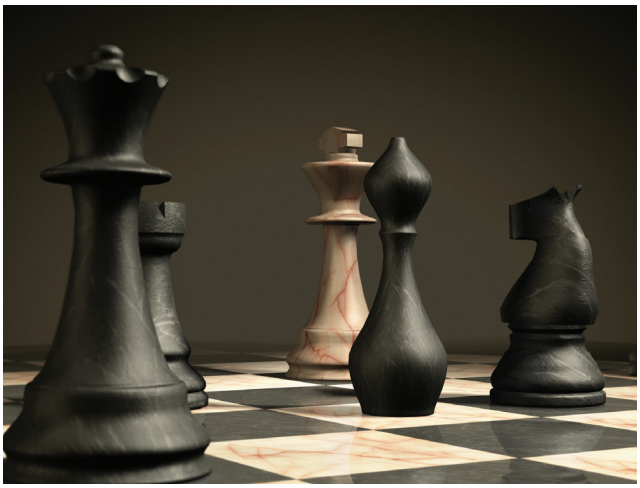

Judson Rosebush on Computer Graphics and Animation

Hal Berghel, *University of Nevada, Las Vegas*

Once animation went digital, we (almost) never looked back. Here I speak with one of the pioneers of animation and multimedia, Judson Rosebush.

Judson Rosebush is a complete technologist. Firmly grounded in both the arts and computer science, he has been at the forefront of digital animation and multimedia since its inception. He was the American editor of *Pixel Vision* magazine, and founded Digital Effects and the Judson Rosebush Company, both of which contributed a continuous stream of innovation to our digital frontiers. He has also been one of the most articulate media theorists of the past several decades. It's in this last capacity that I benefitted most from our association.

I've had the great pleasure of working with Judson Rosebush many times over the past 25 years, and have come to consider him a technology visionary. If you ever get a chance to hear one of his lectures, take advantage of the opportunity. The following should be of interest to technologists of various stripes, software developers, artists, and media critics.

Hal Berghel: How did you get involved in computer graphics and animation?

Judson Rosebush: I was a graduate student at Syracuse University from 1969 to 1970, when Donald Weiner, professor in the Electrical Engineering Department introduced me to the CALD/CAMP [Computer Animation Line Drawings/Computer Animation Motion Picture] programs, written by Sherwood Anderson, who had been a graduate student working with Weiner at Syracuse. These programs allowed you to make 2D or 3D perspective vector drawings on a pen plotter or film recorder.

I made my first plots, and Sherrill F. Martin, a filmmaker in the Boston area, filmed my first vector animations using the SD4020 at MIT's Lincoln Labs in in 1970. In 1972, I produced computer-animated scenes showing arrhythmia monitoring, by using CALD to create film-separation negatives, and then adding color optically. The result was a green grid with the moving EKG in yellow, with blue arrowheads and highlights, one of the earlier computer animations made with color.

In 1974 I made an animated film called the *Polytempi Computer Ballet*, and then the three-minute art film *Space≠*. For the latter, I laid out the film using multiple black-and-white negatives, juxtaposing differing colors using multiple exposures. I used a computer-controlled optical bench at EFX Unlimited in New York for the optical photography, which foreshadowed the end of traditional animation and the optical photograph. *Space≠* was silent, and major fragments of the film are incorporated in some of [pioneering video artist] Nam June Paik's work.

0018-9162/14/$31.00 © 2014 IEEE Published by the IEEE Computer Society

I spent the next years dismantling Anderson's Fortran source for CALD. His code was masterful, his mathematics spot-on. I decided not to build a user interface/programming language and stuck with Fortran subroutines as my primary vehicle. APL was popular at Syracuse, and I also built graphics routines with APL. In the end, after at least three rewrites and with the help of other Syracuse alumni and fellow Digital Effects founders David Cox, Don Leich, Jan Prins, and Bob Hoffman, we developed a Fortran back end and an interactive APL interface that became the production system Visions.

HARDWARE'S EVOLUTION

Berghel: As one of the pioneers in computer animation in a commercial market, you have a unique historical perspective. Please share some of your observations on the evolution of hardware and software.

Rosebush: I grew up with paper tape [and] used punch cards and batch processing on an IBM 370 to make my first movies, testing frames on a pen plotter first. Things had to be correct before filming. Writing single frames to videotape didn't become practical until the 1980s, although we did do logo writing onto an Ampex ESS disk at that time.

