And the Oscar

...three computer scientists—Pixar's Loren Carpenter, Rob Cook, and Ed Catmull—whose software led to movies like *Toy Story, A Bug's Life, Jurassic Park, The Phantom Menace,* and even *Forrest Gump*

BY TEKLA S. PERRY
Senior Editor

Goes To...

Watching the movie *Toy Story 2*, the moment I truly forgot that I was looking at computer-generated images was when Tour Guide Barbie climbed into her car; to me she was simply an actress perfectly cast for her part. And when I watched *Jurassic Park*, I was too busy waiting for disaster to happen to distinguish between the computer-generated images and the real film footage. Nor did I wonder how, in *Forrest Gump*, Tom Hanks could play such an amazing game of Ping-Pong. (He was playing without a ball; the ball was computer-generated later.)

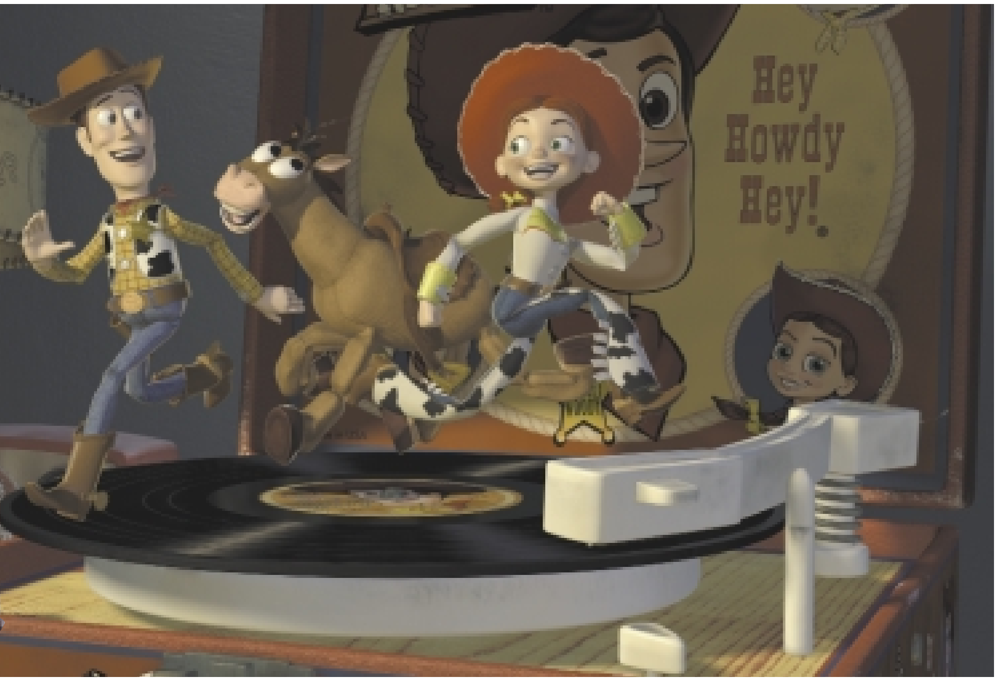
Today computer graphics blend seamlessly into the action of movies. But it wasn't always this way. It took a number of breakthrough developments to get from blocky, plastic-looking images to the realistic worlds generated today.

Computer-generated images are created in several steps: modeling, or describing the objects in mathematical terms that a computer can understand; animation, or describing how the objects move over time; and rendering, or determining which of the images defined by the modeling and animation programs are visible on the screen for every frame, assigning colors to those images, and drawing them.

Of the three, rendering is perhaps the most difficult. Early rendered images were simplistic, made up mostly of straight lines and polygons, and incapable of taking into account complex textures or subtle shading from multiple light sources.

Today's realistic rendering technology stems from the insights, separate and collective, of three computer scientists—Ed Catmull, Loren Carpenter, and Rob Cook. Their seminal work is packaged along with a number of other computer graphics tools into the RenderMan software, marketed by Pixar Animation Studios, Emeryville, Calif., and used by Pixar and many other movie companies.

