors gave me as I departed the stand, with my Academy Award that I almost left behind. I believe they would have cheered if not for the dignified courtroom setting. Much to my amazement, my old colleague Ed Catmull searched me down via telephone in a remote Tuscan village a few weeks later to give me the remarkable news that not only had the jury found Adobe innocent but all the Quantel patents invalid, and even recommended that Quantel be found guilty of defrauding the US Patent Office (the latter recommendation not followed by the judge). How sweet it was! The decision also saved Adobe several hundred million dollars in demanded royalties on Photoshop.

The trials and other tribulations have salted what has otherwise been a sweet trip through exciting times with talented colleagues and (sometimes strangely) inspired patrons inventing a technology that is now pervasive. I hope that these stories, while dutifully capturing the history of the development, also teach that mastering a new technology can be as adventure-filled as discovering a new land and that the characters on the voyage or encountered along the way are what make the effort worthwhile.

Acknowledgment

To Barbara Robertson, friend and author, who gave me writing advice and editorial assistance on early versions of the anecdotal portion of this article.

References and notes

1. A.R. Smith, *Digital Paint Systems—Historical Overview*, tech. memo 14, Microsoft Corp., Redmond, Wash., May 1997. Essentially an earlier submission to AMPAS, which used it in granting a technical Academy Award—the Scientific and Engineering Award—to R. Shoup, A.R. Smith, and T. Porter in Feb. 1998.
2. A.R. Smith, *Painting Tutorial Notes*, tech. memo 38, Lucasfilm, Marin County, Calif., Apr. 1982. Includes Digital Paint Systems Survey from *Computer Graphics World*, Apr. 1982, pp. 62-65. These were the tutorial notes for Siggraph 82.
3. B. Robertson, "Paint Systems," *Computer Graph-*

ics World, vol. 11, no. 4, Apr. 1988, pp. 62-68. A good late 1980s survey article.

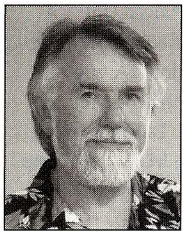
4. M. Hiltzik, *Dealers of Lightning: Xerox PARC and the Dawn of the Computer Age*, Harper Collins, New York, 1999.
5. G. Lavendel, *A Decade of Research, Xerox Palo Alto Research Center, 1970-1980*, R.R. Bowker Co., New York, 1980. An impressive record of accomplishment, presented as a collection of published papers.
6. G.E. Pake, "Research at Xerox PARC: A Founder's Assessment," *IEEE Spectrum*, vol. 22, no. 10, Oct. 1985, pp. 54-61. PARC from the management's point of view—compare with Perry and Wallich, "Inside the PARC," below.
7. T.S. Perry and P. Wallich, "Inside the PARC: The 'Information Architects,' " *IEEE Spectrum*, vol. 22, no. 10, Oct. 1985, pp. 62-75. PARC from the researchers' point of view—cf, Ref. 6. Contains an interesting graphic—the Silicon Valley family tree rooted in PARC.
8. D.K. Smith and R.C. Alexander, *Fumbling the Future: How Xerox Invented, then Ignored, the First Personal Computer*, William Morrow and Co., New York, 1988.
9. I believe that one of the reasons Pixar (and related endeavors) has been so successful is that we fully pursued both halves of the computer "picturing" world with equal tenacity.
10. It should be noted that the two worlds intersect in some places, the most notable being in so-called texture mapping, where a digital image, often painted, is "wrapped" onto a 3D geometric surface to give it the appearance of reality. A very hot topic of current research is so-called image-based rendering, which intimately combines the two realms.
11. E. Catmull, T. Porter, T. Duff, and I received a technical Academy Award, the Scientific and Engineering Award of AMPAS in Feb. 1996 for inventing the alpha channel.
12. J.E. Miller, personal communication, Bell Labs, Murray Hill, N.J., July 1978.
13. W.J. Kubitz and W.J. Poppelbaum, "The Tricolor Cartograph: A Display System with Automatic Coloring Capabilities," *Information Display*, vol. 6, Nov.-Dec. 1969, pp. 76-79.
14. R.G. Shoup, *Old Software on the Color Video System*, Xerox PARC memo, 4 Feb. 1975. Contains a complete manual for the system.
15. R.G. Shoup, *SUPERPAINT Program*, Xerox PARC memo, 5 Feb. 1975. First use of the term SUPERPAINT (that I write as SuperPaint here).
16. R.G. Shoup, "Some Experiments in Television Graphics and Animation Using a Digital Image Memory," SMPTE 13th Television Conf., San Francisco, Feb. 1979. Published in *Digital Video*, vol. 2, Soc. of Motion Picture and Television Engineers, Scarsdale, N.Y., Mar. 1979, pp. 88-98. Republished in *Datamation*, vol. 25, no. 5, May 1979, pp. 150-156.
17. It had a fixed palette of $6 \times 6 \times 6$ colors displayed at the bottom of the screen, variable-sized rectangular brushes, and could save or restore images to or from files; Jim Blinn, personal communication, Microsoft Corp., Redmond, Wash., May 1997; also see R. Leavitt, *Artist and Computer*. Cover photo shows Duane Palyka painting digitally using Jim Blinn's Crayon program.
18. The night the E&S hardware first ran, in fact (Jim Blinn, personal communication, Microsoft Corp., Redmond, Wash., May 1997).
19. A.R. Smith, *Paint*, tech. memo 7, NYIT, 20 July 1978. Also issued as tutorial notes at Siggraphs 78–82. Reprinted in *Tutorial: Computer Graphics*, 2nd ed., J.C. Beatty and K.S. Booth, eds., IEEE Computer Soc. Press, Los Alamitos, Calif., 1982, pp. 501-515. Also reprinted in *Seminal Graphics: Pioneering Efforts that Shaped the Field*, R. Wolfe, ed., a publication of ACM Siggraph, 1998, pp. 427-441. See Appendix C in particular. Contains a brief history of paint programs, including pre-Shoup experiments. First public mention of RGB paint, airbrushing, and so forth.
20. See R. Leavitt, *Artist and Computer*.
21. J. Blinn, personal communication, Microsoft Corp., Redmond, Wash., May 1997.
22. Jim Blinn recalls that he had the program with him when he returned from a summer at NYIT (J. Blinn, personal communication, Microsoft Corp., Redmond, Wash., May 1997).
23. G. Stern, personal communication, Spitz Inc., Chadds Ford, Pa., May 1997.
24. I have a dated notebook, from December 1976, covering the Ampex installation.
25. J. Blinn, *Jim Blinn's Corner: Dirty Pixels*, Morgan Kaufmann, San Francisco, Calif., 1998, pp. 131-135. In particular, see chapter 12, *NYIT: How I Spent My Summer Vacation, 1976*. This is his Sept. 1993 column from *IEEE Computer Graphics & Applications*.
26. M. Levoy, "Frame Buffer Configurations for Paint Programs," tutorial for Siggraph 79, 1979. Revised in May 1980 and reissued as tutorial notes for Siggraphs 80–82.
27. N. Negroponte, "Return of the Sunday Painter," *The Computer Age: A Twenty-Year View*, M.L. Dertouzos and J. Moses, eds., MIT Press, Boston, Mass., 1979 (first paperback edition, 1980), pp. 21-37. Features one of my earliest color reproductions, *Egg on Toast*, at a higher resolution than most computer graphics images at that time. It would not fit into a frame buffer so I had to execute it by frame-buffer-size pieces.
28. N. Negroponte, "Raster Scan Approaches to

