Art and science connect at SIGGRAPH

Torrey Byles, Staff Writer

More than 27,400 people flocked to SIGGRAPH 85 this July. From the 22nd to the 26th the Moscone Center and a number of San Francisco hotels vibrated with the excitement. Over 230 exhibitors showed their wares, as technical sessions explored the higher dimensions of computer graphic technologies and techno-artists and system-scientists exchanged ideas on the showroom floor, in the galleries, and at the many receptions and parties.

The conference played host to numerous new product announcements, including a wave of new graphics-adapter hardware for IBM PCs and new networking software for Apollo workstations. Bill Poduska, founder and chief executive officer of Apollo, held that this year and next are likely to see great advances. “Technology advances in step functions—not in a continuous manner,” Poduska said. “We are now in the middle of this step function.” Following are just a few highlights from the many technical sessions, tutorials, and panels at SIGGRAPH 85.

Animation and rendering

Some of the more popular sessions dealt with mathematical and algorithmic foundations of image generation.

Ken Shoemake, formerly with Pacific Data Images and now an independent consultant, explained a recent discovery he made involving the animation of a revolving 3-D solid. Shoemake was dissatisfied with the traditional Euler angle approach to animating rotating objects, which gave the animator little control of the axis. In some cases the object would tumble with no apparent axis of rotation at all.

Shoemake discovered an algebraic system invented in the 19th century but never seriously applied. Called quaternion curves, the system enables the solid modeler to rotate an object with three degrees of freedom around any axis chosen. The user simply identifies the surface the axis is to penetrate, then, using Bezier interpolations, computes the quaternions. The quaternions give a simple map of rotation that is fast, easy, and interactive.

In a course on “Image Rendering Tricks,” many ideas were presented. In a talk titled “How to Scrounge Cycles,” Turner Whitted, of the University of North Carolina, spoke of the machinations computer graphic artists must employ to handle computationally intense work. “The first thing you do with a ray-tracing program is never use the words ray trace in the comments,” advised Whitted. This will keep the DP personnel from knowing the true—computationally immense—burden of your program. Most people who do ray tracing have a frame buffer at home, he continued. A good way to crunch out a job, claims Whitted, is to divvy out frames to your friends and have them crunch those frames on their home computers. Ideally, use your friends’ resources when they’re away on vacation.

Whitted explained a “recursive lighting model” that calculated the diffusion and attenuation of trans-
mitted light. The elaborate shadow model, claimed Whitted, accomplished ray tracing better than a more straightforward ray-tracing model.

To efficiently debug ray-traced scenes, Whitted suggested rerendering from a different angle. The user can even project specific rays within this new perspective and gain a better understanding of the dynamics of the geometries, color, and light-source parameters. This is much less time-consuming than rerendering a straight-on ray trace.

Then Robert Cook of Lucasfilm discussed the advantages of using randomness in the rendering of patches and fractals. Surfaces that reflect different intensities of light, especially metallic surfaces, are difficult to render smoothly without creating a banding effect—an unsmooth transition from one intensity of reflected light to another.

To determine the light intensity and color values for each pixel, traditional ray-tracing methods sent rays through pixels at regular intervals—through the centers of the upper right-hand corners of the pixels. Cook found that stochastically determining point locations within each pixel through which to send the ray eliminated the undesirable aliasing effects, or “artifacts.” By picking a random spot within each pixel for the ray to penetrate, “aliasing is replaced by simple noise, which is far less objectionable,” says Cook.

The same use of randomness can be applied to the problem of motion blur. In addition to stochastically sampling for a point location, the algorithm stochastically samples for a moment in time to send the ray. The problems associated with motion blur (jaggies, image crawling, flashing, and popping) can thus be eliminated.

Finally, Richard Chuang of Pacific Data Images (Sunnyvale, California) talked about the intricacies of rendering sequences used for television. He advised doing no more detail work than the medium can support. Television has a limited color range, so an animator is wasting time calling for more colors than are possible. Television is low resolution and interfaces two fields consecutively to produce a single frame. So the animator should always render in the lowest resolution possible, and on a per-field basis, not per-frame basis.