Last month that work was awarded an Oscar by the Academy of Motion Picture Arts and Sciences, Beverly Hills, Calif.—the first Oscar of the new millennium—so perhaps it was fitting that



RenderMan made its mark in feature films in the mid-1990s with* Toy Story *and then* Toy Story 2 *[above], in which surfaces had never looked more realistic because they sported a variety of textures. For example, see the vinyl record, the horse made of felt, the leather-like lampshade [top, left], the "old paper" look of the poster [background], as well as the plastic faces of the toy characters, Woody [left] and Jessy [right].

LEFT: MARTIN KLIMEK/LIAISON AGENCY INC. TOP: ©DISNEY ENTERPRISES INC./PIXAR ANIMATION STUDIOS

it was also the first Oscar ever given to computer science.

But the development of the technology that became RenderMan started long before Pixar—or even the Lucasfilm computer group that spawned the company—existed.

Catmull's texture mapping

Early computer-generated images were all straight lines and angles, seemingly constructed of all-too-smooth-looking stacks of boxes stuck together. In the early 1970s, something called bicubic patches was used in an attempt to smooth out some of those sharp lines. This technique defines a surface using only a few points, instead of many individually specified polygons. The computer then generates the needed polygons from the points, allowing many more polygons to be created and allowing surfaces to appear smoothly curved.

But using bicubic patches caused a problem—it was difficult to calculate which surfaces were visible and which were hidden.

In 1973, Ed Catmull, a Ph.D. student at the University of Utah, in Salt Lake City, who was working with bicubic patches, realized that he needed a new kind of algorithm to define visible surfaces—those sections of the image that are not hidden from the viewer by other objects in the picture. He came up with the Z-buffer. The Z-buffer keeps track of the depth (or distance from the viewer) of every pixel on the screen. (Pixels, short for picture elements, are the dots that make up the images in a computer display. Each pixel is assigned one solid color.) The computer then can compare any new object being displayed with the depths of the current pixels on display to decide which pixels are visible.

The Z-buffer is now built into the hardware of all personal computers and video games. ("I get zip for that," Catmull said, "because at that time patenting wasn't part of our thought processes.")

The next step in developing lifelike images was projecting other images onto that surface so it would more accurately represent the material it was hypothetically made of. Catmull, still at Utah, first did this in 1973, putting a photo of Mickey Mouse on a single curved patch, and texture mapping was born. (With texture mapping, a picture of a block of marble can be wrapped onto a surface, or group of surfaces, for example, making a model of a statue appear to be made of marble.) This procedure made computer-generated images more realistic, because instead of simply "painting" the picture with simple colors, the colors of real surfaces could be captured and then attached to the appropriate objects.

Catmull left Utah in 1974 for the New York Institute of Technology, New York City, which had set up a pioneering research program in computer graphics, where he worked on animation, rendering techniques, and motion blur.

In 1979, Lucasfilm Ltd., then in San Rafael, Calif., a company well known for its special effects operation using traditional noncomputer modeling methods, hired Catmull to head its newly created computer division. (Lucasfilm is now in Big Rock Ranch, Calif.)

Ed Catmull

Date of birth: 31 March 1945

Birthplace: Parkersburg, W. Va.

Family: wife Susan, four children, aged 10, 20, 25, and 29

Education: B.S. in computer science and physics, 1969; Ph.D. in computer science, 1974; both from the University of Utah.

Graduate thesis: "Subdivision algorithms for the display of curved surfaces"

Key technical contributions: the Z-buffer, used today for defining what images are visible and what are obscured in all home computers and video games; texture mapping; and a technique for displaying curved surfaces called subdivision surfaces

What he wanted to be when he grew up: a Disney animator

First job: theater usher

First technical job: computer programmer for Boeing Co.

Present job: president of Pixar Animation Studios (known as Lucasfilm Computer Division when he joined it in 1979)

Oddest job: wine steward ("I don't drink")

Patents: a few

Computer: Macintosh G4 tower with a flat-screen cinema display

Favorite Web site: Amazon.com

Most recent books read: *Infinity Shore* (science fiction) and *Diophanpine Analysis* (mathematics)

Favorite restaurant: Phelplela, a Thai restaurant in Mill Valley, Calif.

