Department: Visual Computing: Origins
Editors: David Kasik, dave.kasik@gmail.com
Chris Johnson, crj@sci.utah.edu
Mary Whitton, mcwhitton@gmail.com

The Graphics System for the 80's

Nick England

***Abstract*—Forty years ago, the first General Purpose Raster Graphics Processor made the transition from research project to commercial product. This is the story of the creation of a new graphics system and the startup company that produced it in the early days of raster computer graphics.**

INTRODUCTION

Forty years ago, you could buy a Programmable Graphics Processor from Ikonas Graphics that could produce ray-traced, antialiased images of bicubic B-spline surface objects, volumetric rendering of 3-D data sets, and real-time display of constructive solid geometry operations (see Figure 1).

The processor could also perform fast image processing, real-time 3-D line and polygon drawing, and almost anything else that you could program. This recollection is about how this design and the company that made this product came to be.

Note: I have probably mischaracterized, omitted, embellished, miscredited, and otherwise made mistakes in telling this story from dim memories. For convenience, please insert an imaginary IIRC before each sentence.

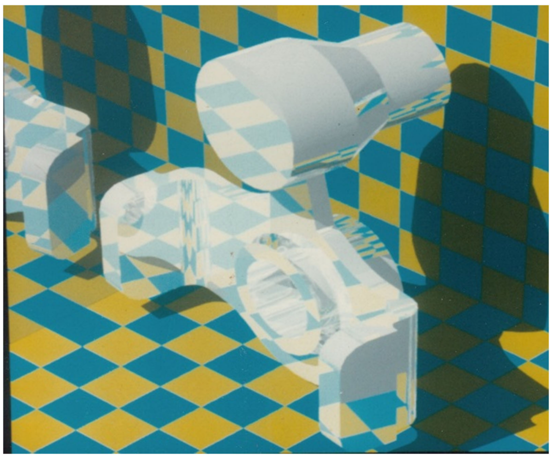

Figure 1. Ray-traced image computed entirely on an Ikonas graphics processor designed in 1979.

Digital Object Identifier 10.1109/MCG.2020.2983816
Date of current version 28 April 2020.

I will try to convey the background of my idea for designing such a system, the environment in which the prototype took shape, the genesis of a company that turned a lab prototype into a product, some of the technical and business lessons learned along the way, the pleasure of seeing smart people use a tool that I helped create, and what legacy this effort might have left.

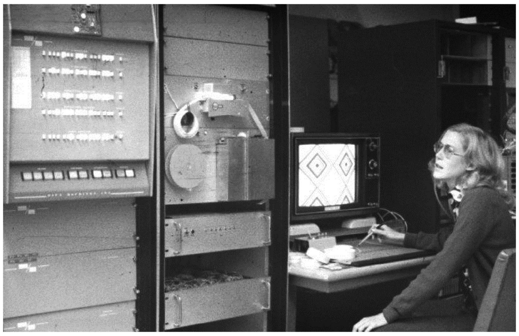

Figure 2. Debbie Ogden's textile design system using Rand tablet, Varian 620, and color display.

GRAPHICS EDUCATION

After graduating in Electrical Engineering and working for a few years, I returned to North Carolina State University (NCSU) in 1972 to learn something about computers and digital hardware. I had done some work as an undergraduate for Professor John Staudhammer, so I contacted him hoping to get a part-time job. I walked into his lab and experienced my own Road to Damascus moment—for there I saw a computer screen displaying a moving 3-D object. I was immediately (and still am) hooked.