Computer Graphics," *Computers and Graphics*, vol. 2, no. 3, 1977, pp. 179-193.

29. The earliest dated documentation I have for this code is dated 13 January 1978. Ed was preparing for Siggraph 78. Siggraph typically has a paper-due date of early January of the corresponding year, so this is probably about when the invention actually occurred, although it might have happened in December 1977, to avoid the last-minute crunch against the paper deadline. I was also preparing a paper for Siggraph 78. The date on the submission is 6 January 1978, and the code I used to generate figures for the paper is dated 28 December 1977.
30. A.R. Smith, *Alpha and the History of Digital Compositing*, tech. memo 7, Microsoft Corp., Redmond, Wash., Aug. 1995. Essentially an earlier submission to AMPAS, which used it in granting a technical Academy Award—the Scientific and Engineering Award—to A.R. Smith, E. Catmull, T. Porter, and T. Duff in Feb. 1996.
31. A.R. Smith, "Table Paint," *Tutorial Notes*, Jet Propulsion Laboratory, California Inst. Technology, Pasadena, Calif., Oct. 1979. Also issued as tutorial notes at Siggraphs in 1980 and 1981.
32. It had a specialized color map structure that divided the available bit depth into i bits for border colors, j bits for infill colors, and k bits for the antialiased transitions between these colors, $i + j + k = 8$; Marc Levoy, personal communication, Stanford Univ., Palo Alto, Calif., May 1997.
33. M. Levoy, personal communication, Stanford Univ., Palo Alto, Calif., May 1997.
34. The first commercial paint product. Shown on show floor of National Association of Broadcasters (NAB) convention in March or April 1980. Quantel Paintbox was shown in a suite (see 1980s entry in Table 1 in the main text).
35. T. Porter, *NAB Trip Report*, Lucasfilm memo, San Rafael, Calif., 17 Apr. 1981.
36. A.R. Smith, "Special Effects for Star Trek II: The Genesis Demo, Instant Evolution with Computer Graphics," *American Cinematographer*, vol. 63, no. 10, Oct. 1982, pp. 1038-1039 and 1048-1050. Mentions Tom Porter and his paint program.
37. T. Porter, *Picture Coding and the Paint System*, Lucasfilm memo, San Rafael, Calif., 13 Feb. 1981.
38. T. Porter, *Picture Handling Using Staging Areas*, Lucasfilm memo, San Rafael, Calif., 17 Feb. 1981.
39. T. Porter, *The Paint System Design, 1st Pass Technical Memo*, Lucasfilm memo, San Rafael, Calif., 18 Mar. 1981.
40. John still works for Lucasfilm. I understand that Thomas brought a background in image processing to bear on the product as well.
41. I do not know the original name for Photoshop.
42. R.L. Phillips, "Computer Graphics in Court: The Adobe/Quantel Case," *Computer Graphics*, vol. 32, no. 3, Aug. 1998; also available online at http://www.siggraph.org/publications/newsletter/v32n3/contributions/phillips.html.
43. Jim Blinn recalls that Garland's program supplanted his own at Utah, after Garland's return from NYIT, because of a larger feature set; J.Blinn, personal communication, Microsoft, May 1997. Both Garland and Jim recall that the two programs were separate developments; J. Blinn, personal communication, Microsoft, May 1997; also, G. Stern, personal communication, Spitz Inc., Chadds Ford, Pa., May 1997.
44. J. Rosebush, *Computer Animation: An Historical Survey*, thesis, Syracuse Univ., School of Public Comm., Syracuse, N.Y., Jan. 1979. "Prepared to fulfill the requirements of third comprehensive examination" (This came from the title page of the thesis). This earlier history was never published. Contains a frame buffer history. Judson did, however, contribute significantly as a writer to the recent documentary video, *The Story of Computer Graphics*, premiered at the 26th Siggraph, Los Angeles, Calif., 1999.
45. I installed Paint at Ampex in December 1976, working with Junaid Sheikh (who would suffer from the Quantel patents in a later entrepreneurial effort).
46. In 1977, Ephraim combined frame-buffer-to-frame-buffer and file-to-frame-buffer copy and composite routines into a general-purpose Copy program. In 1978, he added tablet control to this program to get painting, which inherited 8-bit or 24-bit mode, antialiasing, airbrushing, and so on from the earlier Copy program. In 1978–1979, he fleshed this out into his ept painting system; E. Cohen, personal communication, R/Greenberg Associates, New York, May 1997.
47. I had also noted this. See Smith, "Table Paint," tutorial notes. Dick Shoup based his Aurora products on this concept, too; R.G. Shoup, personal communication, Interval Research, Palo Alto, Calif., May 1997.
48. Dick actually called his environment the Color Video System, including a paint program called Paint; R.G. Shoup, *Simple-Minded Animation on the Color Video System*, Xerox PARC memo, 9 Aug. 1974. Dick formally proposes to hire me in this memo, reporting on several animations that I had already created on the system. Also, R.G. Shoup, *Old Software on the Color Video System*, Xerox PARC memo, 4 Feb. 1975. Contains a complete manual for the system. He added the SuperPaint program to the system in Feb. 1975; R.G. Shoup, *SUPERPAINT Program*, Xerox PARC memo, 5 Feb. 1975. First use of the term SuperPaint (I refer to the entire system as SuperPaint for convenience.) The complete system is in the