APL running on a Tektronics 4013 terminal displayed a single vector graphic frame immediately. This was a big step forward and would be our mainstay of production throughout the early 1980s. After the objects and action got approved, we would run batch production with Fortran.

Seeing representations of color frames was always a challenge. Another Digital Effects founder, Vance Loen, constructed a video digitizer from plans by video artist Bill Etra.

Joe Scala at Syracuse worked with Child Computer to build a true-color, pixel-based system that could image process and run the EXPLOR program, created by Ken Knowlton at Bell Labs.

Digital Effects built its first paint system for [mega marketing firm] J. Walter Thompson in 1979; it utilized a Tektronics tablet and pen along with a color terminal and included code we wrote to draw and paint, albeit crudely. We next built, with the help of Doug Fenster at Syracuse, an 8-bit true-color system around a DEC PDP 11-03 computer that output a legal NTSC [National TV Standards Committee] RGB video signal and was a breakthrough device. I recall that 256 kilobytes of memory cost [US]$25,000. Our second-generation system combined a commercial Lexidata display, a PPD 11-34, and software written in C by Digital Effects' Gene Miller, called the Video Palette II. This included on-screen menus and pen or mouse control, and was used by many early [digital] artists, including Darcy Gerbarg and Laurence Gartel.

In 1981 we acquired a used Dicomed film recorder, which let us write out a full-color frame with a full gradient of shades at a 4K-pixel spatial resolution. It was sharp, had good optics, and made beautiful pictures. Until this time we computed high-contrast separation negatives, optically coloring and texturing them to make all of our pictures. Now we could output exactly what we wanted. We could also preview the full-color images on our Video Palette II or take the pixel images over into the 3D production system.

Our initial work was done using time-sharing and batching on IBM 370 and later on Amdahl 470 mainframes. At Digital Effects, Robert Hoffman and Christine Shostack built an APL-based graphics system, Visions Business, and the company bartered with Scientific Time Sharing Corporation in Bethesda for computer time.

Later, we time-shared on IBM 4331 computers owned by Rapid American, and at some point circa 1982 we bought a Harris 800 mini-mainframe with a hard drive and tape unit, and ran the CPU operation internally. The Harris had both Fortran and an APL, 3 Mbytes of memory, a 300-Mbyte hard drive, and a 1,600-BPI [bits per inch] tape drive. This configuration, with the Harris computing frames and the Dicomed shooting, ran 24/7 for years. One unsung component was a system called Runsheet, written by Hoffman, which managed a shot from the time it was submitted to production until it came out as a piece of film.

During the early 1980s, we purchased one of the very first Silicon Graphics IRIS workstations, as well as an IBM PC. Hoffman and Shostack ported the Visions Business Systems product to the PC, and Hoffman ported Visions to the IRIS. Paul Yurt joined the company at the time as chief engineer.

TRON

Berghel: Your company, Digital Effects, was one of the four studios that contributed to the classic [1982] animated movie, *Tron*. How did you get involved in that production and what were the greatest technological challenges to making that film? Place this effort in the context of the available technology of the time.

Rosebush: I heard about *Tron* at SIGGRAPH that year and got an appointment with art director Richard Taylor at Walt Disney Studios. I returned to New York with the Bit scenes and the formation of the electronic warrior scene that starts the film. Robert Abel and Associates [Abel], Information International, Inc. [III], and Mathematical Applications Group, Inc. [MAGI] were already on the project; I guess Digital Effects had enough credibility, and there was enough work, that we fit in.

The Bit character is a vibrating stellated icosahedron inside a

dodecahedron—the two shapes are geometric duals—and Disney gave us rotoscope positions of where Bit should be, its size, and its state. Deliveries consisted of original camera negatives and a matching high-contrast matte [print].

The formation of the electronic warrior consisted of creating a coplanar database of the figure such that each polygon could separately animate inward, with animated light rays pouring around until the until the figure formed. Taylor requested that we test putting diffusion filters in the optical pathway of the Dicomed. The final shot is an in-camera double or triple exposure that photographs multiple elements onto a single VistaVision negative.

The real technical challenges on *Tron* weren't done by us, Digital Effects (or perhaps even by Abel, who did have a transparent vector capacity), but by MAGI and III. MAGI had to increase their object capability as well as the sophistication of their lighting model. I think in many ways they also defined the look of the product with their starker, very graphic spaces and action—such as the light racers—which defy physics.