John had created a computer graphics research group as part of the department's new Signal Processing Lab—back in the days when university fiat required all computing resources to be centralized behind glass walls and fed with punched cards, the Lab's computer had to be called a Signal Processor. It was indeed a very fine Signal Processor, but the Adage AGT-30 was also a great interactive 3-D graphics display embedded within a highly capable 30-bit computer. The AGT-30 had a hybrid digital–analog section for providing 3-D transformations of vector lists on the way from core memory to a calligraphic display. The interactive software tools were amazing in an era of punch cards and dial-up teletype. There was an on-screen editor that even let you use button switches for search/replace/insert, etc. Only a person who has edited code using a Model 33 Teletype can appreciate how wonderful this was. You could insert assembly code into a FORTRAN program to handle real time interrupts and the like, and the instruction set included indirect addressing so making linked lists for creating/editing 3-D object representations was a natural. It was a great machine on which to learn 3-D graphics.

John had brought in a smart gang of grad students and undergraduates to form an amazingly creative research group where hardware and software development were seamless partners. If you needed hardware, you designed and built it; if you needed software, you wrote it—it really was all the same process. Ed Tripp and Dave Wooten designed and built an interface from some additional generic core memory to the AGT-30. That let Jeff Eastman create an interactive 3-D modeling program on the Adage. John got a donation of the bits and pieces of an early educational computer system that included a slo-mo magnetic video disk, a Varian 620 minicomputer, and a Rand tablet.

Glen Williamson redesigned the disk to act as a 640×480 full color video refresh buffer (one track each for R, G, and B) and Jeff built a FIFO interface from the Varian to write scan lines to the disk. Meanwhile Ed and Dave built a high-speed DMA link between the Adage and Varian. There it was: one of the earliest full color raster display systems tightly coupled to an interactive 3-D computer. As the next step, Jeff wrote a scan-line hidden surface display algorithm on the Varian (in assembly language) which received data from the 3-D modeling program running on the Adage and created a color shaded surface display via the video disk.[1]

This hardware/software system started running not long after I showed up in the lab. My first task was designing and building an interface for the Rand tablet and writing a simple sketch program. Later, Debbie Ogden used these tools to help create a paint program for textile design (see Figure 2).

Hardware, software, whatever was needed, we built it. I think that our hardware design orientation (modular functions with well-defined interfaces) showed up in our software as well—I remember that it seemed quite easy to borrow code from others in the lab.

This fertile (and fun!) mixture of hardware and software innovation was where I learned graphics, hardware engineering, and software development. That set the stage for what was to come next in this tale.

MOTIVATION

Around 1977, Professor Ray Stroh got a small research grant from NASA Langley Research Center to investigate new ways of generating cockpit displays. NASA wanted to experiment with raster displays for the "glass cockpit" of the future—up until then everything had been calligraphic display based. I latched onto Ray's project (I honestly do not know what would have happened if he had tried to say no) and started thinking about how to create real-time raster display of current cockpit info plus support NASA's future display concepts such as "highway in the sky."

By this time, I was experienced and proficient at designing/building pieces of hardware and writing graphics and image processing software. The conventional microprocessors of the time were way too slow to handle this task. At first I thought about dedicated hardware vector and character generators. I built a prototype hardware vector generator that ran about 100 ns/pixel, but it was not clear to me how I could turn that into a complete display generator. I had also designed (but not yet built) a fast processor to drive a calligraphic display—something like an updated AGT-30—but that design was not suitable for driving a raster display either. And then a "perfect storm" of new technology appeared.

BUILDING A MACHINE

Remember that this all happened pre-Internet. The way you found out about something was through magazines and conferences. And it was before gate arrays and ASICS. The only people who could make chips were really big companies like AMD, Texas Instruments, Signetics, etc. A series of articles appeared in one of the trade rags about AMD's new 2901 bit slice family – 4 bits of arithmetic unit, register files, etc., per chip. Put together four of these chips and you had the guts of a very fast 16-bit computer. Other articles appeared about TRW's new 16 × 16 multiplier and multiplier-accumulator chips (you would not believe what a huge pain it was to build a multiplier out of off-the-shelf chips). And another article appeared about Mostek's new 16K × 1 DRAM (wow – 512 × 512 × 1 display in only 16 chips!).