permanent collection of the Computer Museum and is still (occasionally) functional.
49. T. Porter, personal communication, Pixar, Richmond, Calif., May 1997.
50. M. Gardner, *Wheels, Life and Other Mathematical Amusements,* W.H. Freeman, New York, 1983. Contains reprints of all his "Game of Life" writings from *Scientific American.*
51. R.C. Minnick, J.C. Huang, R.G. Shoup, and A.R. Smith, "Cellular Logic," *Hardware, Software, Firmware Trade-offs: Proc. 1971 IEEE Int'l Computer Soc. Conf.,* IEEE Press, New York, 1971, pp. 25-30.
52. J. von Neumann, *Theory of Self-Reproducing Automata,* A.W. Burks, ed., Univ.of Illinois Press, Urbana, 1968. Completed after von Neumann's death.
53. This edition of the book never reached final publication although I was paid for my contribution, which eventually became my "Introduction and Survey of Polyautomata Theory," article published in *Automata, Languages, Development,* A. Lindenmayer and G. Rozenberg, eds., North-Holland Publishing Co., New York, 1976, pp. 405-422. This served as the proceedings of an international conference on "Formal Languages, Automata, and Development," Noordwijkerhout, the Netherlands, Apr. 1975. Artificial Life 0. A complete survey of CA (cellular automata) theory. For nontrivial, self-reproducing CA, see A.R. Smith, "Simple Nontrivial Self-Reproducing Machines," *Artificial Life II,* C.G. Langston et al., eds., Addison-Wesley, Reading, Mass., 1992, pp. 709-725. These were the proceedings of the "Workshop on Artificial Life" held Feb. 1990, Santa Fe, N.M. Based on my 1969 PhD dissertation work.
54. R.G. Shoup, *Simple-Minded Animation on the Color Video System,* Xerox PARC memo, 9 Aug. 1974.
55. Short for picture element. Others, notably IBM and AT&T, tried *pel,* a term never adopted by the computer graphics community. A pixel is commonly three or four bytes. Frame buffers and digitized images are often measured in mega- or gigapixels. Three bytes, three times eight bits, can hold $2^{24} = 16,777,216$ values, or colors in the case of a pixel—that's 16 megacolors.
56. The Shoup frame buffer (picture memory) is only conceptually 2D. For economic reasons, he implemented it with shift-register memory chips, which means that a given pixel has to be shifted, one pixel at a time, out of the chips to get access to it. It is organized as a single, long shift register with counters that keep track of what line and pixel you are on. The now-popular random access memory (RAM) chip, allowing direct access to a given pixel, became cheap enough only later. In either case, software techniques disguise the underlying hardware and make it appear as a 2D array. The first dense-enough RAM chips (Intel 1103) available when Shoup made his design decision were twice as expensive and half the density of the shift-register chips. The second frame buffer he planned to build, but never did, would have been RAM based.
57. If you look closely at a color TV screen, you can see the tiny red, green, and blue phosphor dots that glow at different intensities to make colored light. The television monitor mentioned is one used in a broadcast studio rather than at home, although a home set works as well. The clean TV signal at a studio deteriorates by the time it reaches a home receiver. The Shoup frame buffer generated "broadcast quality" video adhering to the NTSC (National Television Standards Committee) standards for TV in the US.
58. SuperPaint and most paint programs intended for use by professionals actually employ a stylus on an electronic tablet instead of a mouse, use of which is likened to painting with a bar of soap.
59. This does not mean that each phosphor dot or RGB (red, green, blue) triad of dots corresponds exactly to a single pixel. To anyone who has changed the screen resolution of a computer monitor from, say, 640×480 to 1024×768, this should be obvious. The number of screen phosphors remains constant while the number of pixels displayed increases a great deal.
60. Arguably the first in the sense that the Shoup frame buffer was the first serious one used for interactive graphics, there being earlier experiments—for example, a 3-bit (8-color) frame buffer at Bell Labs (see Appendix C of *Seminal Graphics: Pioneering Efforts that Shaped the Field,* R. Wolfe, ed., ACM Press, New York, 1998). There were undoubtedly image memories in use by the military at this time, but we have little information about them, such as whether they were interactive. Some historians and patent attorneys draw a distinction between Shoup's picture memory, based on shift-register chips, and the later E&S frame buffer, built from RAM chips. This is a minor distinction. From a programmer's viewpoint, either can be addressed randomly, although the implementation is certainly slower and clumsier for shift-register memory than RAM.
61. HSV is later institutionalized in the PostScript language—renamed HSB (B for brightness)—at the company Adobe Systems by its founder, John Warnock, another graduate of PARC (and the University of Utah before that); and see A.R. Smith, "Color Gamut Transform Pairs," *Computer Graphics,* vol. 12, no. 3, Aug. 1978, pp. 12-19 (Siggraph 78 Conf. Proc.); reprinted in *Tutorial: Computer Graphics,* second edition, J.C. Beatty and K.S. Booth, eds., IEEE Computer Soc. Press,