Simulation

The application of graphics to real-time simulation originated with flight simulators, but the scope and complexity of graphic simulation has increased to encompass both more sophisticated visual simulation and the modeling of objects and processes, computer prototyping. Some of these newer applications are scientific analysis, and architectural, mechanical, and genetic engineering.

Robert Langridge, director of the computer-graphics laboratory at the University of California, San Francisco, spoke of the usefulness of simulation in understanding the complex chemical interactions during genetic reproduction. Watson and Crick's physical model of the double-helix DNA molecule—a perceptual aid of revolutionary dimensions—expands scientific understanding even more when it is electronically represented and manipulatable by computer. Computer models are indispensable for studies of such large molecules as DNA and other proteins and interaction of molecules with each other, which are needed in drug design.

Simulation is becoming an effective tool for the design of robots and automated factory cells, according to panel speaker Randy Smith of SRI International. Specifically, robot/workcell simulation helps the robot programmer determine the reachability of objects in the robot's work area, detect collisions between robots and obstacles, and monitor and optimize the cycle time of robot motions. The Advanced Technology Division of SRI, Smith's research group, is developing an off-line programming system for robots called WORKMATE—WORKcell Modeling Analysis Training Emulation.

NASA's Ames Research Center is using simulation graphics to understand the flow of particles over three-dimensional objects, according to Thomas Lasinski. Just as the principal tool for wind-tunnel analysis is photography, computer graphics and simulation are the principal tools for numerical simulation. Using a passel of IRIS workstations, DEC's and CDC minis, and Cray XMP com-
puters, research scientists simulate such phenomena as the reentry into the atmosphere of the Space Shuttle, fluid flow around the rotor blades of a turbine, and other simulations that involve the calculations of many nonlinear (Navier Stokes) fluid dynamics equations.

Peter Doenges of Evans and Sutherland described a complex simulation system E&S helped develop for Daimler Benz of West Germany to test the performance of new automobile prototypes. The system not only generates a 180-degree field of view at 60 Hertz refresh, but controls the subsystems of vehicle motion, instrument display, and operator performance measurement. Other applications for simulator systems, according to Doenges, are helicopters ships, amphibious hovercraft, submersibles, and land, space, and remotely piloted vehicles. Group. “We want to make an integrated environment that lets you know what is happening in the model space, in the data space, and in other subfunctions. The question is, how many things can we manage at once?” Dunn suggested that, ultimately, control should be shared by both human and machine. Working together, the two agents should be considered “partners at the interface,” according to Dunn.

Such ultimate interfaces will rely heavily on windows and concurrent task control. Bill Laaser, formerly with Xerox PARC and now with Metaphor, noted that the ideal of multitasking is changing. Previously, in shared environments, the CPU was the most valuable resource, and multitasking was geared to make the most efficient use of it. With the advent of microcomputers, where each user has a CPU, the user becomes the most valuable resource. Now multitasking needs to be geared to give maximum leverage to the user.

The question of how far we can go with interfaces was a subject for controversy at the session. People generally recognized that too many cues, prompts, and windows can lead to distraction and confusion. Many in the audience voiced the opinion that computers are tools, and users of computers need to acquire a certain level of understanding of the tools before they can use them effectively. Therefore, computer interfaces need not employ elaborate cognitive mechanisms. Still others envisioned computers that would substantially aid the user in executing the task at hand. One member of the audience suggested a self-teaching system where the user learns the principles of the task being performed. CAD workstations would teach engineering principles, and office systems would teach accounting, typing, file management, etc.

Debates over font formats

With major innovations emerging in electronic printing and publishing, it was no wonder the session on font formats was emotion charged and controversial as various vendors and type designers exchanged views on the direction they felt the technology should go.

Font formats are the procedures for characters to draw themselves. The procedure turns the binary representation of an alphanumeric character, along with its parameters for size and font style, into a “bit map.” A bit map is the specification of which pixels (on the CRT) or which dots (on the printed line) should be black to draw the given character.