Favorite movies: *Who's Afraid of Virginia Woolf?, Court Jester, Amadeus*

Biggest career surprise: "That I'm sitting here as president of a 500-person company"

Leisure activities: "What leisure?"

Carpenter's fractals

Meanwhile, Loren Carpenter was working in the computer-aided design group at Boeing Co., Seattle, Wash. He had long dreamed of seeing his beloved science fiction and fantasy stories displayed on the movie screen, and had transferred to the CAD group to gain access to the equipment that might let him realize that dream. So, besides displaying images of airplanes, he was thinking about how to get the Boeing computers to display realistic clouds and landscapes around those airplanes.

In 1978, Carpenter read Benoit Mandelbrot's *The Fractal Geometry of Nature*. The term fractal, coined by Mandelbrot, refers to mathematical shapes with fractional dimensions. The book had photos of landscapes generated by using fractal geometry. But, said Carpenter, the methodology explained in the book allowed designers to generate only small landscapes as if viewed from a distance, with a constant texture quality. A jagged ridge, say, would get smoother when the virtual viewer was moved closer because the world had a uniform and static level of detail.

"I wanted to be able to create [on the com-

MARTIN KLIMEK/LIAISON AGENCY INC.

RenderMan is used to incorporate animation with human actors in such feature films as* The Lost World *[top] and* The Hollow Man, *which used previously unavailable degrees of shading [see shadow on the bed beneath the skeleton, above]. But the most realistic animation so far is due later this year in* Final Fantasy *[right], for which special shading programs were developed to create computer-generated human beings.

TOP: UNIVERSAL PICTURES/PRESSLINK INC.; BOTTOM LEFT: COLUMBIA TRISTAR/
ZUMA PRESS.; BOTTOM RIGHT: COLUMBIA TRISTAR

puter] landscapes for my airplanes," Carpenter said. "But the method described did not allow you to stand *in* the landscape and have a level of detail in the background and foreground that was consistent with reality, to have a situation where the magnification factor can vary by 1000 in the same object. And that was what I needed."

So Carpenter, as he described it, "threw the problem back into my subconscious. I built a little structure of all the things I knew, leaving a little hole where the answer would go. And every once in a while, I looked into the hole to see if anything fell in. One evening when I looked, something was in there—a whole class of algorithms. Within two minutes of seeing that, I knew how to make lightning bolts, landscapes, clouds, and a host of other things that had an infinite variety of detail and scale and would also animate, because they had a consistent geometry. Randomness is easy, but making it stable and random is very difficult."

This class of algorithms is known today as procedural modeling. The subset that he used to make fractals is called midpoint subdivision. The mathematical concepts behind it are old, but the application to computer graphics was novel. Basically, in procedural modeling, the computer considers an object, defined in software, that is part of the picture it is trying to render and asks whether it might be visible or not to a camera.

If the object might be visible, the computer checks that it is not too complicated or too big to be drawn. If it can't be drawn, the object is then split into smaller objects, and the loop repeats. After an object is drawn, the calculations are thrown away, so the computer's memory use is minimal. Once a subset of the image is reduced to elements that are visible and can be drawn, it is displayed and the memory reused. The infinite variety springs from the methods of splitting and drawing.

Carpenter wanted this breakthrough technology to go out in the world and be used by people everywhere. "But," he said, "I knew if I told people about it, they probably wouldn't believe me, or it would at least take a long time before someone was willing to try it."

He also had heard that Lucasfilm was starting a computer graphics department, and wanted to be part of it. "But who was I, some programmer at Boeing, thinking I could go to Lucasfilm, when they could heat their building by burning all the résumés they were getting?" he said. So, he concluded, he had to make a movie, both to demonstrate his algorithms and to get Lucasfilm's attention.

Working evenings and weekends, using an engineering computer at Boeing, and shipping computer tapes 1500 miles away to a film lab where they were laboriously transferred to film, Carpenter spent four months during 1980 making the two-minute film, *Vol Libre*, in which the viewer flies through a range of mountains. This was the first time fractals were used to create realistic landscapes that viewers could, in effect, move through and see from different perspectives, not just observe from a distance. The final print was ready in August on the day Carpenter was to deliver a paper in Seattle at Siggraph, the most important annual conference for the computer graphics community.