A lot of *Tron* was made by creating high-contrast, black-and-white, physical transparencies of the geometric sets, and photographing them with colored gels on an animation stand. In fact, most of the film isn't computer animation at all. It's a tribute to Taylor that all the elements work together.

VIRTUAL REALITY

Berghel: I recall hearing a few of your talks on virtual reality [VR] 20 to 25 years ago where you predicted that once sufficiently advanced haptic systems were developed, virtual reality would become integrated into daily life as next-generation multimedia. All that I see now is computationally intensive, though content-uninspired, video games and movies, simulation interfaces for training, 1970s-style head-mounted displays, and hand-gesture interfaces. Is VR stuck in time?

Rosebush: Technology doesn't always go the direction we think it will. But inventory the capabilities of today's smartphone: text in and out, sound in and out, still pictures in and out, moving pictures in and out, a touch-sensitive screen for 2D analysis, measurement of acceleration for 3D analysis, knowledge of [your] surface position on the planet, and a wireless connection to other smartphones as well as vast back-end computing power.

Two examples. An app that lets you hold and look into the screen and see what part of the real star and planet field lies behind it. Move the device and the viewing field moves. Another app lets a user hold the phone up to a sound source, and the system returns the name of the song and the artist. This demands considerable back-end processing. I guess neither of these applications are, strictly speaking, VR but both are certainly "virtual" in some sense … though not anticipated by predictions made 25 years ago.

I think that you're correct that haptic interfaces may not have taken the forms we thought they would, but when you shake an iPhone to clear a picture off, that is a haptic process. Products like Google Glass suggest we're still growing technology here.

EVOLUTION OF COMPUTER GRAPHICS HARDWARE

Berghel: As you know, I heaped considerable praise on several of your CD-ROM titles in my reviews, especially *Isaac Asimov's The Ultimate Robot*, *Gahan Wilson's The Ultimate Haunted House*, and *Ocean Voyager*. What impressed me most was that your productions were consistently cerebral. How did you get into the business of entertainment CD-ROM titles, and what were the business and technical challenges you faced?

Rosebush: I am particularly drawn to taxonomies; my thesis was on the taxonomy of computer graphics, and much of my contribution at Digital Effects was in defining the variables of an object or scene. Some can be approximated very simply, like color. Other variables evolve; at first glance transparency is a scalar, but the moment you turn it into a triplet you acquire colored filters.

Between the first computer graphics of the mid-1960s and the mid-1980s, most of what was required to specify a realistic, 3D, synthetic world had been worked out.

Between the first computer graphics of the mid-1960s and the mid-1980s, most of what was required to specify a realistic, 3D, synthetic world had been worked out. To whatever extent music notation gave us a way to replicate [certain] sounds, we now had a notational system to represent the 3D visual world and movement within it. After the other computer animation pioneers and I got overrun by either desktop workstations and PCs or supercomputers and server farms, I did a series of documentaries with Laurin Herr's Pacific Interface, Inc., in the late 1990s that explored multimedia, VR, HDTV, digital cinema, gaming, and other emerging technologies.

The direction I chose to pursue was interactive multimedia systems. It was both affordable and

interesting. Herr encouraged me to build with HyperCard, and the hypertext language provided powerful tools for organizing multivariate (who, what, when, were, why) as well as multimedia data (text, sound, pictures, video, process). Unlike a linear book or film or TV [show], an individual HyperCard allows a nuclear datum to represent a specific "who-what-when-where-why" multiaxis model, and then access it in the context of a personality, a place, a chronology, and so on.

I wrote and directed *Isaac Asimov's the Ultimate Robot* CD-ROM for Byron Preiss, and it was published by Microsoft. The project began with a series of Asimov's short stories Preiss had licensed. Around this we gathered a library of book covers and wrote a section on elementary robotics, incorporating animations we built for the project, video clips about robots and robotics, a robot timeline, and an interactive robot toolkit that programmer and graphic artist Matt Schlanger built in Adobe Director.

The technical challenges included how to fit everything onto a 650-Mbyte CD-ROM.

Haunted House was my second project with Preiss. Gahan Wilson developed the houseful of weird characters. Game designers Walter Freitag and Barbara Lantz structured the underlying game, Schlanger took over programming, and illustrator Kathy Konkle animated the characters. We built this property in Director. People think of Director as a language to move things on a screen, which it is, but it also provided us with a language equivalent to HyperText, so that we were able to develop a lot of state management code to keep track of the game inventory and status. The game incorporated both the solving of a lot of traditional puzzles transmogrified into computer games (for example, hangman or sliding puzzle) along with software that tracks [a player's] movements and behaviors. For example, *House* provides an electric prod to the [player], and how the [player] chooses to use it does matter.