Well, you can guess the result—I realized I could now build a programmable graphics processor plus frame buffer that was fast enough to create the kind of 3-D displays NASA wanted—and give me a wonderful new toy with which to play. I started drafting a design on the backs of sheets of fanfold printout (no PC-based CAD back then) and ordering parts—only $60 each for those new 16K × 1 DRAMs.

There weren't any PC-based printed circuit layout systems back then either, so I built the prototype the same way we had built other projects in the lab. I used wire-wrap IC sockets on 8" × 10" printed circuit boards laid out using the standard technique of the time—black tape on acetate sheets plus Bishop Graphics stick-ons for DIP sockets and edge fingers. I was not absolutely certain of my design (no hardware modeling software then either), so I just laid out power and address lines in the memory array and left all the control logic to be wire-wrapped. I also laid out a board to take the 40 pin DIP packages for the 2901 (eight of them made up my 32-bit processor) and more edge connectors for ribbon cable to connect to the 64-bit wide instruction RAM. This machine had a separate instruction memory (composed of multiple fields to control multiple simultaneous functions) that ran in parallel with the data memory. The processor accessed a 512 × 512 × 2 frame buffer over a 32-bit bus that also had an interface from the host Varian 620 minicomputer. The frame buffer fed a video digital-to-analog converter so I could also run the system as a 256 × 256 × 8 frame buffer (see Figure 3).

Once I had established the processor's instruction set, Ray Stroh wrote a cross-assembler that ran on the Adage. I built and debugged the system and wrote a Bresenham algorithm vector drawing routine for the 32-bit processor.

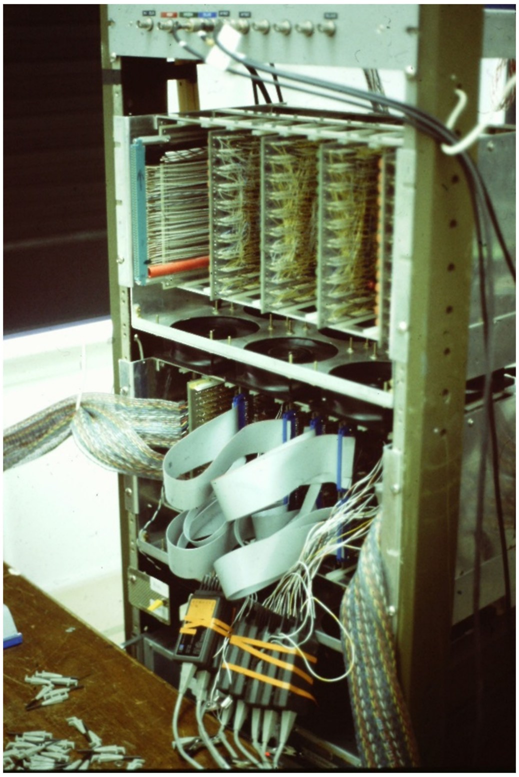

Figure 3. Prototype graphics system at NCSU with logic analyzer probes attached.

Ray's program generated a display list on the Adage (where it was also displayed on the calligraphic display), passed the display list to the Varian which then sent it along to some static RAM on the display processor's data bus. And there we had it—a real-time raster display system that pleased NASA and looked like it would be a lot of fun to play with.

Naturally, I wanted to do more with it. I got inspired by attending SIGGRAPH in 1977 and started building an augmented-reality system for interactively modeling real-world objects. At this time, fellow grad student Turner Whitted was rendering B-spline surfaces, and I wanted to create models for his display program. Glen Williamson and I built a stereo display from two TV cameras pointed at an object and presented a combined image to the user via a polarization stereoscope (two monitors combined with cross-polarized images). It displayed B-spline patch meshes and cursor objects using my new graphics system combined with the TV camera images.

