Doug Roble: Computer Vision, Digital Magic

Doug Roble

Interview of Doug Roble by Y.K. Leong

Doug Roble is world renown for his important contributions to computer vision and computer graphics and for pioneering applications to movie special effects and animation.

Roble did a joint degree program in engineering and computer science at the University of Colorado and went on to Ohio State University (OSU), where he did his PhD in computer science (on computer vision). He was an assistant faculty at OSU for a year before joining Digital Domain in Venice, California as a software engineer in 1992. Expanding on his PhD work, he developed the 3D tracking software TRACK for camera position calculation and scene reconstruction. This helped artists determine where to best fit graphics into images that have been filmed. For this software, Roble received a Technical Achievement Academy Award (Academy Certificate) from the Academy of Motion Picture Arts and Sciences in 1998. His subsequent work in the development of fluid simulation system earned him, together with Nafees Bin Zafar and Ryo Sakaguchi, a Scientific and Engineering Award (Academy Plaque) in 2007. This work allowed graphic artists to create large scale surging water effects for the movies *The Lord of the Rings: The Fellowship of the Ring, The Day After Tomorrow* and *Pirates of the Carribean: At World's End.*

He has been the Creative Director of Software at Digital Domain since 1993. He is the Chief Editor of the Journal of Graphics Tools and is on several panels and committees of SIGGRAPH (Special Interest Group in Graphics), the most prestigious computer graphics conference, including its Advisory Board. He has given invited lectures and keynote addresses at many major conferences, most recently at the Annual Meeting of the American Association for the Advancement of Science in 2007. He received the Distinguished Alumnus Award from Ohio State University in 2002. He is a voting member of the Academy of Motion Picture Arts and Sciences, Visual Effects Branch.

Roble was one of 4 invited speakers at the Symposium on Mathematics and Science in Digital Media, Technology and Entertainment held at the Raffles City Convention Centre on 1 July 2007 and organized by the Institute jointly with the Department of Mathematics, NUS. The symposium was supported by the Media Authority of Singapore to introduce and publicize the new field of interactive digital media to the general public. *Imprints* took this opportunity to interview Roble during the symposium. The following is an edited and unvetted version of the transcript of the interview in which he spoke with passion and animated enthusiasm about his early work on computer vision and its subsequent breath-taking impact on digital media and the entertainment and movie industry.

Imprints: Your B.S. was in electrical engineering and computer science way back in 1984. Was it some kind of joint program or major?

Doug Roble: It was a joint program actually because there was no full computer science degree in the University of Colorado at that time. The only way to get into computer science, which I knew I wanted to pursue, was to do an electrical engineering and computer science degree. Also I wasn't sure what I wanted to pursue – I knew that electrical engineering was interesting as well. It turned out to be a good thing because an electrical engineering degree offered much more math than a typical computer science degree, especially at that time. Computer science at that time sort of required you to have linear algebra, maybe a little bit of Boolean math, but electrical engineering gave me a good foundation in calculus, multivariate calculus and signal processing. I've come to use a lot of those basics much more than I thought I would. So it was a joint degree, a sort of double major.

I: Was a double major common in those days?

R: Kind of. Computer science was an emerging field. If you wanted to get a computer science degree, that was the way to do it. There was no pure computer science degree at that time, as I recall it. I think it was only a couple of years later that the University of Colorado had one.

I: What attracted you to Ohio State University subsequently to do your graduate studies in computer science?

Continued on page 16

Continued from page 15

R: Remember that, way back in 1984, Ohio State University had, and still has, a very strong graphics program. Back in 1984, they were associated with a company called Cranston/Csuri Productions, Inc which was a pioneer in computer graphics. They did a lot of the first commercials using computer graphics. In fact, they had done this famous commercial with a very shiny robot woman talking about the beauty of canned food. It was associated with Ohio State University. It was one of the pre-eminent computer graphics school at that time. Stanford hadn't even started its computer graphics program at that time. Ohio State was doing it, so I wanted to go there.

