

Keynote Address



Walt Bransford, SIGGRAPH 98 Conference Chair:

It is an honor, indeed, to introduce Jim Blinn, from Microsoft Research, as the SIGGRAPH 98 Keynote Speaker. For almost three decades, Jim has coupled his scientific knowledge and his artistic abilities in the advancement of computer graphics. His many contributions include

- the Voyager flyby animations for space missions to Jupiter, Saturn, and Uranus
- the animation of Carl Sagan's PBS series, *Cosmos*
- the PBS education series, *The Mechanical Universe* and *Project: Mathematics!*

Jim has been to every SIGGRAPH conference. He is the first recipient of the Computer Graphics Achievement Award. He developed many fundamental techniques, including bump mapping, environment mapping, and blobby modeling. As Graphics Fellow at Microsoft Research, he is investigating the mathematical underpinnings of image rendering.

Now, there are a lot of Jim Blinn stories, and I'll tell you what mine is. It was at a session in the 1980's when he was explaining this bug that had ruined one frame in an animation sequence. So he showed us how to fix the bug by holding up the film and cutting out the frame with a pair of scissors. And then that's when I knew, well this is the group for me!

Please welcome, Jim Blinn.

[*applause*]



Jim Blinn, Keynote Speaker:

So, 25 years.

Since I've been to all 25 of these things, they seem to think that I would have something useful to say at the keynote address here. So, in trying to

figure out what I was going to say, one of the things that occurred to me was what my friend Alvy Ray Smith always complained about people saying in situations like this, and I resolved that I'm not going to say that.

[*laughter*]

If you want to know what that is, you're going to have to ask him.

I'm not the only 25-year person here, although there is not an official record kept. I understand that Lou Katz and Jim Foley have also been to all 25 of these. Tom Wright was a member of the club for a while, but I'm not sure if he's still here. And there might be others that I don't even know about, and if so, let me know afterwards and I'll apologize for not introducing you at this opening.

Anyway, so what am I going to say? Well, in the first place, to believe that I actually went to all 25 of these things, probably I should go back and give kind of my little historical retrospective of all the conferences. This is decorated by scans of the covers of probably one of the very few complete collections of SIGGRAPH conference proceedings.



1974, in Boulder, it was held at University of Colorado. We all stayed in college dormitories instead of luxury hotels, and we had assigned roommates. I don't remember who my roommate was at the time, but I remember he was one of the speakers, which I was very impressed with, because that was before I got to the point of presenting anything myself. The conference itself: One of the first papers of the conference was Fred Parke giving a presentation of his talking face, which research he did at [University of] Utah. And now we have an entire session on faces, but this is kind of roughly what his talking face looked like.

There's actually something that I'm a little disappointed in face research, in that it's got its own version of the teapot, which they've kind of missed. Fred Parke's animation with talking face was reciting a poem by Emily Dickinson. It goes something like,

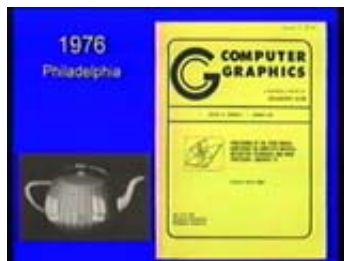
How happy is the little stone, that rambles in the road alone,
and never cares about careers, and exigencies never fears.
and... etc., etc., [Emily Dickinson stuff](#).

And it seemed like, this would be a good thing for all the face researchers to try to reproduce and see the evolution of how convincing that is in time. Maybe they can pick up on that and do it from now on.

This is also the conference I first saw Alan Kay give his presentation demonstrating the Xerox paint program. I didn't know Alvy and Dick Shoup at the time, but I was very impressed by the demonstration, and I went off and did something based on that a little later on. And Alan Kay was just sort of sitting in shorts and a blue shirt and just kind of hopped up and sat on top of the table in front of the room and gave this talk. So things were a little less formal than they are today.



1975, Bowling Green. This is where I met Nelson Max and we agreed to collaborate on some pictures of his everting sphere thing. He actually emailed me the control points of this everting sphere over the Internet, which existed in 1975, and that was kind of a fun project. Also the film *Hunger* was shown. I think that was the entirety of the film show that year.

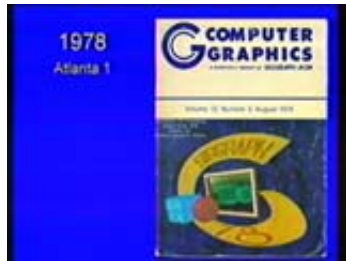


1976 happened during my stay as a summer employee at the New York Institute of Technology, so we all kind of piled into cars and drove down to Philadelphia. This is the year that I showed the first teapot paper with the reflections on it. And also the film show at that time was still pretty impromptu. We just ran off a video tape of the teapot being rendered the night before driving down there, and when they asked for submissions during the show, I just hopped up on the stage and plugged it in and narrated it live. Also, again things were pretty informal, I remember Ephriam Cohen gave his paper by just kind of getting up and scribbling on the blackboard in the room rather than even having any prepared slides. This is the conference I met Steve Levine at, who encouraged me and Martin Newell to put together a course for the next year in San Jose.