The mesh computation for my SIGGRAPH '78 paper presentation[2] actually happened on the Adage and was then displayed via my graphics processor. My plan was to use the TRW multipliers for 3-D transformation as well as for patch evaluation and subdivision. Mary Whitton was busy wire-wrapping the matrix multiplier boards, but I never got to use them for patch evaluation (or to finish my Ph.D. program).

[Aside: Mary and I had wed in 1974. After seeing how much fun we were having in the graphics lab, she switched from teaching junior high school math to being an enthusiastic electrical engineering student.]

STARTUP

Then, other graphics researchers saw what I had built in the lab and wanted a toy/tool like that for themselves. John Staudhammer and some of his other students had built a couple of hardware systems in his basement, so Mary and I figured we would follow his model. John was smart enough to get out of that business just as we decided to give it a try—our sophisticated business plan was to build four or five systems, take whatever money we made, and go on a vacation to Europe. When the money ran out, we would come home and get real jobs.

If that sounds like we didn't know what we were doing, it's because we didn't. Neither of us had any business experience. We probably should have named the company Clueless, Inc., but after toying around with the usual Tekagraphitronix names we settled on Ikonas Graphics Systems (fellow grad student Stavros Boinodiris contributed the name). Corporate headquarters was the back room of our rental house.

Our first corporate purchase was a power screwdriver to disassemble my model railroad in that back room. Our next purchase was a computer system. Mary assembled a Heathkit H-11 (DEC LSI-11) while I built an ADM-3K terminal kit (kits were cheaper). We also started designing/building a 512 × 512 × 8 demo graphics system. Neither one of us had any electronics manufacturing experience so we went with what we knew—the same wire-wrap board construction I had used at the university (see Figure 4).

At SIGGRAPH '78, we distributed a one-page flyer promising to build most anything a customer wanted and picked up our first purchase

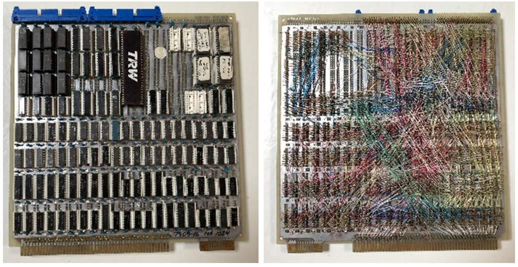

Figure 4. Wire-wrapped Ikonas programmable matrix multiplier.

order—a batch of frame buffer boards for Chuck Csuri's graphics lab at Ohio State. It was not a complete system because Marc Howard (NCSU lab alum) wanted to build his own system there.

The first order for a complete system, including programmable graphics processor, came from Don Woodward at the University of Texas Health Science Center, Dallas, TX, USA. That first order really defined the system we'd produce (eventually making about 400 copies). As Don and I talked on the phone he'd ask questions like "Will it do 1024 × 1024 as well as 512 × 512?", "Does it have video input?", etc., and I'd reply "Sure!" if I had even a vague idea of how I would design such a feature. I certainly had no shortage of naïve optimism and self-confidence.

PRODUCT INTRODUCTION

I designed the system around a synchronous multimaster bus with 32 bits of data plus 24 bits of address. Via this bus, both the host computer and the embedded graphics processor had the same access to memory, control registers, etc. (see Figure 5). Some customers bought display systems with no processor, while a few others added their own custom processors. I added a few more functional units to the Ikonas version of my original processor. The result was a machine that was general purpose, yet optimized for graphics and imaging functions. In a single instruction you could add two numbers, multiply two numbers, increment a loop counter, write a pixel, and branch to a subroutine based on a condition code—all simultaneously in a single clock cycle.

The frame buffer memory could be accessed in multiple ways—one pixel at a time, multiple pixels at once, or as 32 bit general purpose RAM. The frame buffer memory could be used to store a Z buffer as well as multiple frames of color info. The video display stream carried 32 bits per pixel as RGBA or, via a bit-level crossbar switch, in most any other format. The bus and processor clock was independent from the video pixel and frame rates which were completely programmable.