When he finished showing the film, "the audience erupted," Carpenter said. "And Ed Catmull, then head of the computer division at Lucasfilm, and Alvy Ray Smith, then director of the computer graphics group, were in the front row. They made me a job offer on the spot."

Recalls Smith, "I hired him immediately because he had an efficient algorithm for rendering landscapes and other complex phenomena and because he also had a sense of mission, like the rest of us, to make movies and tell stories with technology."

Cook's light model

Back in the Northeast, Rob Cook was at Cornell University, in Ithaca, N.Y., studying for a master's degree in computer graphics. At the time, he recalled, nearly every computer-generated object looked like plastic. People didn't want them to look like plastic—they were trying to create computer images of all sorts of materials—but no one knew why they were unable to succeed. Cook decided to tackle the problem for his graduate project.

The problem turned out to be in the existing software model for reflective light, or, as it is also called, light-surface interaction. This model used the color white for highlights on an image (the brightest reflection of the light source on the image) and the color of the material for the diffuse reflection. "That turns out to be the perfect model of how light reflects off plastic," Cook said, "because plastic is initially white [and] is colored by mixing pigment into it. So the highlight comes from the light that bounces off the surface, the plastic itself, and the diffuse reflection comes from light that penetrates the surface and hits the pigment, which is colored." But for other substances, such as metal or wood, the highlight contains color; therefore, the model for reflective light needed to be changed.

Cook wrote a paper about his discovery, which somehow found its way to Lucasfilm. Much to his surprise, he got a call from Lucasfilm's Smith in April 1981 and was offered a job as a software engineer.

Said Smith, "I was assembling the top stars in computer graphics. Rob was the newest one,

Loren Carpenter

MARTIN KLIMEK/LIAISON AGENCY INC.

Date of birth: 7 February 1947
Birthplace: Brighton, Mich.
Family: wife Rachel, one child, aged 31
Education: B.S. in mathematics, 1974; M.S. in computer science, 1976, both from the University of Washington, Seattle
Graduate project: a new rendering algorithm for a very small machine
Key technical contributions: the A-buffer hidden surface algorithm, the rules to make procedural modeling practical, the midpoint subdivision method for rendering fractals
What he wanted to do when he grew up: "Something that nobody had ever done before"
First job: greenskeeper's assistant at a golf course
First technical job: computer operator for Boeing
Present job: senior scientist at Pixar Animation Studios (known as Lucasfilm when he joined it in 1981)
Patents: Nine
Computer: Silicon Graphics Octane (office), Dell Pentium 3 (home), Sony Vaio (laptop)
Favorite Web sites: cnn.com, drudgereport.com, google.com, http://mars.jpl.nasa.gov/mgs/index.html, http://near.jhuapl.edu/, and http://antwrp.gsfc.nasa.gov/apod/astropix.html
Most recent book read: *Deepness in the Sky* (science fiction)
Favorite restaurant: Guaymas, in Tiberon, Calif.
Favorite movies: the original *Star Wars, The Day the Earth Stood Still*
Biggest career surprise: "When my first fractal mountain came up on the screen"

established by his groundbreaking materials-modeling paper. We *had* to have him."

80 million polygons

Shortly after Cook joined Lucasfilm, the project that was to become RenderMan began.

"We wanted images that were complex enough to be used in feature films," Catmull told *IEEE Spectrum*. "We wanted the artists using the program to be able to control both lighting and shading. And we set a goal of being able to handle 80 million polygons per frame." Existing rendering programs were manipulating up to 100 000 of the polygons that a computer uses to describe an image. The 80 million number "seemed sort of crazy at the time," Cook recalled. But, said Catmull, "we wanted to pick a number so ridiculously high that it would make us rethink the process."

In 1980, Loren Carpenter made a two-minute short, Vol Libre, *to demonstrate his breakthrough method of rendering objects. For the first time, viewers could see a scene depicted consistently from different perspectives.*

LOREN CARPENTER

The rendering software, being written for a VAX 11/780, running at 1 MIPS, was to incorporate Catmull's work with texture mapping and Carpenter's work with fractals. (It now runs on a 14-processor Sun Sparcstation. With each processor running at 400 MIPS, that is 5600 times faster than the VAX originally used.)