I: Did it ever occur to you to take up engineering instead?

R: Well, no. After my bachelor's degree, I used that time to figure out what I wanted to do. I knew it was computer graphics. I was fascinated with what I saw was happening with the movies. Remember it was 1984. *Star Wars* and *The Empire Strikes Back* had already come out. *The Return of the Jedi* had just been finished. I wanted to do that. *ET* and all those great films that were using traditional effects could have used computer graphics as well. This is what I wanted to be doing.

I: After your PhD, you were in academia for less than one year in OSU. Was it a calculated plunge to go into the digital industry at a time when digital media was at its infancy?

R: Absolutely. I knew I didn't want to be in academia. I wanted to work in films. It was a good opportunity because it allowed me to jump into the beginning of the bit of the domain. The company [Digital Media] had just been formed. It opened its door in 1993 and I was its 31st employee hired. So I was right there at the beginning, and it was a bite of a bullet, and I was a bit scared that the company might not last very long. But it did work out fine.

I: It must be quite fun to start at the beginning.

R: It was. It was amazing. It was crazy.

I: How recent is this discipline of IDM [interactive digital media]? How do you define it?

R: When you put the word "interactive" in front of "digital media", it becomes a whole different thing. Interactive digital media tends to mean games graphics, maybe even visualization. I'm in digital media, not so much interactive digital media, which really started in the nineties with video games and things, when *Doom* and the first 3-D games came out. Now movies and games are coming closer and closer together. It's all so crazy. I tend to define interactive digital media as that where in response to the user's input, something on the screen changes. I work in films where it doesn't matter what the user is doing. The user can leave the room and it still gets projected on the screen. The problems we are trying to solve are vastly different. Most of the things we render take ages and ages to render, from an hour to 24 hours. With video games you work with 60 frames a second or you're in trouble.

I: Some time ago there was some kind of movies where the audience actually participates in choosing what is going to follow.

R: There was this little teeny experiment where you get to choose between (1) and (2) endings. It was just an experiment. Maybe it will change some time but you don't want the majority to win. I think much more likely you will have DVDs where one person gets to choose the plot rather than the majority who's watching a film. It's democratic but that doesn't make sense.

I: Could you tell us something about your most exciting research work? Is your PhD research related to your later research work in industry?

R: Indeed, my PhD work was in trying to use computer vision to help computer graphics. I took the basics of that and re-did it for Digital Domain when I first started the program *Track* which is a computer vision toolkit that basically allows artists to look at an image in some film and extract as much information as possible – where the camera was, what the scene looks like, all the 3-dimensional information you can possibly get from a photograph. This has been something that I've been working on for the 13 years I've been there. I continuously backtrack to add new features to it. It got me the Academy Award. That was the best thing that could have happened and was probably my most exciting research work although the third generation stuff that I am doing is very, very cool. It's such a visceral feeling when you see things that are flowing like water and look like water. It's very fun. Right now, I'm looking at all sorts of stuff. One of the things that interest me right now is hair. That's probably my most current research work along with other people at Digital Domain. By the way, nothing happens by yourself. It's all part of a team. There's a group of people working on hair and it's fascinating.

I: What do you mean by working on hair?

R: Well, hair is a big deal. There are three aspects of hair. There's modeling hair, styling hair – putting hair on some head or body so that it looks like a human head. Once it's in that position, you want to animate it, simulate it so that when wind is blowing or when somebody runs his hand through the hair, the hair moves correctly. And third, render

Continued on page 17

Continued from page 16

hair. How do you render hair with a hundred thousand very thin little strands that are semi-transparent and light bounces off it in a very interesting way? How do you render it so that it looks correct? By the way, everybody has different hair than you do. It's all very different.

I: It sounds extremely computationally difficult.

R: Exactly. It's a huge computational problem – a hundred thousand strands of hair, each one continuously curving – simulating them in a discrete fashion is very difficult. So you have to make simplifications and adjustments that are good enough so that the audience is fooled.