This is the cover that was designed by Maxine Brown, in fact. But this was also the issue that had the first color pictures inside the Proceedings. This [image] is done by Don Greenberg and his crew, I didn't do that one myself, but it's a sample what the color images looked like at the time. So, Martin and I put together this course on, as Steve Levine put it, the course was supposed to contain "how you do all that cool stuff that you guys do at Utah." Which we did. And this was the year that I met Andy van Dam who came up to me in one of the course receptions and complained bitterly about how crappy the course was. [*laughter*]

Um, this is also the year that the film show died in the middle because the power went out. Also during Martin's talk, one of his slides got jammed in the projector and some helpful person from the audience came up to try to unjam it and turned the slide tray upside down to get at it. [Dumping all his slides on the floor.] This is also the year that I met Turner Whitted. We both discovered we were working on similar problems of rendering patches and we got into such an excited discussion at the back of the auditorium that somebody came around and told us to shut up.



The next year, '78, the first time it was in Atlanta. And this is the year that SIGGRAPH was held concurrently with some sort of fashion show and the meeting hall that we were supposed to be in was preempted and they kicked us into, I don't know, a parking garage or something with very low ceilings. And so they had to go around and make three or four duplicates of everybody's slides so they could have several slide projectors around the auditorium, simultaneously, so people could see it because they didn't have a big throw that they expected for the real slide projector.

SIGGRAPH had to work its way up to be respectable in the computer exhibition business. This is the year that the first color picture [cover] came out which Dick Phillips asked me to design. It was the year that I talked about bump mapping, and, in fact, I got Lance Williams and Gary Demos on both ends of the coast to help me a little bit on the rendering of that. This is also the year that I brought a roll of exposed film to the conference, hoping I'd be able to get it developed in time to give my talk, and, in fact, I did, and that's what the picture of the orange was. This is also the year that I met Pat Cole, who later came to work with us at JPL.



The next year, the first time it was in Chicago. This is the first time I showed the Voyager fly-by movies by Jupiter, and, again, things were sufficiently informal that I just kind of brought it with me to the conference thinking there'd be some place I could show it. Then they were looking for things to put together the film show at the last minute, and they stuck that in, as well as other things. Also, this year, Pat Cole did a film and video retrospective of earlier computer graphics films. I think is interesting that, even in 1979, there was enough stuff and enough interest in the history to put together retrospectives, even then.



1980, in Seattle. This is the year I remember, one of the things, was that I forgot to pack socks when I went onto the plane, so I wore the same pair of socks for the first half of the week until I found a place where I could buy some more. This is, of course, the year that Loren Carpenter showed his movie, *To Fly*, which was the hit of the show there. It was one of the first fractal mountain animations, and it was kind of one of the first pictures that I looked at and said, "that can't be a computer image, that looks too good!" I'm easily fooled, I guess, but it did look too good at the time, but, nowadays, you can look at it and, you know, sort of tell because your expectations are increasing.

That year the film show, actually, didn't have enough room for everybody, so they actually had two showings of the film show consecutively in one night. The second showing, since this is all kind of done live and people narrated their things live, we sort of abbreviated. I kind of felt sorry for the people who came to the second show because they only saw half of it because everybody was too tired to do the [whole] show the second time.



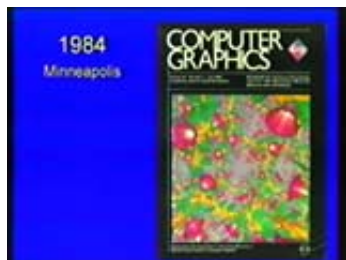
1981, in Dallas. This is when I first met Yoichiro Kawaguchi, who showed me a book that he'd published with some of my pictures in it. And this is one of the first times that we had an art show that had images viewed directly onto a framebuffer display. Usually, computer graphics images had been photographed and then you displayed the photograph of them. But photographs always looked washed out or the wrong color, or something like that, and when you're producing the pictures, you're looking at the display, so I figured seeing the actual display in the art show would be a great idea and so James Seligman helped me out with that. And, primarily, the images were ones that I had produced and that David Em had produced. We felt a little bit self-conscious about that, but those were ones we could get at easily (rather than trying to promote only our images.)



Anyway, 1982, in Boston. Showed the Voyager-Two Saturn movie there, and this is when I did the blobby paper and some cloudy light reflection papers. This is where I met Vibeke Sorensen, who later came and worked with us at the Art Center [College of Design].