Los Alamitos, Calif., 1982, pp. 376-383. Discusses the HSV algorithm.
62. A.R. Smith, *Vidbits* (video), Xerox PARC, 1974. Exhibited at the Museum of Modern Art, New York, 1975, and on WNET television show *VTR*, New York, 1975.
63. I'm particularly tickled by their *Cadillac Ranch* "sculpture" outside Amarillo, Texas, about 100 miles from my hometown. It consists of about ten of the biggest tail-fin Cadillacs ever manufactured in Detroit, of successive years, buried nose first in a row in the middle of a wheat field.
64. Newman's principal contribution at the time was the first textbook, with Robert Sproull, on computer graphics; see W.M. Newman and R.F. Sproull, *Principles of Interactive Computer Graphics*, McGraw-Hill, San Francisco, 1973. William is the son of famous English computer pioneer Max Newman, a fact I didn't know until 1999.
65. S. Brand, "Fanatic Life and Symbolic Death among the Computer Bums," *Rolling Stone*, 7 Dec. 1972. Describes the climactic culture clash between the California and New York branches of Xerox over PARC.
66. From Shoup quote in T.S. Perry and P. Wallich, "Inside the PARC: The 'Information Architects,'" *IEEE Spectrum*.
67. J.T. Kajiya, I.E. Sutherland, and E.C. Cheadle, "A Random-Access Frame Buffer," *Proc. IEEE Conf. Computer Graphics, Pattern Recognition, and Data Structure*, IEEE Press, New York, 1975, pp. 1-6. Describes the E&S frame buffer.
68. A. Schure, quoted in "N.Y.I.T. Puts Computers to Work for TV," by L. Gartel and J.L. Streich, *Millimeter*, June 1981. The source of the famous quote.
69. I became familiar with BCPL at Xerox PARC and used it briefly at NYIT. The C language is a direct descendant of BCPL, the "B" to its "C". The "CPL" part evidently stands for Cambridge [University] Programming Language.
70. A lovely estate nearby NYIT with several houses for various members of the McGrath family, in-laws of David Rockefeller. David DiFrancesco and I, and later Garland Stern, shared the chauffeur's quarters over another four-car garage.
71. R. Adams, *Watership Down*, Avon Books, New York, 1975. From a review in *The London Times*: "I announce, with trembling pleasure, the appearance of a great story."
72. I did implement a crude approximation to this kind of airbrushing, too, with a brush of random dots clustered more densely in the center. See D. Em, *The Art of David Em: 100 Computer Paintings*, text by D.A. Ross and D. Em, Harry N. Abrams Inc., New York, 1988.
73. E. Emshwiller et al., *Sunstone*, video, created at NYIT, 1979.
74. *Quantel Limited, Plaintiff*, v. *Adobe Systems Incorporated, Defendant*, in the US District Court for the District of Delaware, before the Honorable Roderick R. McKelvie, Courtroom 4A, J. Caleb Boggs Federal Bldg., 844 King St., Wilmington, Del., Sept 1997. I testified on 12 Sept. 1997.
75. I had not yet received my second technical Academy Award, with Dick Shoup and Tom Porter, for "pioneering inventions in digital paint systems," awarded in 1998 (see Ref. 1).
76. R.G. Shoup and A.R. Smith, "1990 ACM Siggraph Awards: Computer Graphics Achievement Award," *Computer Graphics*, vol. 24, no. 4, Aug. 1990, pp. 17-18 (Siggraph 90 proceedings).



Alvy Ray Smith recently retired from Microsoft, where he was its first Graphics Fellow, to devote time to digital photography. He invented, directed, originated, or was otherwise instrumental in the following developments: first full-color paint program, HSV color model, alpha channel and image sprites, Genesis demo in *Star Trek II: The Wrath of Khan*, Academy-Award-winning Disney animation production system CAPS, and the Visible Human Project at the National Library of Medicine. He is the co-founder of Pixar Animation Studios.

Smith has a PhD from Stanford University and an honorary doctorate from New Mexico State University. He received two technical Academy Awards for the alpha-channel concept and for digital paint systems.

Readers may contact Alvy Ray Smith at http://alvyray.com.

SuperPaint: An Early Frame Buffer Graphics System

Richard Shoup

The union of digital computing and video was made possible by high-density integrated memories in the early 1970s. This new technology led Xerox PARC researchers to develop SuperPaint, the most famous of the early pixel-based frame buffer systems and one that portended a significant part of the future of computer graphics in television by illustrating the Pioneer spacecraft missions to Venus and Saturn.

In the early 1970s, integrated-circuit memories of significant size (1 Kbit and greater) became available. These chips made it practical for the first time to build a digital memory system—specifically, what came to be known as a frame buffer—containing enough bits to hold a standard video image. Minimally, a video image contained 640 pixels horizontally, 480 pixels vertically, and 8 bits in depth, for a total of 307,200 bytes.[1] This development in turn opened the door to the now-familiar world of pixel graphics and the use of digital technology in a wide range of imaging applications.

SuperPaint, one of the first pixel-based frame buffer systems, originated in the Computer Science Lab at the Xerox Palo Alto Research Center (PARC).[2] I initiated this project at PARC in 1972, designed most of the hardware for the system, fabricated it with help from a number of other lab members in early 1973, and coaxed it into displaying its first picture in early April 1973. I also wrote most of the software for the system over a period of several years. Alvy Ray Smith, Bob Flegal, and Patrick Baudelaire also wrote significant routines and made other contributions to the project.

Of course, the advantages of raster scan displays over line-drawing calligraphic displays were well known at the time. But from a larger perspective, we realized that the development of SuperPaint signaled the beginning of the synergy of two of the most powerful and pervasive technologies ever invented: digital computing and video or television.

At least two major problems had to be solved:

- making a digital frame buffer that operated at video rates and was fully compatible with video displays, cameras, recorders and other television technology, and
- finding ways for a human user to create and interact with full-color 2D images and, later, animations.

The former required some clever engineering. For the latter, we borrowed heavily from well-developed painting, graphic arts, and animation techniques in traditional media.

Hardware description

The SuperPaint frame-buffer memory system consisted of 16 printed circuit cards filled with 2-Kbit shift register chips (Intel 2401)—altogether, $640 \times 486 \times 8$ bits. We used shift register memories rather than the just-introduced 1-Kbit dynamic RAM parts (Intel 1103) because of their significantly higher bit density (more than a factor of 2), lower cost, and established reliability.