I: Do you actually apply the laws of fluid mechanics to the motion of the hair? It sounds incredible.

R: Absolutely. It's hard – it goes back to mathematical archeology, things like Cossart curves, ringed together in a chain to represent continuous curving segments. This was invented back in the 1930s to deal with bending objects. Hair is wonderful, it doesn't stretch though. You got to make sure that the hair does not stretch and building that into the math is important.

I: Did you have to invent some new concepts or techniques to resolve some of these issues?

R: We're working on it. A group in France has laid down the foundations. We're trying to take some of their ideas and modify them so that we can use them. We're not finished yet. Maybe ask me again in a year. Indeed, we invented some cool stuff.

I: Has anybody written some kind of foundational textbook on such things?

R: Not so much. Actually there is a very good researcher in Switzerland, named Nadia Magnenat-Thalmann. She's been working on clothing and hair for her entire life, and has been doing some very good research answering some very good questions. She has a paper in a book on hair and clothing simulation. It's a computer graphics book with a lot of math in it.

I: Were there any IDM problems that contributed significantly to the development of any area in mathematics or computer science?

R: Ah, things that feed back into computer science and mathematics. Absolutely. From me, not so much. I haven't really had a lot of impact outside... well, some of the stuff I have done, things like fluid simulation. We started looking at fluid simulation – basically, the computer graphics community, not just me – in terms of how to create water that look realistic, which was a completely different approach from what computational fluid dynamics people were doing. They wanted to model water or fluid in a very precise way. Towards that end, they had to simplify the problem constraints because you can't model water realistically if you've got a very complex domain. We looked at it from an entirely different angle. We didn't care that much if it was totally realistic, but we wanted to put it in a very, very complex domain indeed. We have arbitrary boundaries, moving boundaries and all those stuff, and we wanted water to look real. We stood on the shoulders of a whole bunch of computational fluid dynamics rather than feeding back into it because we approach the problem in a totally different way. So we have attracted some of the attention of pure fluid mechanics people. And they said, "Oh, you're doing it that way. That's very interesting." So, Stanley Osher, Tony Chan invented level sets. I don't even know if they realized how important it was going to be. They took it and applied it with Ron Fedkew to fluid, and this is a brand new field. It's a new way of doing it and the computer graphics media have adopted level sets to do all sorts of amazing things, and that has gone back into the mathematics. I think that's one example.

I: What about the classical Navier-Stokes equations? Any contribution to it?

R: We use a subset of the Navier-Stokes equations. The ones for inviscid fluid are pretty puzzling. For the ones we do use, we are trying to push solving them faster and faster. Also, we have pushed ahead trying to capture the details. Whenever you are solving the Navier-Stokes equations numerically, there is a lot of filtering going on. You always lose details. All this stuff get lost in the mathematics of fluid. We recently (when I say "we", I mean the computer graphics teams – people in Berkeley and Stanford are really leading the way) are coming up with ideas of putting back the detail into the fluid simulation so that the detail isn't lost, or if we do lose it, we put it back in a possible way so that it looks good. The goal is always to try to render water that looks exactly like water. Water is very non-viscous, and that kind of fluid simulation is very hard to do. We're getting closer; we can do milk. Milk is easier, it's viscous and doesn't have all the sharp edges that water has.

I: Are creative computer programming skills necessary for a successful career in IDM? Can such skills be taught to any beginning mathematics graduate student?

R: Sure. Thinking in terms of math is very similar to thinking in terms of computer programming. When we hire new people, no matter where they're coming from, we expect that they know how to write code. Teaching programming

Continued on page 18

Continued from page 17

to mathematics students is such an important part – using the computer nowadays, no matter what. That's just Mathlab or Mathematica. But having to write your own piece of software, to implement something you have done, at least in applied math – I don't see how you can get by without it. It gives you that ability to say, "Oh, I wonder if I can write this as a code and see it on a grand scale."

I: There are people who seem never to be able to write computer programs.