Shade Trees

This rendering software was also to incorporate a physics-based means of defining how light reflects from surfaces—one that grew out of Cook's discovery concerning the need to change the existing light model. And Cook added something else.

At Lucasfilm, Cook had a collection of materials on his desk, including a block of wood. "I wanted to make a computer image that looked like wood," Cook told *Spectrum*. "Looking at the piece of wood, I could see that it was incredibly complex—the grain was a different material from the rest of the wood, which varied in color, and there were scrapes and nicks on it."

"I realized that you needed a general way of describing a surface that would take into account all these interactions (between the grain and the color and the scrapes, for example), and let you create complex, rich-looking stuff. What you really needed was the flexibility provided by a computer program."

Out of that idea came programmable shading. Cook created a special language in which to write such programs, called Shade Trees, both because it used a tree-like data structure and because it was inspired by a piece of wood. (That language was used for five or six years. In 1990, a standard interface specification for RenderMan was completed by Pat Hanrahan, now the Canon USA professor at Stanford University in California, and, as part of that specification, the Shade Trees language was rewritten. Cook's original language is considered an ancestor of what is used today.)

"Early on," Cook told *Spectrum*, "I thought these programs would be 10 or 20 lines long at most. But the Shade Tree program for *Luxo Jr.* [a pioneering short film made using the Lucasfilm Renderer] had 100 lines of code [see photo, p. 49]. And now these programs are hundreds of thousands of lines long, with all sorts of details, and there are separate programs for every surface in the scene."

"Making shading into a language was a genius move by Rob," Smith said. "He couldn't have foreseen how far the idea would go. Only a computational idea as large as a language could have handled the evolution."

A friendly competition

Developing the core rendering algorithm turned into what Catmull, Carpenter, and Cook refer to as a friendly competition.

The basic problem was how to determine, in a complex image of many colors, what color to assign each pixel. (The shapes of the objects are defined by the modeling software; the colors of pixels are calculated by the renderer.)

"We decided to try two approaches," Catmull said. "I took one approach, Loren [Carpenter] and Rob [Cook] took a different one, and we tried to outdo each other."

The approach taken by Catmull dealt with the screen image as a set of regions described to the computer as polygons. Each region is cut up into squares of a predetermined size, each of which will be represented by ne pixel. To determine the color of each pixel, the software needed to analyze each square, calculate how much of each color appeared in that square, create proportionally one single blend of the actual colors, and assign that color to the pixel.

This method eliminated the risk of creating jaggies—stairstepped lines that are anathema in computer graphics. "It was more hopeful," Carpenter said, "because it would give us precision on the edges of things. We would be able to tell that an edge covered 0.396 of a [little square to be represented by a] pixel, and could put in 0.396 of that color in the pixel, and not simply approximate. It had the potential of producing cleaner images."

But the processing requirements of such precision were immense.

Meanwhile, Carpenter and Cook were looking at images as sets of individual points within each square. "That made it a simple equation for the computer to solve," Carpenter said, "basically, 'What is the color here at this point?'"

To produce complex images of many colors, it was necessary to sample a few dozen points for each pixel and average the colors. The usual approach to this so-called point sampling was to pick points in a simple grid pattern. But this created annoying visual effects like jaggies and Moiré patterns.

Random points do the trick

To eliminate those effects, Cook devised what is known in computer software as a Monte Carlo technique. He introduced randomness. Picking random points eliminated the jaggies and Moiré patterns—problems in the image instead showed up as patterns of noise, a different effect, but still unacceptable.

"But," Cook said, "there is another way you can pick points. You pick a point randomly, and the next point randomly, but you throw the next one out if it is too close to the first point." This is known as a Poisson disk distribution, and is modeled on the distribution of cells on the retina of the eye, which also have a seemingly random pattern but with a consistent minimum spacing. It eliminated the annoying visual effects.

Fortunately, it simplified several other tasks that had to be handled by the rendering software.