The entire memory recirculated as one long 307,200-pixel shift register, which was shifted synchronously with the video display at standard television scan rates. Thus, any particular pixel in the memory could be accessed only when the desired scan line and pixel time rolled around—an entire frame time of 33 ms in the worst case. (This was not as disadvantageous as it sounds, as will be explained below.) The pixel clock was 83 ns, which at standard video scanning rates produces square pixels on the display. By adopting standard video parameters, we could use readily available television studio monitors, record the output signal on available videotape or videodisk recorders, and transmit it on existing closed-circuit or broadcast television channels.

This synchronous recirculation structure also

1058-6180/01/$10.00 © 2001 IEEE

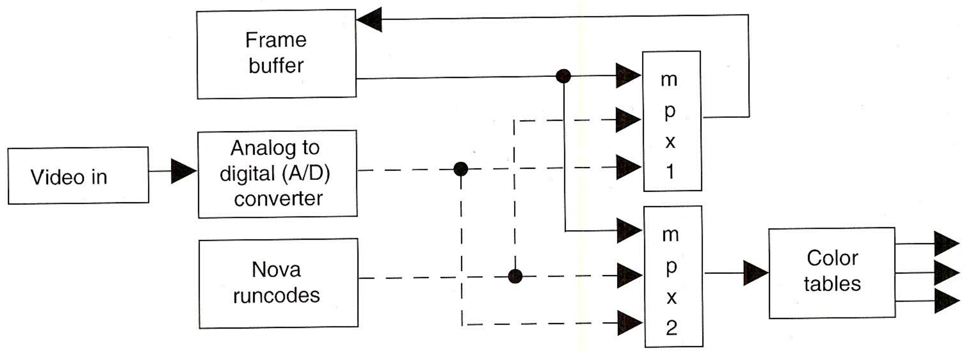


Figure 1. Block diagram of the SuperPaint memory system. Solid lines show the paths used during static display. Courtesy of Richard Shoup.

permitted writing into the frame buffer in real time from any external standard video source such as a camera or tape recorder. Figure 1 shows the data paths of the frame buffer including the shift-register memory, the computer interface (discussed later on), and video inputs via a real-time analog-to-digital (A/D) converter.

Two multiplexers, operated on a pixel-by-pixel basis, controlled the flow of pixels. Multiplexer 1 controlled data flowing into the image memory, taking its input from either the computer, the video input stream, or the memory for recirculation. In addition to computer-controlled switching, several experimental circuits were added from time to time to implement other pixel-rate functions, such as area filling and various kinds of pixel value replacement.

Multiplexer 2 controlled data sent out to the color lookup tables (256 × 8 × 3 static RAMs producing separate red, green, and blue streams) and thus to the digital-to-analog converters (D/As) and out to the video displays. This multiplexer could also switch at full pixel rate among the three input streams, thus allowing pixel-by-pixel control of overlay effects such as brush-over-canvas and video-input-over-canvas, without storing the result of this pixel stream merge into memory.

Eventually, a second set of color tables and D/As were added to the system so that separate menu and canvas displays could be supported, each driven by a selection from the 8 bits in each pixel. Typically, 4 bits were allocated to each of the two images. Then, judicious setting of the color tables permitted several display modes, including separate menu and canvas images on two monitors, switching between menu and canvas on a single monitor and a transparent menu overlaid on canvas on a single monitor.

Using the color tables, one could easily create and change various palettes of displayed colors to suit the artist's intent. By changing color-table definitions rapidly under program control, we also implemented a limited but surprisingly useful array of animation functions and effects.[3] In particular, the program could change the color of a large number of pixels on the screen much more quickly than writing new values for all of them into memory.

A commercially available analog RGB-to-NTSC (red-green-blue–to–National Television Standards Committee) encoder provided standard composite video output for broadcast or recording to videotape. The system could also be synchronized to an incoming video signal by means of its own sync generator—an essential feature for any real-world production studio or on-air operation.[4] An 8-bit real-time A/D converter provided monochrome video input from a video camera or videotape recorder.

A Data General Nova 800 minicomputer served as the system controller, which we programmed using the BCPL language (a predecessor to C) and some assembly code. Two Diablo 31 disk drives were attached, with removable 2-Mbyte cartridge disks.

From the Nova controller, a run-length-encoded read/write interface permitted specification of a scan line number (y coordinate), a pixel number (x coordinate), and a run length of up to 255 pixels. Although a controversial choice at the time, this shift-register-based architecture and interface actually performed faster on most typical tasks—even random line drawing—than the PARC Alto machine built at the same time using the latest 1-Kbit DRAMs. By keeping a sorted list of pixel runs waiting to be written to the frame buffer, the software could efficiently utilize the runcode hardware as scan lines became available in the raster scan recirculation. With standard solid-color paint brushes, for example, only one run per scan line

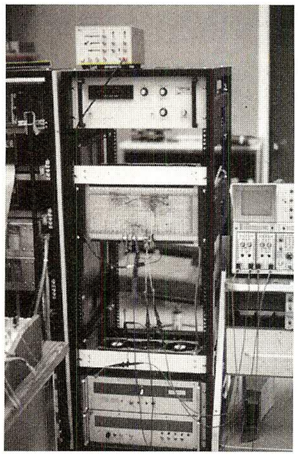


Figure 2. SuperPaint frame-buffer system rack. Courtesy of Richard Shoup.

was needed regardless of brush size, and painting could always take place at the full 30 frames/second rate.

At the top of the 5-foot equipment rack shown in Figure 2 is the 8-bit video digitizer (a full 6 inches of rack space and a cost of more than $12,000 at the time), a set of fans, and the SuperPaint frame-buffer memory card cage. The space just below would soon be occupied with another card cage containing the control cards, color lookup tables, D/As, and the mini-computer interface. A second set of fans and several power supplies are at the bottom of the rack.

The shift-register memory card (16 copies) was a printed circuit board, while all control and data-path circuitry was implemented using Augat-style wirewrapped prototype cards. Originally, there was to have been an additional cage of memory cards providing a second 8-bit frame buffer in the center of the rack, but this was deemed too expensive and was never added. A second 5-foot equipment rack held the Data General Nova 800 minicomputer, two Diablo 31 disk drives, a cage of miscellaneous interface cards, and additional power supplies.