Actually almost everything was programmable, including the display resolution—the downside being that when you turned the system on, nothing worked until you programmed it.[3] The upside of display programmability was that the system became a favorite of visual perception researchers and of graphics terminal developers like DEC, Tektronix, and HP

While I defined the system architecture and designed many of the plug-in boards, development now began including many additional people. We kept on with the "that's how we did it in the lab" philosophy, recruiting people we had known at the NCSU lab: Steve Holzworth to write software and Allan Sadowski part time to help build boards. Mary's brother Robert took on the financial management (not many bookkeepers

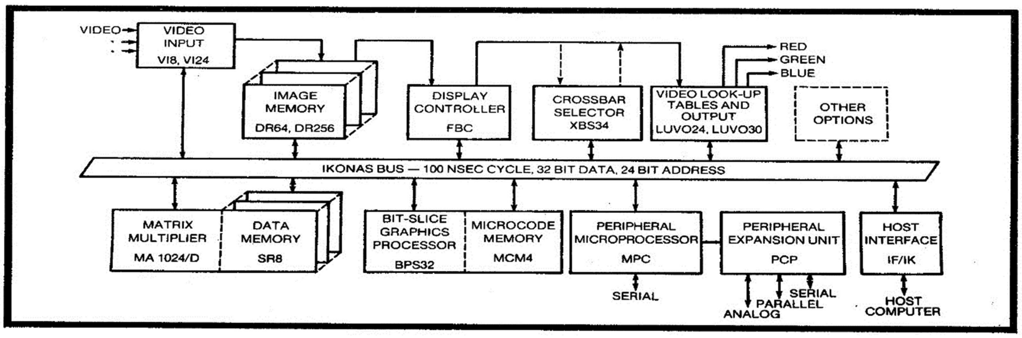

Figure 5. Ikonas Graphics system block diagram.

Figure 6. Students building early Ikonas wire wrap boards.

had a Ph.D. in math). In high hopes, we moved the company out of our house to an office a few miles away. As additional orders started showing up, Allan roped in more NCSU students who turned up to solder and wire-wrap in the afternoons after class (see Figure 6).

Turner Whitted and John Jarvis at Bell Labs got system #2, and Henry Fuchs and Fred Brooks at UNC-CH got #3. Things mushroomed from there and soon we needed some full-time employees. One of Allan's classmates was Xuan Le who volunteered "I have a cousin." Actually, Xuan had several cousins, and eventually we ended up with an assembly area full of Vietnamese refugees. Pot luck lunches were great!

For SIGGRAPH '79, we had finished the new backplane and chassis, built a 512 × 512 × 24 display (a full-color frame buffer was a pretty big deal back then), interfaced it to our LSI-11's Q-Bus, and had some demo programs and images to display. I had not quite finished the new processor design, so I packed a suitcase full of chips and tools. My plan (honest) was to finish the design the first night, wire the boards the next day, and write running demo code before the exhibits closed. This was something far beyond naïve optimism and that plan did not survive past the first night.

But, boy, did we have images to display—Turner Whitted presented a paper with his first famous ray tracing images. So we loaded Turner's images from our 8" floppy disks (one disk each for red, green, and blue) and attracted an amazing amount of interest—we had the hit hardware of the trade show displaying the hit imagery of the technical conference. For SIGGRAPH '80, Turner