1983, this is the award year. First time they gave out awards, and I was happy to share the honors with Ivan Sutherland for the two awards they gave out. Also, the framebuffer show, we had one there, wasn't working very well, so I spent like an entire day behind the curtain reprogramming the thing to set the [color look up table] registers properly for the framebuffer so it would look good.



1984, in Minneapolis. This is when I showed the first *Mechanical Universe* excerpt/demo, which was the physics series. This was about

two hours worth of stuff I'd done that year, packed into a three minute little movie-trailer version with a few appropriate comments. It was my first attempt at comedy, and fortunately people recognized as such. In fact, a lot of people came up to me afterwards and said, "gee, Jim, I didn't know you had a sense of humor." [*laughter*] This is also when *The Magic Egg* was shown. This is the collaborative effort of a lot people doing an Omnimax movie because they had an Omnimax theater in Minneapolis. This is the year that Tom DeFanti made up a bunch of kind of fake ribbons saying, "SIGGRAPH 85 - 89 Attendee." And was kind of the beginning of the ribbon phenomenon.



1985, in San Francisco. Showed more *Mechanical Universe* images, and this was the year that they had the course on image rendering tricks where I demonstrated my skill debugging hardware [that Walt mentioned in his introduction]. And this was the year that Rob Cook handed out ribbons that said "Party Jury" on them. That was to help us to get into all the parties, you see.



Second time it was in Dallas in 1986. This is the year that I saw, what I think, is still one of the most incredible computer graphics films of all time, by John Lasseter, called *Luxo Junior*. Afterwards, I went up to him and asked him, "gee, John, this lamp thing here, is that the mother or the father?" But I don't remember which one he said it was. I think it's up to your own imagination.



1987, Anaheim, first time. I did a course in 1987 on the production of *The Mechanical Universe*, which is first and only one-man show that I did for an entire day. And, unfortunately, I was so exhausted from three

weeks worth round-the-clock, making five or six hundred slides for this that I practically collapsed on stage in the middle of it. But, I made it through. This is also when they finally came up with something useful to say about ray tracing in Paul Heckbert's *Ray Tracing JELL-O®* paper.



Second Atlanta conference. Showed excerpts from a production about the Theorem of Pythagoras that I'd been working on. We were still looking for grant money to produce a series on high school level mathematics. We didn't have much interest, at the time, from the US, but we had some interest from some people in Japan. And so, in 1988, there was a big protest organized by Tom DeFanti and Todd Rundgren saying, "Keep Jim Blinn in America," where they went around wearing signs in front of the film show. And it worked because SIGGRAPH gave me a nice grant as seed money for *Project: Mathematics!* and we went on and produced it.



1989, in Boston, the second time. Just before the conference, in fact, Todd had organized it so that his band was performing in a rock club, and he invited me to come on stage and play the trombone in one of his pieces. I expanded my SIGGRAPH repertoire a bit that year. That was the year that John Lasseter organized a session called "Bloopers, Outtakes, and Horror Stories of SIGGRAPH Films," and so I was able to show a few of those in that. And that year I showed just the opening sequence to the *Project: Mathematics!* I had actually something in the film show practically every year up to this point, and this one was only like five seconds long, but at least I got it in.



1990, in Dallas. This is the first time we had the SIGGRAPH Bowl,

which I was sort of the mascot of. Basically, I waved applause signs when we wanted people to applaud. And I showed to SIGGRAPH the first results of the *Project: Mathematics!* thing. Maxine Brown did a nice thing for me. She made up a hundred ribbons for me to hand out with the notation "I hugged Jim Blinn." [*laughter*] But only people who earned them got the ribbons.



Las Vegas in 91. Las Vegas was, actually, a little bit big for SIGGRAPH. It kind of got lost in there and diffused through the entire town, so I don't have a whole lot of recollections from that one. [*laughter*] That was the year, however, Loren Carpenter first did his red/green paddle thing, and that was a big hit. And, I understand, it was dicey getting it to work in time, but it did, and everybody loved it, and we have another one of those this year.



1992, Chicago two. This is the year that I had the maximum number of ribbons on my badge. The organizers of 1992 contributed to that because they all made their own ribbons saying various odd things. And, also, being sort of a student of presentations and so forth, I realized that, when you're talking like this and they put you on the monitor, if you wear dark clothing, you disappear into the background. So, instead of using the green sweater that I had been doing, I went out and had my mother knit me another [light green sweater](#) so I would show up better against the backgrounds. [*laughter, applause*]



1993, second year in Anaheim. This is the year they had the virtual sex panel which, uh, everybody had a good time with. 1993 was also the year that had, what I think is probably the most amazing SIGGRAPH

party that's ever happened, which was the party at the Richard Nixon Museum. I have this image of watching Timothy Leary ranting and raving on stage at the Nixon Museum, twenty feet from the grave of Pat Nixon, and it's... just really cosmic, I don't know.