One such task was effecting "motion blur." A real object moving, when photographed by a real camera, is blurred along the direction of the motion, whereas computer objects, unless appropriately altered, appear sharp even when animated.

"Creating motion blur was probably the single hardest problem we had," Catmull told *Spectrum*.

For motion blur, the computer has to consider, besides what colors appear at each point, what appears at each point over time. Using point sampling, Cook explained, the group was able to find a simple solution. "To every one of your random samples, you assign a random time between when a traditional camera's shutter would open and when that shutter would close." Averaging those creates a blurred image.

Nearly the same trick worked to simulate depth of field. A camera's aperture can be adjusted to make everything in the picture sharp or to allow only a certain range of images to be in focus. To simulate a lack of focus using point sampling, each randomly selected point is assigned a random spot on an imaginary lens.

Point sampling worked so well to solve so many problems that Catmull's more precise approach was dropped.

The revolution, Smith said, "was that a single algorithm solved so many of the classic computer graphics problems, including the really tough nut, motion blur. It was an exceedingly beautiful solution. Rob and Loren did a marvelous job, and Ed egged them on."

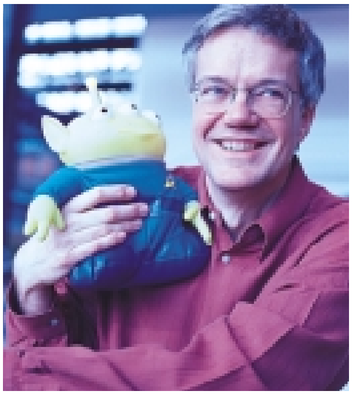
Once the core algorithms were settled, Carpenter laid out the architecture for the rendering software, basing it on an earlier program he had written called Reyes, with much input from Catmull and Cook. (Reyes is named after California's scenic Point Reyes National Seashore, one of Carpenter's favorite places to go to think, and is also an acronym for Renders Everything You Ever Saw.) Cook then wrote the final program, called, at the time, the Lucasfilm Renderer.

When the decision was made to market the software outside the company, a standard way of interfacing to other graphics programs was specified by Hanrahan, who had joined the team in the mid-1980s. That interface specification was tagged RenderMan. (While the Pixar product is widely known as RenderMan, its official name is something of a mouthful, "PhotoRealisticRenderMan, a RenderMan-compliant renderer.")

RenderMan debuts

Part of the software that was to become RenderMan was first used by Lucasfilm in the 1982 movie,

Rob Cook



MARTIN KLIMEK/LIAISON AGENCY INC.

Date of birth: 10 December 1952

Birthplace: Knoxville, Tenn.

Family: wife Mary Anne, one child aged 7

Education: B.S. in physics, 1973, Duke University, N.C.; M.S. in computer graphics, 1981, Cornell University, Ithaca, N.Y.

Graduate thesis: "A reflection model for realistic image synthesis"

Key technical contributions: physics-based reflection model, Monte Carlo ren-dering, programmable shading, Ofoto scanning software

What he wanted to be when he grew up: a doctor or an astronaut

First job: working at Weigel's, a convenience store

First technical job: assaying gold ore in a chemistry laboratory

Present job: senior scientist at Pixar Animation Studios (known as Lucasfilm when he joined it in 1981. He left in 1989, returned in 1997)

Oddest job: computer programming in the intensive care unit at the Hospital of the University of Pennsylvania