Figure 3a shows the frame-buffer backplane; Figure 3b is the first picture captured in the SuperPaint frame buffer when it came to life in April 1973. Input was from a monochrome video camera pointed at the author. Judging from the clipleads on the backpanel, the hardware still had a few bugs at this time. To trigger the capture of this image, I pulled a cliplead off the backpanel using my knees! The card I'm holding says, "It works! (sort of)." The interface to the Nova 800 controller had not been even plugged into the system yet at the time this image was captured. To avoid losing this image, I was forced to plug the interface card in and debug it and its software with the power still on. Later, I was able to clean up some of the missing bits in the image with a small heuristic program.

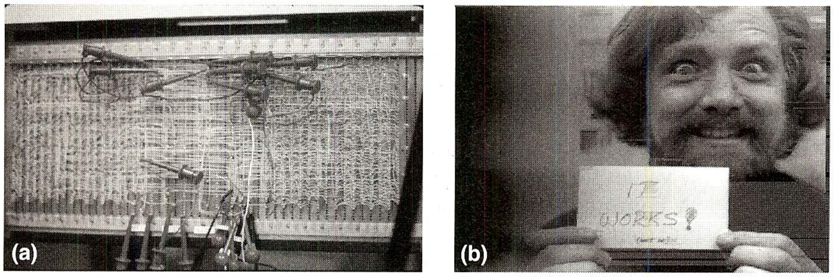


Figure 3. (a) The SuperPaint frame buffer backplane, with clipleads. (b) First image: "It works! (sort of)." Courtesy of Richard Shoup.

Software description

Before any paint programs existed for the SuperPaint frame buffer, I wrote a "jaggie removal" program to generate smooth straight lines at any angle, without the jagged stairsteps always associated with raster computer displays at that time. Today, this well-known technique is called antialiasing. The algorithm, written in Nova assembly language, imitated a standard video camera by computing an approximation to the intersection of the desired line with an imaginary circular gaussian scanning spot. I was in a hurry to implement this technique because I had already submitted a paper on it.[5] Fortunately, the results were even better than anticipated, as seen in Figure 4.

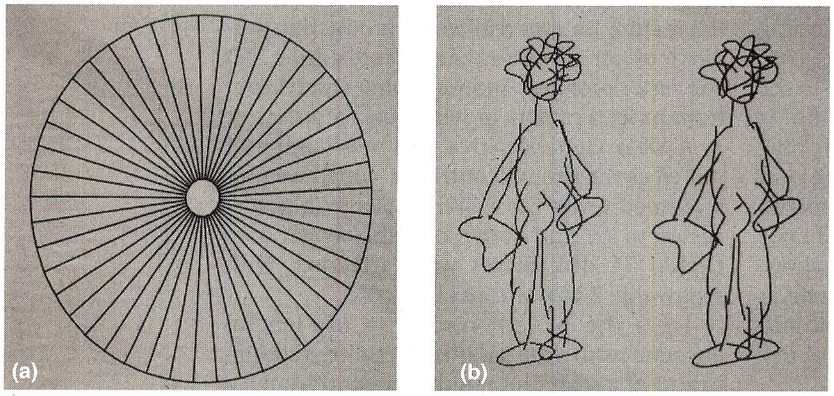


Figure 4. (a) Antialiased wagon wheel; (b) Original and smoothed drawings: *Two Guys* by Bob Flegal (1973). Courtesy of Richard Shoup.

Over the next several years, a lot of experimentation went on concerning how to use the pen and tablet for both painting and for control functions, how best to control and select brushes, and how to edit pictures and color spaces. Other experiments dealt with dynamic interaction and with the system's color-table animation capabilities. Details of some of the system's operation and applications have been published elsewhere.[6,7]

After many versions, the menu evolved to include the functions shown in Figure 5 such as Paint (the paint brushes on the left and right); Shrink 2X, Expand 2X, Move, and Copy (center-left section); Store and Load (bottom

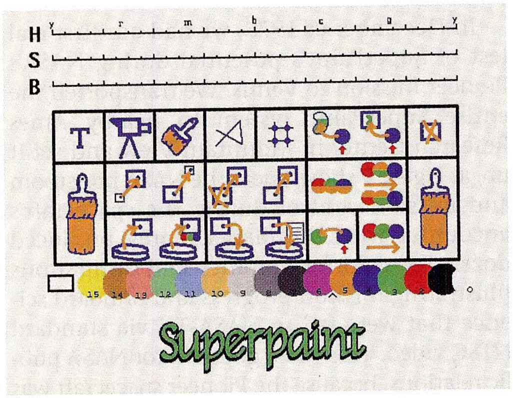


Figure 5. *SuperPaint menu* by Richard Shoup (1977).

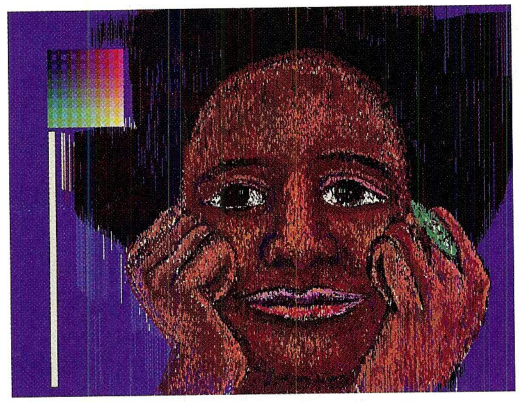


Figure 6. *Black Girl* by Fritz Fisher (1973). Courtesy of Richard Shoup.

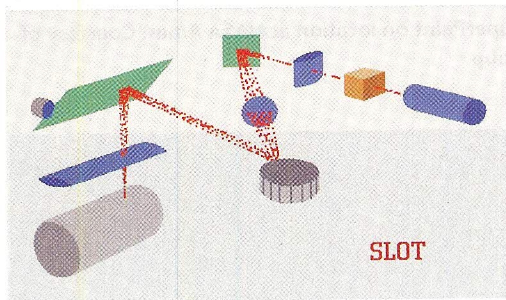


Figure 7. *SLOT animation* by Bill Bowman (1974). Courtesy of Richard Shoup.