This is also the year that I got married the day after SIGGRAPH was finished, and somebody produced ribbons saying "Bride" and "Groom" for Amanda and myself at the conference.



1994, in Orlando. One of the main recollections I have of that is another very amazing party, put on by Bruce and **Carmi** where they rented out the entire MGM studios for us to go and cavort around at one o'clock in the morning. I'm still impressed by that.



Los Angeles, 1995. I don't know what this picture is. This is showing SIGGRAPH getting into pain or something. This was the year I was on the Electronic Theater Jury myself with Alvy and David Em, and got the chance to pick the films that were shown. This was the year that the movie *Toy Story* came out, which was one of the major achievements in computer graphics up to that point. And this is the session where there was a panel session called "Ask Dr. SIGGRAPH" where we had a bunch of us up on stage attempting to come up with funny answers to questions that people would pose. Jim Kajiya and I were kidding on stage, and Jim said, "hey, Jim, why don't you come up and work for me at Microsoft." And I said, "well, I don't know, give me your number, I'll give you a call." I had no idea at the time that, a month and half later, I'd be working at Microsoft.



Okay, 1996. New Orleans. This is the time that Al Barr and I actually got into some parties, without tickets, by crashing in and attempting to look important - and it worked! [laughter] And I showed one of the papers excerpts at the film show that year.



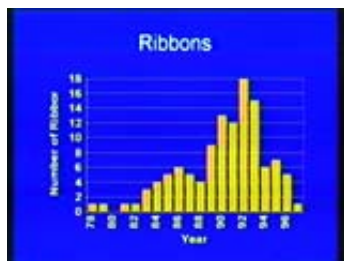
1997, Los Angeles. I don't remember a whole lot about the conference because I spent so much time talking to friends in the area, and I didn't see much of the conference.



And, 1998, I gave the Key Note address. [laughter] There was a big history thing and what not.



And, 1999. [laughter] Well, I'm getting a bit ahead of myself here.



So, let's go over and kind of see what this all means. First of all, we have to examine the ribbon count. [laughter / applause] And you can see the physical manifestation of that in the SIGGRAPH Time Tunnel out there.

This is kind of an interesting thing, if I can work this in the slide sorter viewer in PowerPoint.

[shows composite of his entire slide presentation]

Gives kind of a nice little everything-all-at-once of the entire 25 years of SIGGRAPH Conference Proceeding cover history here.



I want to kind of go through and give some observations on how things have changed over the last 25 years. You can kind of see the difference, this is the first Conference Proceedings. *[holds up 1974]* Actually, this is actually just the abstracts. At the time, they didn't know whether this was going to be interesting enough for people to see the whole papers printed. And, I understand, actually, the papers were printed in some other journal elsewhere, which I never got a copy of, unfortunately. So, you can see things have gotten thicker. *[holds up 1998 versus 1974]* And they use a lot heavier paper. This thing weighs like 35 pounds. And they use a lot smaller type than they used to, so you can fit more into the proceedings. *[flips through 1998]*

Well, let's go through a few kind of changes between then and now.



Click for text

Then. In 1974, that was before the invention of pixels, almost. Displays were calligraphic, which means you steer the beam around on the screen to trace out lines. Now, they're rastergraphic.



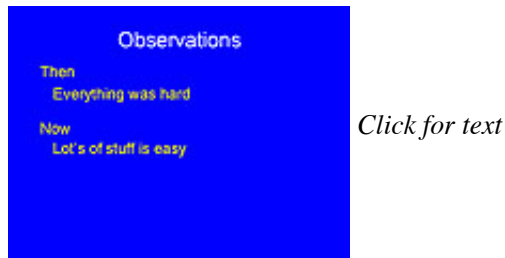
Click for text

Then. Most of the graphics was done on mainframes or minis, which cost in the \$100,000s range. Then things worked down into workstation end of things, which is in the \$10,000s range. And now, you can do pretty major things with just a desktop machine in \$1,000-2,000 range. The change in cost of these things of a factor of a couple hundred or so,

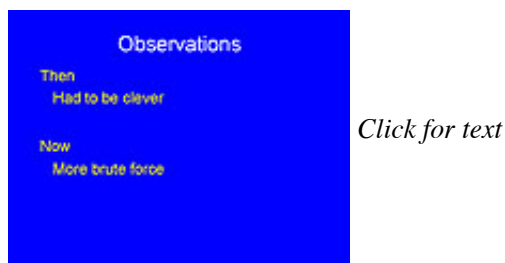
over the years, has obviously had implications in terms of accessibility. Now, a lot more people can get at it than used to.