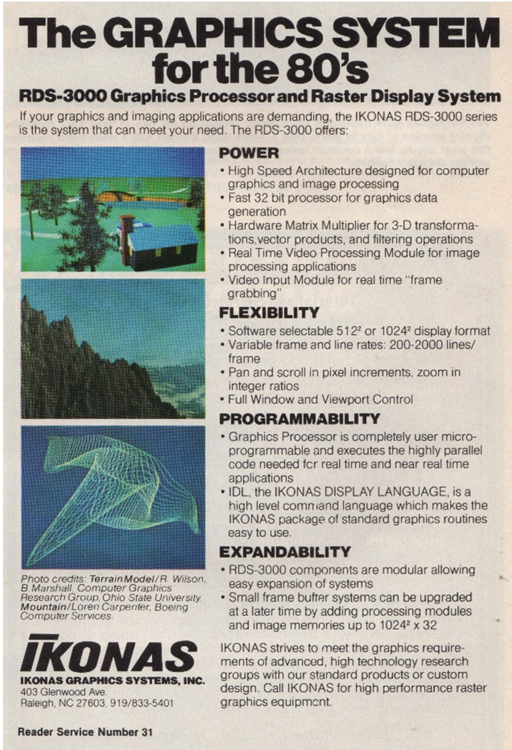

Figure 7. Advertisement in the first issue of IEEE Computer Graphics and Applications magazine.

said "You ought to contact Loren Carpenter at Boeing." Loren's fractal mountain images drew crowds to our 1980 booth. . .and we also could demonstrate the graphics processor this time. Over the years we continued to showcase customer images along with our own creations—SIGGRAPH and NCGA shows were always hectic, fun, inspiring, and rewarding.

GROWING THE COMPANY

Our system really did become a standard for graphics research labs and it was wonderful to see that our tools were being used in industry and academia (see Figure 7). We provided information on interfacing to the internal bus and several customers built hardware as well as software. For a couple of years over half of the SIGGRAPH technical papers were from researchers using our equipment.

Gary Bishop at UNC-CH and Preston Gurd at Waterloo both created C compilers for our graphics processor. Those compilers let our users be far more productive and creative than

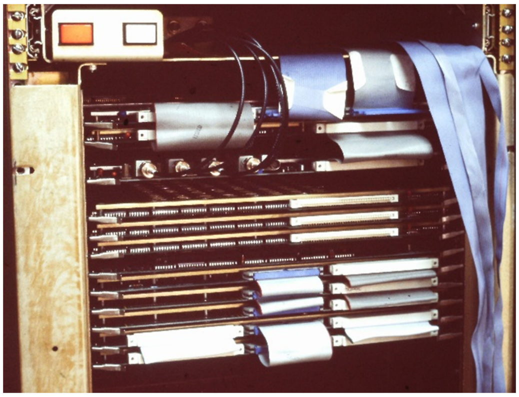

Figure 8. Final version—Ikonas/Adage RDS-3000 with 15” × 15” printed circuit boards.

having to write 64-bit assembly code that looked like the following:

```
SOLNEG: B6 RA11 CARH0 RPS ALUMAR BD
    NCCMEMAC JMPRDF SOLWT1
SOLWT1: RA6 PR ALUMAR CCMEMAC JMPDF.
SOLWR1: RA2 B6 REOS LRESWR MASHIKA
```

Along with the manufacturing expansion, we had recruited NCSU grads Henry Rich and Pete Evans to work on additional hardware design. Mary designed a programmable matrix multiplier and demonstrated B-spline patch evaluation with it to finish her MSEE degree.

We also added several more software developers to create tools and a graphics language (IDL) for real-time display applications. NASA, Lockheed, Boeing, and many other aerospace groups used IDL for cockpit display prototyping; the last Ikonas systems in daily use were still generating displays in a cockpit training simulator over 25 years later. NASA definitely got their money’s worth out of that initial university research grant.[4]

We had a wonderful team of people and it was a fun and exciting time to be on the cutting edge of graphics development. Besides working with academic and industrial research labs, we delivered systems to a number of firms in the entertainment industry—Marks & Marks, Robert Abel, Lucasfilm, Atari, and others. It was great fun to be able to cheer for “our pixels” on TV or at a movie—yes, of course it was all produced with their software, but we still took pride in supplying useful tools to creative people.