Figure 8. "Over Easy" graphics by Damon Rarey (1977). Courtesy of Richard Shoup.

center); Text, Video In, Make Brush, Draw Lines, Gridding, Area Fill (top row); and various forms of color table animation (right-center). Some functions were qualified by a rectangular area defined by dragging two corner-shaped markers. During use, currently available brush shapes were displayed along the bottom of the menu in place of the SuperPaint title shown in this figure.

Most useful were the animation functions and effects made possible by the color lookup tables. These included Color Cycle (for flowing motions), Reveal (for progressive development of an illustration), and Step (for stepwise motion). These are detailed elsewhere.[3]

Applications and anecdotes

The following images and comments will serve to illustrate some of the early experiments and applications of SuperPaint.

For the first paint program, the runcode hardware wasn't working yet, so for a short period all paint brushes were one pixel wide (and any height)! Artist/mathematician Fritz Fisher had taken a job as a night guard at our building in order to obtain access to the system, and images like the one in Figure 6 often greeted us in the mornings.

Figure 7 was the first full animation done on the system. It was created one frame at a time by artist/designer Bill Bowman using a separate real-time video disk system (300 frames of 4 analog video channels) attached to the RGB outputs of SuperPaint. Unfortunately, this disk had a habit of crashing its heads on a regular basis, despite heroic efforts on the manufacturer's part. We were lucky to complete this animation of the Scanned Laser Output Terminal (SLOT), the first laser printer built by Gary Starkweather at PARC.

One of the first applications of computer graphics in broadcast television was the use of SuperPaint in the PBS television series "Over Easy," produced in the late 1970s at KQED in San Francisco. The producers of the show wanted some innovative graphics and animation for their new series and had heard about some interesting graphics work at Xerox PARC. They came to visit, and thus Damon Rarey and I first met and became long-time collaborators. Damon produced many graphics and eye-catching animations, such as those in Figure 8, that were used regularly in the "Over Easy" show—at a cost far below conventional methods of the time.

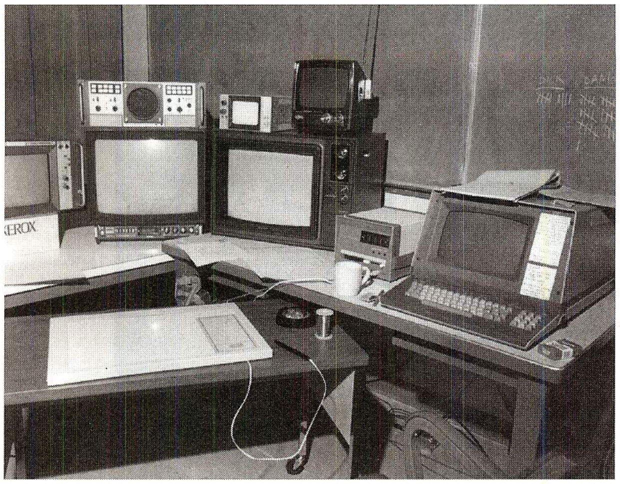


Figure 9. SuperPaint on location at NASA Ames. Courtesy of Richard Shoup.

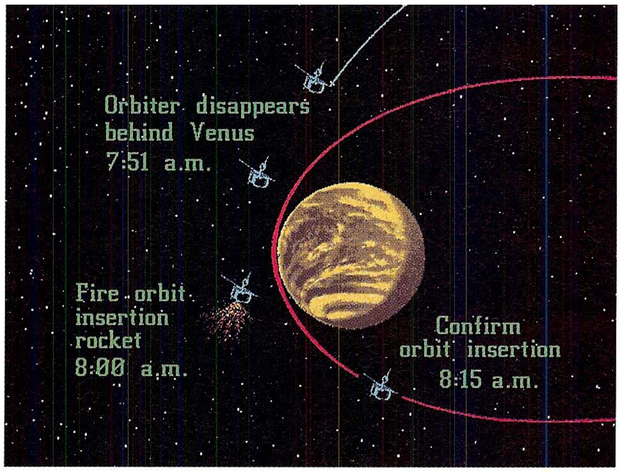


Figure 10. *Pioneer Venus orbit insertion* by Damon Rarey (1978). Courtesy of Richard Shoup.

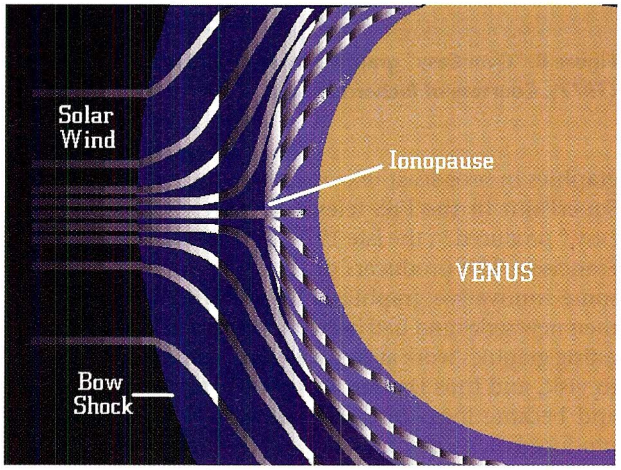


Figure 11. *Solar Wind* by Damon Rarey (1978). Courtesy of Richard Shoup.

In December of 1978, we had our first real test of SuperPaint's potential during NASA's Pioneer mission to Venus. We transported the entire SuperPaint system to nearby Ames Research Center in Mountain View and set it up, as Figure 9 shows, near the NASA pressroom and mission control during the spacecraft's encounter with the planet. Rarey produced dozens of colorful graphics and animations illustrating mission progress and onboard science that were fed to NASA-TV via standard NTSC video. This was a godsend for NASA public relations, because the Pioneer spacecraft was not equipped with an imaging camera. The chalkboard in Figure 9 shows a tally of the seemingly unending series of demonstrations we gave to mission scientists, press, and numerous other visitors from around the world.

The color-table-Reveal animation in Figure 10 showed insertion of the Pioneer Venus spacecraft into orbit around Venus during the planetary encounter. This animation was used on the live NASA-TV program originating from Ames during the mission. Unfortunately, the director cut to it just at the moment the Mission Control scientists were cheering the orbit's confirmation, and some news organizations complained mightily that we had stepped on the live video with some new-fangled computer graphic.