So, at the time, there weren't very many people around, because it took, like, institutions to fund these displays that now we have on our desktop. Now everybody is getting into the act.



Back then, everything was hard. It was hard just getting a line on the screen. It was real hard getting the line recorded onto film or on video. Now days, everything is easy. That might come as a surprise to all these people who are trolling away, fudging pixels in these high-res special effects movies. But, a lot of the stuff you can get now is off-the-shelf and you don't have to reinvent *everything* in order to get any sort of thing on the screen at all.



Then, since computers were slow, the emphasis was on clever algorithms that made large scale decisions to skip things or to make simple pictures. Nowadays, computers are fast enough so that you can get away a lot more with brute force type algorithms.



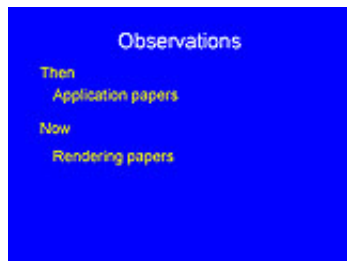
Then, most of the applications of computer graphics were in the

computer-aided design and data analysis. And that was what computer systems were sort of designed or aimed at. Nowadays, a lot of the applications are in special effects, image processing, graphic arts, and so forth. And so programs are designed to do that.



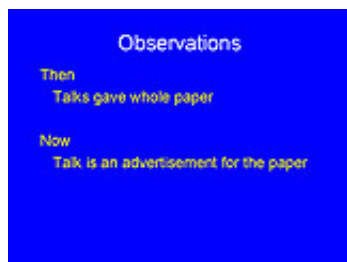
Click for text

Then, most of the papers were concerned with the feasibility of something. Is it at all possible to make a cube rotate on the screen? Is it all possible to make something that looks vaguely like a human face? But, just getting it to work at all was the major accomplishment. And having it be incredibly good looking was a little beyond what we could do at the time. Nowadays, people are focusing on more refinement and practicality of things, and our standards are a lot higher. Just putting a face on the screen is not interesting unless you can make it look like a really good face. Which is what's going on nowadays.



Click for text

Also, in going through the old conference proceedings, I noticed an interesting pattern as well. Back then, most of the papers were application papers. There were things like, you know, computer applications in cartography and computer applications in architecture and civil engineering, biomedical applications, data analysis, and social sciences and so forth. Nowadays, most of the papers are on rendering, which is like how to make pictures of faces, or cloth, or plants, or various geometric things in multi-resolution surfaces, and what have you. So, the applications don't show up so much, which is maybe something we ought to think about a bit. Although, I'm sure that SIGGRAPH people are interested in application papers, but probably fewer people are sending them in.



Click for text

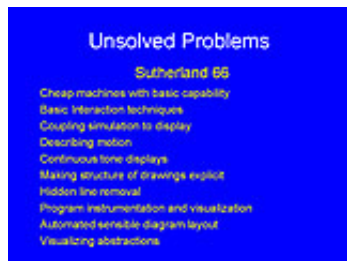
Back then, the papers were simple enough so that the talk you gave at the conference, you could give the whole paper and tell what you did. Nowadays, the talk is more like a movie trailer version of the paper. There's no way that you can get everything in a paper into a twenty-minute talk because the things are a lot more complex and a lot more involved. Nowadays, a SIGGRAPH paper, in fact, is a yearlong effort, a yearlong multimedia project. First you have to write the paper. Then you have to do the video that goes with the paper. Then you have to do the finished version of the paper. Then you have to make the slide show version of it. Then you have to perform it in public. And then you have to put it on your website. So, you know, life is a little more complicated now than it was getting something published in SIGGRAPH.

Well, what else am I going to say? I seem to have a few more minutes here.



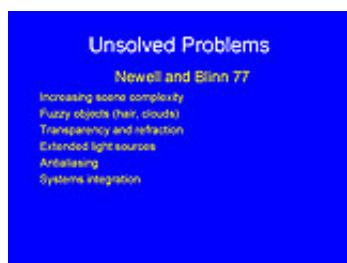
Click for text

One other kind of venerable thing that people have done over the years is the concept of unsolved problems. That was first done in 1966 by Ivan Sutherland, and, eleven years later, Martin Newell and I wrote up a list of unsolved problems. 1987, Paul Heckbert did one. 1991, there was an entire SIGGRAPH panel on it. And so, 1998, I'll do my part again.



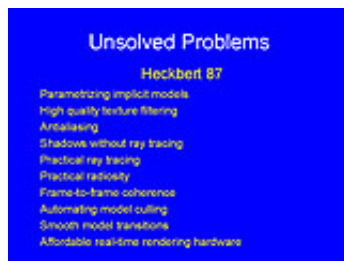
Click for text

Let's just kind of review the old ones briefly. In 1966, these were what Ivan thought were the unsolved problems. Basically, cheap machines - we've got that now. Basic interaction techniques are the things that we use a lot with mice and so forth. Coupling simulations to display - there are a lot of things doing that. Most of these things have been addressed quite a bit, although not everything is perfectly solved.