Ocean Voyager … featured a cast of four animated characters set in an underwater submarine, which a player may drive anywhere there is water on the planet. All the water is depth-mapped, the sub has a periscope, and a multitude of information resources (microscope slides, books, video cassettes, whale tail profiles) assists the [player] in rescuing a seal. The game challenges and hints to the player. Because there could be an infinite number of pathways to the solution, we developed a goal-directed game engine with a master clock that is able to trigger events as well as [issue a] warning, [such as when a player is] low on air.

Before 2000 [we] successfully migrated these software capabilities to the Internet, and although at first we didn't have the bandwidth of CDs, the Web has provided an amazing vehicle to build multimedia applications.

GETTING 3D PERSPECTIVE

Berghel: What were some of the most notable technological breakthroughs in computer graphics and computer animation?

Rosebush: I have already mentioned that prior to CG [computer graphics] we really didn't have a formulistic way to describe a 3D world, and that all got worked out from the mid-1960s through the 1980s.

Much of the first computer animation we did responded to clients who sought cost-effective ways to replicate existing animation technologies, especially the 2D layered cel system and the streak and strobe photography of animation stand-motion graphics. That didn't last long. Once the art directors discovered they could tell stories in 3D, they did.

Well before 1970, all the big ideas were understood. These included the fact that you could animate 3D-perspective objects, that objects could deform, and that kinematic as well as dynamic methods could be employed in visualization. You could write formulas about how water could exit a pipe, apply the formulas to your digital water, and, using the formulas, calculate and display the results in a time series fashion [animation]. In the 1990s a term for this was invented: scientific visualization.

In terms of creating art, I believe that the ability of an artist to define a set of rules and then let the computer calculate and edit a result, or what I have called *proceduralism*, is one of the most important ideas in art during the late 20th century. This abstracts the creative process because instead of working directly on the canvas, the artist works on a procedure that creates the canvas.

Initially, most computer animation was kinematically driven using transformation matrices and matrix multiplication. Speaking for myself, it wasn't until I wrote *The Computer Animator's Technical Handbook* with Lynn Pocock in 2002 that I understood that kinematics had a rich background and a deep relationship with robotics. I think one of the big breakthroughs was to incorporate dynamic and goal-directed systems into animation, including constraints and inverse kinematics. Suddenly there was another way to animate. Goal-directed behaviors are also important. …[In] our game engines, we imbued characters with properties like hunger, need for air, aggression, and love, so that as the game clock ticked, these various goals could compete with one another to drive character behavior.

Every now and then something does come along [that] shows the full capacity of the medium…. Techniques like the moving camera freezes of *The Matrix* or the shifting planes of *Inception* suggest to me

that ideas remain to be discovered and exploited. And *Pi* suggests that the boundary between the synthetic and the real has dissolved.

Berghel: In the remaining space would you care to predict where computing will be in 25 years. (Understand that there will be a follow-up in this column in 2040!)

Rosebush: The developed world has gone digital: all commerce, all media, and, increasingly, all government. Money, banking, and stocks are just digits; products and inventory are controlled with bar codes; cell phones require GPS. The Internet and smartphone are transformative innovations; however, their future development might be incremental (data rates) rather than innovative.

Computing is cheap and ubiquitous, and its effects will be felt in sociological terms. It's now in the hands of individuals as well as big business, big police, and big government. Unlike statistics, which generalize from the sample to the whole, big data seeks out the individual. Every financial transaction, piece of mail, website visited, and phone call is now tracked. Image recognition, transactional analysis, and automated debiting define the new world. The mechanics of this are complete; developments during the next 25 years will be analytical and directed toward controlling individual access to information, monitoring behavior, and automating income generation for the state and organizations with maximum computing resources. See George Orwell's *1984*.

At the same time, we can expect to see sophisticated data analysis to better understand DNA and disease, increase energy efficiency, grow crops with less water, and packet switch everything from bits to railroad cars.

During this time the developing and modern worlds will struggle to equalize, and citizens will struggle to counterbalance the police states.

Hal Berghel *is an ACM and IEEE Fellow and a professor of computer science at the University of Nevada, Las Vegas. Contact him at hlb@computer.org.*