The color-table-Cycle animation in Figure 11 showed the flow of particles emanating from the sun as they passed around Venus, which the Pioneer probes were designed to detect and measure. The NASA scientists, and especially the public relations people, were overjoyed to have a way to explain the Pioneer mission's exciting science results to the public.

Although most of our NASA graphics were serious science illustrations, occasionally we tried to be simply entertaining. After surviving the plunge through the atmosphere of Venus, one of the Pioneer probes mysteriously stopped transmitting just after landing on the hot Venusian surface. Damon and I thought perhaps we knew what had become of the probe, as we illustrated in Figure 12. The hot gasses shown in the animated Venusian sky in this figure flowed continuously through the use of color table cycling.

A year after Pioneer Venus, in 1979, we were invited to bring SuperPaint back to NASA Ames again to play a similar role in a subsequent Pioneer spacecraft mission to the planet Saturn. During both of these planetary encounters, our animations were seen on nearly every major television network around the world. On one occasion, Rarey did a live interview and SuperPaint demonstration on Italian network television from our location at Ames.

In late 1979, Rarey and I left Xerox and founded the company Aurora Systems in an area of San Francisco that has since become known as Multimedia Gulch. At Aurora, we developed several further generations of painting and animation systems, manufactured and sold these systems, and supplied computer graphic services for the video broadcast and production markets throughout the 1980s.

Current status

At present, the complete SuperPaint system resides in the permanent collection of the Computer Museum History Center at Moffett Field in Mountain View, California. On occasion, it is still operational.[8]

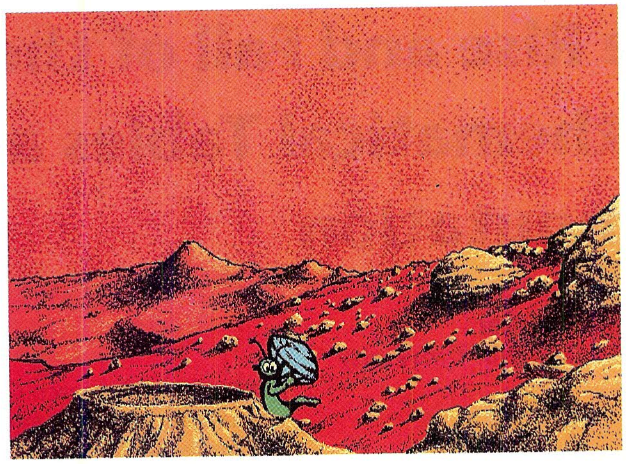


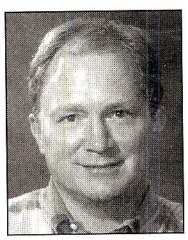
Figure 12. *Green Guy* by Damon Rarey (1978). Courtesy of Richard Shoup.

Acknowledgments

This work was entirely supported by the Xerox Corporation at its Palo Alto Research Center. Inspirational predecessors to SuperPaint included the Alto bitmap display by Thacker and others at Xerox PARC (Feb. 1973), CharGen Paint by Kay and Purcell at Xerox PARC (1972), the first commercially available frame buffers by Ramtek (1972-73), the Paint program by Noll and Miller at Bell Labs (1969), the Tri-Color Cartograph analog-disk-based paint system by Kubitz and Poppelbaum at the University of Illinois (1968), and especially the Scanimate real-time video animation system by Harrison and Honey at Computer Image (circa 1967).

References

1. SuperPaint was not the first such frame buffer nor the first digital paint program. For definitions and additional background relating to early frame buffers and painting systems, see the accompanying article in this issue by Alvy Ray Smith, "Digital Paint Systems: An Anecdotal and Historical Overview."
2. M. Hiltzik, *Dealers of Lightning: Xerox PARC and the Dawn of the Computer Age*, HarperBusiness, New York, 1999.
3. R. Shoup, "Simple Animation by Changing Color Definitions," *Proc. ACM SIGGRAPH Conf. Computer Graphics and Interactive Techniques*, ACM Press, New York, 1979.
4. To our knowledge, SuperPaint was the first fully video-compatible pixel-based computer graphics system.
5. R. Shoup, "Some Quantization Effects in Digitally-Generated Pictures," *Soc. for Information Display Symp. Digest*, Soc. for Information Display, San Jose, Calif., 1973.
6. R. Shoup, "Menu-Driven User Interfaces for Videographics," presented at the SMPTE 17th Television Conf., San Francisco, Calif., Feb. 1983. Published in *SMPTE J.* and *SMPTE Video Pictures of the Future*, Soc. of Motion Picture and Television Engineers, Scarsdale, N.Y., June 1983.
7. R.G. Shoup, "Some Experiments in Television Graphics and Animation Using a Digital Image Memory," SMPTE 13th Television Conf., San Francisco, Feb. 1979. Published in *Digital Video*, vol. 2, Soc. of Motion Picture and Television Engineers, Scarsdale, N.Y., Mar 1979, pp. 88-98. Republished in *Datamation*, May 1979.
8. If you would like to assist in getting the SuperPaint Nova 800 in full working order again, or can otherwise read Diablo-31-style 8-sector packs written by a Decision disk controller circa 1972, please contact the author at rgshoup@rgshoup.com.



Richard G. Shoup is cofounder and president of the Boundary Institute, a nonprofit research center for the study of leading-edge physics and mathematics (http://www.boundaryinstitute.org). Shoup obtained a BSEE and a PhD in computer science at Carnegie Mellon University. His doctoral dissertation (1970) was one of the first to explore field-programmable gate arrays and the idea of restructurable computing. He was awarded an Emmy in 1983 by the National Academy of Television Arts and Sciences for his graphics work at Xerox PARC in the 1970s. In 1998, he was also honored with an Academy Award for Scientific and Technical Achievement by the Academy of Motion Picture Arts and Sciences.

Readers may contact Richard Shoup at Boundary Institute, P.O. Box 3580, Los Altos, CA 94024 or rgshoup@rgshoup.com. See also http://www.rgshoup.com/prof.