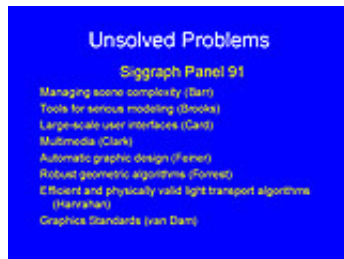
Click for text

The things that Martin and I talked about in 1977 were mostly rendering problems rather than interaction problems. Just making more complex scenes, fuzzy objects - this is back, of course, when we could make cubes, and that was about it. Teapots were the state of the art. Fuzzy objects - hmm - I guess we got that covered now. Transparency, refraction, and so forth.



Click for text

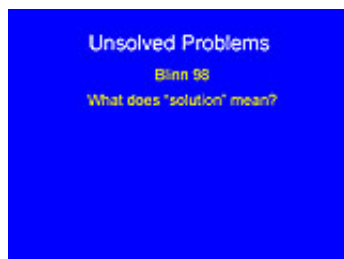
Paul Heckbert had another set of rendering things that he published in 1987. What's interesting about this list is he's again getting into the area not so much of how to do something at all but how to do something practically and fast, and, you know, in a production sort of environment rather than just showing feasibility only.



Click for text

And then the SIGGRAPH panel, by 1991, the thing had gotten complex enough that it took an entire panel to come up with some unsolved problems, and these were the ones that individuals in the panel found important.

So, what am I going to do in 1998?



Click for text

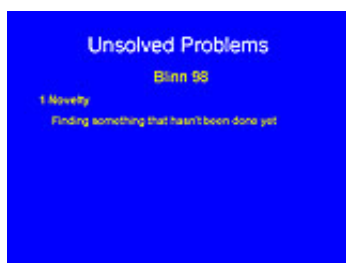
Well, in thinking about this, first of all, the question comes up "what does *solution* mean?" Is a solution to a problem mean being able to do it at all, in 8 or 9 hours long rendering and so forth, or does the solution mean doing it in a practical sense and doing it quickly enough so that people can use it other than as an existence proof. There are some problems in either of those categories - some things that nothing has been touched and others that people need to make practical in order to make usable.



Click for text

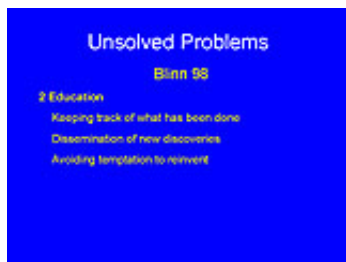
Before getting into those, I want to talk a little bit about some problems which nobody put in their list and that have been addressed, but I won't say have been completely solved. For example, non-photorealistic rendering was something that nobody really predicted a need for in any of these things. Likewise, image based rendering. And, something which I call the "tyranny of the framebuffer," which is the recognition of the fact that having just one piece of memory where the entire image is stored is, actually, rather inconvenient in today's systems which have multiple windows moving around and you have to re-render things that get uncovered and so forth. What's really a lot nicer is if you have, kind of, almost a return to the mechanism of the calligraphic systems where you have a list of things that you are going to draw. All you, as the programmer, need to do is to manipulate that list and then the hardware dynamically composites all the items in that list, together, into one image on the screen. This is one of the components of the Talisman system that Jim Kajiya and his colleagues put together a couple of years ago. Talisman includes a lot of things, but I think that's one of the most interesting aspects of it, and I think it will effect how displays are built in the future.

But, these are problems that nobody had in their list and, maybe, it means that nobody thought they were worth solving, but, any event, let's march forward here. What are my ten unsolved problems in computer graphics?



Click for text

Problem number one - finding out something that hasn't been done yet. [laughter] There's thousands of people out there, all beavering away, figuring out how to draw this, how to draw that, how to interact with that and so forth, and it's really becoming difficult to figure out something that somebody hasn't already invented somewhere else.



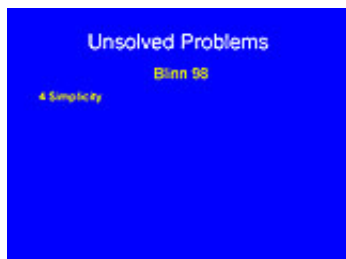
Click for text

Problem number two - this is kind of related to the first one: Finding out if somebody has done it yet or not. That is, keeping track of all the things that have been done. It used to be that SIGGRAPH was the only place that would publish computer graphics papers, and so all you had to do was read the SIGGRAPH Conference Proceedings and you knew you were up to date. But nowadays there's lots of other journals, and it takes more and more effort to make sure that you know what's happening. On the other side of the fence, disseminating the discoveries that you make yourself is also a problem, figuring out the best way to expose as many people as possible, so they don't go through and be tempted to reinvent it, although it's sometimes a lot more fun to reinvent things.