Patents: four

Computer: Dell Inspirion 700, Silicon Graphics Octane

Favorite Web sites: my.yahoo.com, rottentomatoes.com, amazon.com, nytimes.com

Most recent book read: *Poisonwood Bible* and *Guns, Germs, and Steel*

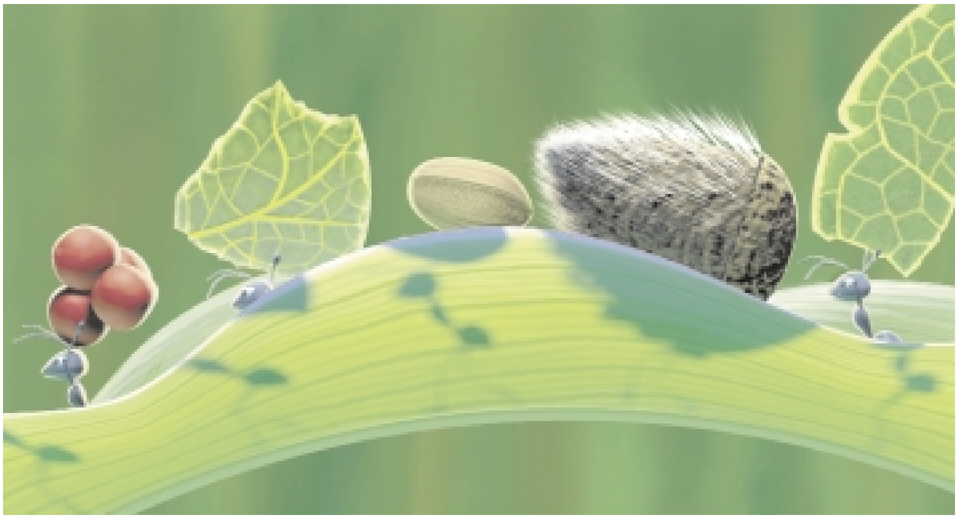
Favorite restaurant: Azie, in San Francisco

Favorite movies: *North by Northwest, The Dead Zone,* Kenneth Branagh's *Henry V, Dark Star*

Biggest career surprise: getting a phone call from Lucasfilm

Leisure activities: playing the piano, sailing, hiking, reading, sculpture

Car: BMW M3 convertible



The first publicly shown film to use Lucasfilm's Renderer was the 1984 short, The Adventures of Andre and Wally B*. [top, left]. For more realism, the first Pixar short,* Luxo Jr. *[top, right], in 1986, had a 100-or-so line shading program. More realism still came in* A Bug's Life, *in 1998 [right]. Its shading programs had hundreds of thousands of lines.*

TOP PHOTOS: PIXAR ANIMATION STUDIOS; BOTTOM PHOTO: ©DISNEY ENTERPRISES INC./PIXAR ANIMATION STUDIOS

Star Trek II: the Wrath of Khan, for what is known as the Genesis Sequence. The first publicly shown film that used the completed Lucasfilm Renderer was *The Adventures of Andre and Wally B.*, which debuted at the 1984 Siggraph conference in Minneapolis. At the 1986 Siggraph, in Dallas, the short RenderMan-enabled film *Luxo Jr.* was shown and, as this writer can attest, simply blew the audience away. Said Catmull, "It wasn't about computer animation anymore, it was just this little lamp that made everything else irrelevant." The RenderMan images were now complex enough to be feature films themselves, not just obvious special-effect sequences. The goal had been met.

Meanwhile, in 1986 the Lucasfilm computer graphics division spun out into a separate company that became Pixar. The company's first full-length feature film to use RenderMan, *Toy Story*, was released in 1995.

Today RenderMan is widely used by movie studios. In the last 10 years, 8 out of the 10 films that won the Oscar for Best Visual Effects used RenderMan—*The Matrix, What Dreams May Come, Titanic, Forrest Gump, Jurassic Park, Death Becomes Her, Terminator 2*, and *The Abyss*.

The Oscar

So on 3 March 2001, the first movie-year-2000 Oscar went to "Rob Cook, Loren Carpenter, and Ed Catmull for their significant advancements to the field of motion picture rendering as exemplified in Pixar's 'RenderMan.'"

And this, Carpenter told this writer just before the ceremony, makes RenderMan a piece of history. "A hundred years from now," he predicted, "people will be making films with computers, and they won't remember how it was before. This is when it essentially started."

But it won't end here. In fact, Carpenter and Cook and others at Pixar have started work on the next generation of rendering software; the original RenderMan is now some 15 years old. Cook has already reportedly written a prototype testing some new ideas. They again have thrown out all the previous assumptions about rendering, and are starting with a blank slate. They aren't even assuming, Carpenter said, that the next-generation software will break images into pixels. "Pixels were designed assuming computers had very little memory," he said. "We are throwing that assumption away to see what happens next." ●