The panelists agreed that the procedure for transforming a binary character code into a bit map should be executed by the printing device, not the main system CPU. The CPU should simply send the text, typeface name, and font size to the printer, then let the printer figure out the geometry involved.

The primary controversy revolved around the proper way to think about character fonts. Some held they were simply another form of graphics; others felt they were a
special case of graphics. The distinction is important to scaling. Those who believe that fonts are another form of graphics held that changing the point size of a font is simply a matter of multiplying its bit map by some constant coefficient. Those who disagreed pointed out that this is an oversimplification. “Readability is nonlinear,” stated Phillipe Coueignoux of Data Business Vision, one of the first researchers in the area of font mathematics. As a font is made larger or smaller, its proportions change as do the spaces between characters in the text (the “kerning”). “Fonts are not meant to be scaled or rotated” and therefore text is not graphics, Coueignoux said.

Disagreeing with him was John Warnock, founder and president of Adobe. Adobe, with its font-generating software, Postscript, is one of the leading companies ushering in the new “desktop publishing” revolution. Postscript is currently used in Apple, QMS, and Dataproducts printers, as well as in Linotype typesetters.

“Handle text exactly the way you handle graphics,” held Warnock. Because Postscript characters are scaled, the 20 bytes of information provided generate many point sizes and variations (such as boldface, italic, and underline) for a given font set.

Vaughn Pratt, one of the founders of Sun Microsystems and a professor at Stanford, examined the nine basic curve families used to describe font shapes. He recommended a trade-off in process/memory costs. To define character outlines with lower order equations (simple lines), many separate equations are necessary. To define outlines with the higher order equations, fewer equations are required but more coefficients per equation are necessary. The happy medium between the two, according to Pratt, is the conic group of equations with seven degrees of freedom. These reduce storage requirements, can be drawn quickly, form the basis for higher order curves, and are already built into the CGI standard (but are not yet built into CORE, GKS, or PHIGS).

Metafont is another algebraic language, like Postscript, for defining character shapes. Developed by Donald Knuth six years ago, it has been a longstanding language. Metafont uses the higher order parametric cubic equations with 10-11 degrees of freedom. Metafont is public domain software.

Film and Video Show

The SIGGRAPH 85 Film and Video Show, which played two gala evenings, became the main event. The show screened two and a half hours, with many separate entries, of video animation. Entries were divided into five classes: classics, education, art, science, and production demo reels.

Pieces ranged from the sublime to the hilarious, altogether demonstrating the emotive power of computer graphics. The show’s grand finale, Tony de Peltrie, was remarkable because it realistically rendered human facial expressions and fine finger movements at the piano, portraying the life and personality of a man, a bar piano player, who was sadly past his prime (see Figures 1-5). People went away saying, “What terrific ‘acting.’”

The short begins with Tony playing a piano alone on a stage. Nostalgic, he remembers the golden age of his career. “For twenty-seven years, six nights a week . . . ah, it’s tough being an entertainer . . . Seems I’m not in anymore . . . I was the best, the craziest, and now, well, it’s dust to dust, and somebody new comes along.”

Tony de Peltrie was created and produced by an independent group of computer animators: Pierre Lachapelle, Phillipe Bergeron, Pierre Robidoux, and Daniel Langlois. All four are graduates of the University of Montreal. The group used the university’s computer resources to make the film. Notable among these resources was the software environment called DADS. Programmed in Pascal, DADS contains a library of 700 procedures that “allow noncomputer users to create, display, and animate three-dimensional objects instantly on a graphics screen,” according to the creators. Also used to produce the piece were Cyber 835 and 855 computers, a Raster Technologies 1/25S monitor, a Tektronix 4014 terminal, and a Norpak Supervision terminal. The piece lasted eight minutes and contained more than 11,000 frames—each requiring five minutes of computer computation to composite the final version. The film’s creators estimated four man-years went into the creation of the film.

Figure 4. Tony dreams of the faces, music, and times of his past.

Figure 5. Tony leaves this world and all the memories that had come to be his whole world.