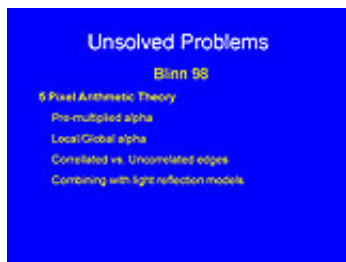
Click for text

Number three - systems integration. That is putting all the techniques into one system. We've got a lot of people doing cloth. We've got a lot of people doing faces. We've got a lot of people doing dance motions and interaction and speech and what have you. Putting that all into one system is still something that we've got a lot of inroads into, but something that's always a tricky business.



Click for text

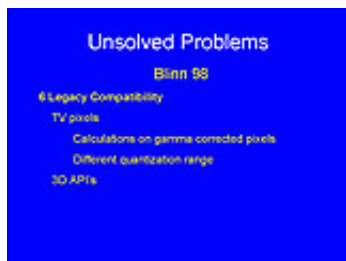
[4] Simplicity. Making this so that's not so complicated that you can't figure out how to use it. Now, simplicity is a tricky concept. I'm not sure, in fact, that simplicity is possible in anything that is complex enough to do something useful. If you look at systems like the human brain or the telephone system or any of these things that have been built up over the years, they have layers and layers of different components in them, and they still work together pretty well. And so, simple systems might not be achievable, but they're still a goal that you should try to make it as simple as possible.



Click for text

Number five - pixel arithmetic theory. This is something that, actually, I've been spending a lot of time playing around with recently, and that has to do with the idea of "what is a pixel?" and "what sort of arithmetic do you do on it?" About 1984, I guess it was, Porter and Duff came up with their paper on alpha blending and the concept of alpha pre-multiplied into the pixels, and still a lot of people doing computer systems haven't gotten that concept. Pre-multiplied alpha is a good thing, virtually always, but the fact that there are systems that don't use it indicate that we maybe don't understand the problem completely, that there's more going on than just that.

There is such a thing as an alpha that's stored in each pixel that indicates the shape of the entire object and there is a global alpha that is applied to the entire picture to make it, like, fade-in and fade-out, things like that. Alpha assumes that the edges of the object are completely uncorrelated - maybe there should be some additions to correct for correlated edges instead of uncorrelated edges. And when you start dealing with alpha blending of things stacking on top of each other, each of them are partially transparent, you think of them as, maybe, colored cellophane or something like that. Then you start getting to spectral matching and how do we combine the alpha concept with the idea of light going through transparent, colored glass. And combining this with models of light reflection where, every time light of a particular color bounces off something, it gets changed to another color. All of these things indicate to me that we don't completely understand this process and all aspects of it to form a kind of "unified field theory of pixel arithmetic" and so this is something that I think is worth pursuing.

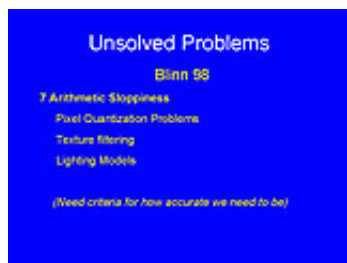


Click for text

[6] Kind of on a related idea, there's compatibility of older systems. Two, in particular, I'm keeping in mind. One is TV pixels in the upcoming, uh, let me call it convergence of TV and computers. We are going to start running into this problem more and more, and that is that TV pixels are typically gamma-corrected and computer graphics pixels are typically not. And doing calculations on these things, a lot of people just kind of sweep that difference under the rug and do the arithmetic on gamma-corrected pixels as though they were not, giving the wrong answer. Not

very wrong, but sort of wrong. Also, TV pixels have a different range of quantization - instead of going from 0 to 255, they go from 16 to 235, they have a narrower range. Making sure that we can do arithmetic on TV images the same way we can do arithmetic on computer graphics images requires all these conversions back and forth which is either slow or, if you don't do the conversion correctly, it's inaccurate, and you've got this balancing act that you're doing. How do we do that?

The other legacy compatibility issue is 3D APIs. There are a bunch of them around. Some of them are very widespread and popular, but they were invented many years ago, and we've thought of new ways of doing things since then. And how do we figure out how we can put new capabilities into the 3D APIs that we're doing in the future while not completely leaving all the old ones in the dust. Legacy compatibility is, for all these issues, something that we're going to have to deal with more and more as we have more and more legacy stuff.