R: And they can do proof? What? No. Computer programming is easy, come on.

I: Certain computer programming is not as straightforward as proof and can be quite tricky. Don't you think so?

R: I don't believe that. When it was first invented, it was tricky. You have to be very careful. When I was first doing computer programming, I did it on ancient machines. You have to do it in assembly language and it was all very arcane. Now there are so many tools available in computer programming. Modern C++ languages or Java or any kind of programming language gives you so many advantages that once you learn the basics of looking at a problem – you learn iterative, looping statements, recursion, you learn how variables work – all of a sudden, you are writing code.. All these things are mathematical concepts. It's not hard. You do use them when you are doing proofs.

I: The older generation of mathematicians somehow or other seems to loathe computer programs. There's a perception that computer programming is a young man's game.

R: You can do it, you just don't want to. What younger generation writing codes. I won't buy it. Certainly young men are doing it. But you can do it. No excuses. You're being lazy [*laughs*]. If you are good at math, you can do computer programming.

I: Reconstructing a three-dimensional object from a two-dimensional image like a photograph seems amazing, if not unbelievable. Is it theoretically possible for known methods of reconstruction to fail in at least some contrived cases?

R: Oh, absolutely. If you just have a single image of a 3-dimensional scene (you just take one photograph), it's impossible to figure out what's going on. There's no way, without any extra information, to know the 3-dimensional nature of that scene. Even if you have multiple photographs of some scene, where you can do triangulation and the various computer vision techniques to figure out what's in that scene, there are still things like scale invariance. There's no way of telling whether the photograph of the fire truck you are taking a photograph of was a real fire truck or a toy fire truck. You can't tell the difference without some measurements you actually took at the scene.

I: How many photographs do you need to reconstruct a solid object?

R: You usually get away with two. Three helps. Of course, you can always reconstruct the object that you see, you have to have coarse refinement between the two. If the camera didn't move very much, then there are limits to the accuracy because the pixels are a discrete measurement of the world and there is a built-in error. Computer vision is all about managing the error. So if you identify a feature within an accuracy of a pixel or two, and then you move the camera only a little wee bit, then the error involved in that feature identification overwhelms the mathematical induction that you can do. So the result that you get is not so good. But if you have a couple of, or multiple, photographs with decent baseline, then you can do amazing reconstruction nowadays. But you have to be able to identify a correspondence between a feature on one photograph and the feature on the other photograph. So if I take a picture of a chair, and then I move the camera and take another picture of the chair, there are parts of the chair I can't see that I could see in the first. So those pixels are fine and I have to infer the details and kind of make them up.

I: This must have been applied in astronomy.

R: Oh, of course. That's exactly how they determine how far the stars are. You use a telescope and wait a couple of days for the earth to have moved, and then you get a very long baseline and you can use triangulation.

I: The first animation in films was based on a frame-by-frame representation. How does the current animation in films differ qualitatively, and not just quantitatively, from those earlier ones?

R: Well, first of all, your question ignores one of the aspects of cinematography of the early animation. It wasn't frame by frame representation; it was frame by every other frame representation. If you go back to old Disney cartoons, hand-drawn Disney cartoons, because it was such work to animate every frame, to draw picture for every frame, Disney said, "Okay, we don't have to do that; we will only animate every other frame." Animation on two. If the motion is very rapid, sometimes you have to draw animation on one, where you actually draw a separate image for each frame. If you look at the old Walt Disney films that were poorly animated, you can see the difference. If you step through it on a DVD, you will see the fact that the images were held for 2 frames and then they move to the next frame. So right after that, qualitatively and quantitatively, we now

Continued on page 19

Continued from page 18

animate every frame. Because the computer is doing it, it's pretty much as easy to do it on one as it is on two. There's no advantage doing on two. In fact, it looks smoother. But other than that, the technology of computer graphics has to do a lot of other things that you could not do on a hand-drawn animated way like motion whirl. Every film that is done nowadays renders not only the image but the image as it is moving, and so you get motion whirl. Without that, it looks very harsh and rigid. You pick it up and say, "Oh, it's fake. That looks like a computer-generated thing." With motion whirl, that makes it look more real. Unfortunately, motion whirl is very expensive. It's a sampling problem. It's a big signal-processing problem where now not only do you have the image rendered, but now you have to move it through time in order to get that broiled. And time is very continuous. Whenever you hear the word "continuous", you go, "Ooohhh, it's continuous, there's a lot of data there." So figuring out how to do motion whirl exactly in a reasonable amount of time is a tricky thing that we work on all the time.