As the business expanded, we naturally ran into financing issues but we still didn’t know all that much about running a business. We decided to merge into a local software company in a deal put together by a venture capitalist to raise money through a public offering. We realized that may not have been our smartest move when the SEC padlocked the doors of the investment bankers that were going to handle this public offering. Eventually, in 1982 Boston-based Adage, Inc. bought the entire mess in order to acquire Ikonas. This was the same Adage that had made the AGT-30 on which I learned graphics. Adage moved manufacturing to Massachusetts and we no longer had Vietnamese spring rolls at our pot luck lunches. Sigh.

Things definitely changed as part of a larger company (see Figure 8), and one particular benefit was the ability to hire a few new engineers. Tim Van Hook came from Ohio State where he had developed animation in Chuck Csuri’s lab and built an extremely fast polygon-drawing hardware system in a startup that had folded. Tim was amazingly creative and productive. His academic training was in Fine Arts, but he taught himself software and hardware development. In 1985–1986, Tim developed the dexel buffer (linked-lists of front and back surfaces at each pixel) for his ray-tracing and real-time CSG code mentioned earlier.[5]

The hardware was the same that I had designed in 1979–1980, but it took time for software development that would exercise it to its full potential.

LESSONS AND LEGACY

We learned a few lessons along the way.

- Building tools for smart people is fun—understanding your customers and their needs are key.
- Manufacturing products is tougher and far more expensive than making lab prototypes.
- Creating a viable business without any experience can be stressful (understatement!).
- It takes innovative software to show the benefits of flexible hardware.
- Smart, honest partners are the best investment you can make.
- When going through airport security, do not refer to your wire wrap tool as a wire wrap gun.

Tim was the major thread that carried on whatever legacy my early programmable graphics processor has had. Tim, Mary, and I left Adage and started Trancept Systems. There Tim designed a board-level programmable graphics + imaging + computation processor (with multiple processing elements and 200-bit-wide instructions) that plugged into Sun Microsystems workstations.[6] After Sun bought Trancept in 1987, Tim developed volume rendering and other interesting software, added imaging instructions to Sun's SPARC processor, and eventually left for SGI. At SGI, he was architect for the Nintendo 64 graphics processor, then cofounded ArtX where he was architect for the Nintendo GameCube graphics processor.

ArtX was then bought by ATI and I would like to think that Tim's early experiences with the Ikonas system provided some DNA that still shows up in today's GPGPU concepts. Eventually ATI was acquired by AMD (the same company that had provided the chips for my first programmable graphics processor), completing a cosmic computer graphics hardware circle of some sort.

All-in-all it was a Grand Adventure thanks to all the wonderful coworkers and patient customers who made it possible.

REFERENCES

1. N. England, NCSU Graphics Lab. [Online]. Available: http://www.graphics-history.org/ncsu/index.htm
2. N. England, "A system for interactive modeling of physical curved surface objects," *ACM SIGGRAPH Comput. Graph.*, vol. 12, no. 3, pp. 336–340, Aug. 1978.
3. N. England, "A graphics system architecture for interactive application-specific display functions," *IEEE Comput. Graph. Appl.*, vol. 6, no. 1, pp. 60–70, Jan. 1986
4. N. England, Ikonas Graphics Systems, Inc. [Online]. Available: http://www.graphics-history.org/ikonas/index.htm
5. T. Van Hook, "Real-time shaded NC milling display," *ACM SIGGRAPH Comput. Graph.*, vol. 20, no. 4, pp. 15–20, 1986.
6. N. England, Trancept Systems, Inc. [Online]. Available: http://www.graphics-history.org/trancept/index.htm

Nick England is now retired after multiple startup companies and has fun restoring vintage Teletype and U.S. Navy radio equipment (see http://www.navy-radio.com). To contact him with comments, corrections, additional info, etc., you can try decvax!harpo!duke!unc!mcnc!ikonas!nick, or, better yet, nick@graphics-history.org.

Contact department editors David Kasik at dave.kasik@gmail.com, Chris Johnson at crj@sci.utah.edu, and Mary Whitton at mcwhitton@gmail.com.