I: IDM is becoming visible in many fields other than the entertainment industry, like medicine, robotics, telecommunications, geography and architecture. Will the coming future of IDM depend largely on advances in engineering and technology, for example nanotechnology?

R: Nanotechnology? Maybe not so much. Certainly, if you really think about it, the latest chips from Intel and AMD are very much nanotechnology – they are cramming 4 full-blown processors in a single chip. That has a huge impact – the fact that we can do things in parallel. A lot of what goes on in computer graphics is embedded in what we can do in parallel. This is very good for us. We love the fact that processors are now becoming multi-processors all the time. We buy the latest things from Intel with 8-core or quadcore quad processors. We get 16 processors in the machine. We immediately jump on it and start using it. That's nanotechnology that has a direct impact on us. Other than that, there isn't much nanotechnology. Quantum computing, maybe. There have already been some theoretical uses of quantum computing for computer graphics.

I: Does that mean you have to develop new techniques of software or computational methods?

R: After we had third-generation computer graphics and computer vision, both have adjusted themselves to take advantage of the parallelism of processors. It's a big thing. The artists love it because it makes everything faster. It's not hard to do. With fluid simulation, it's tricky because you need to write optimization parts of the fluid simulation like conjugate gradient, preconditioned conjugate gradient, and other tricks. Once you've done it, then all of a sudden, your fluid simulation comes on.

I: Do you give courses on digital media? Do you have any students?

R: The courses that I give are typically SIGGRAPH courses which are one or two-day courses at annual conferences. People who sign up go there to be educated. In terms of students, for courses, no, but we do have internships. We have interns, Masters or PhD interns, people who come here constantly. Right now, I'm working with a student of Tony Chan, UCLA and there's also a student from a Swedish University working on hair simulation. Yes, there are students. Internships usually last about 4 months. Either I or another one of our R & D staff is the advisor of these guys.

I: What is your advice to a mathematics student who wants to have a career in your field?

R: If you are in applied math and you know how to program, I suggest you look at the last couple of years of SIGGRAPH's proceedings. SIGGRAPH is the pre-eminent conference on computer graphics and usually there are 90 to 100 papers accepted to the conference each year. Take a look at those and see what kind of mathematics is currently going on. At this conference we have Peter Schroeder talking about differential geometry, very hard, very cool stuff, and you will immediately get a sense of the kind of mathematics that is useful to computer graphics and interactive digital media. I'd just focus on that kind of stuff. Going back to what we talked about programming, if you want to work in our industry, you have to know how to program. We don't hire pure math people who just sit and do math. You have to come up with an idea and make a tool that the artists could use, and then the artists... that's the best part of the whole deal, especially in the film industry. There's that lovely feedback when you're working hand in hand with some very creative artistic people. You create something and they will immediately turn around and use it the way you haven't thought to do and they give you a new idea. And you say, "Okay, I'll be taking that back and use it differently." It is so satisfying, it's much better than writing a paper and summing it up in a journal and getting some people saying, "Oh, I saw your paper." This is writing something where people are immediately using it to create brand new things and entertain people and show people new concepts. That immediate feedback, sitting next to the artists and they have great ideas from a whole different perspective than what I can offer. That's the best part of the job. Just that constant creativity from all different sources – we have mathematical creativity, some people read a new paper and say, "Look at this new technique" and then they go and show that to an artist, and he said, "Oh, look, I can do that, it's so fun."