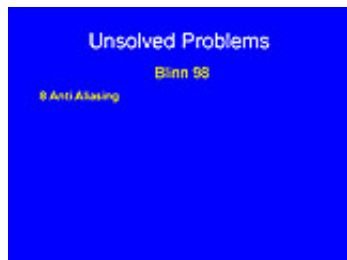
Click for text

Number seven - arithmetic sloppiness. There is an interesting phenomenon that I'm discovering as I learn what the recent developments in real-time 3D hardware are going on, as well as real-time software rendering systems. And that is, in the old days, when we did computer images, we knew it was going to take 45 minutes, so we did the arithmetic right. Nowadays, these things are almost real-time, but if you take this little shortcut or this little approximation, you can make it real-time, but it's not as accurate. And, so, there's this jump of quality in order to get to real-time, since computers aren't quite fast enough to do it correctly in real-time. So a lot of the systems make pretty astonishing, crude approximations to what the actual calculation ought to be for blending pixels together or for doing the geometric arithmetic and so forth. I find this kind of irritating that people are making these things, but they do it for a reason - because it makes it faster, and the trade off between speed and quality is something that everybody has a different position where their application fits.

Problems with dealing with quantized pixels, when you have either eight or even as few as five bits per pixel, doing the arithmetic properly is important, but maybe not that important. Texture filtering: a lot of that is being shortcut a lot in real-time systems. Lighting models... there's something that always bothered me about lighting models. Bui Tuong Phong is a great guy and he did wonderful work and so forth. But he introduced this concept of the cosine power. That is, the light reflection for specular reflection was a function of how far your viewing ray was away from a reflected ray. And in order to sculpt this function into

roughly the shape he wanted, he just took the cosine of that angle and raised to some power. You use a larger power, you get sharper highlights - a little smaller power, you get fuzzier highlights. The thing is, this has no physical basis whatsoever, but all graphics systems today, now, do highlights this way, and the idea of the cosine power, what's the cosine power of your surface? Surfaces don't have cosine power. This was a crude approximation at the time. There are better ways of calculating things like this. I'd like to see cosine power retired and better approximations being done.

Anyway, what this ultimately boils down to is the fact that we need better criteria about how accurate we need to be. A lot of these sloppy calculations are done, and, you know, purists will say, "ick, that's awful." But somebody who does it says, "well, look at the picture - it doesn't look all that much worse than doing it right." Then you look at it and you say, "well, they do look pretty close." So what we need is better criteria for how accurate we need to be to be able to properly place ourselves on the performance and quality curve here.

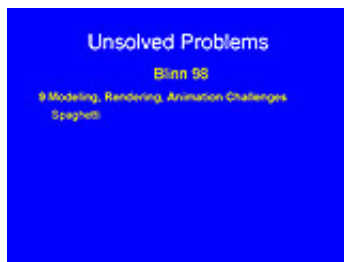


Click for text

[8] Antialiasing. Antialiasing of textures is still, uh, we have various approximations to that and we can do better. In particular, one of the things that we want to do is to be able to make text that remains readable as we take a frame of text and flying it around in three dimensions with perspective and so forth. Being able to do the filtering properly to make it readable. Antialiasing is a problem that will always be with us. I might actually state that, well, when you're talking about problems and predicting things, sometimes it's maybe better to say what never will be done, because that guarantees that somebody is going to come along and do it. For example, at the turn of the century there was somebody in the patent office, I guess, who said all possible existing inventions had already been invented - there's nothing more to invent. And, of course, they're quoted all the time. There's a quotation from Bill Gates that said 640KB of memory is good enough for anybody. He's right, actually. [laughter]

So I think may I can become more famous by saying what can't be done, because a hundred years from now, people will say, "Jim Blinn said this couldn't be done, and, of course, we've done it." So the things I *want* to be done, I'll say *can't* be done.

"Nobody will ever figure out how to do antialiasing." [laughter]



Click for text

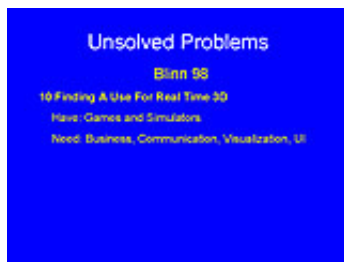
[9] No list of unsolved problems is complete without some challenge to the community for modeling, and rendering, and animation of something. It used to be just "how to do computer graphics on something," but we need to say:

- modeling is figuring out the shape of it
- rendering is how to make a picture of it
- animation is figuring out how it moves with time

So here is my challenge to the computer graphics community. [*puts up "spaghetti"*] Now, people have been doing cloth for quite a while, which is basically two dimensional shape and they're figuring out to make it crumple as it falls and the intersection and so forth. Spaghetti must be a lot easier problem because it's just one dimensional instead of two-dimensional. So, I want you to be able to figure out to drop a piece of spaghetti onto the plate and how it squiggles up and model the sauce on there for the frictional coefficients and so forth. [*laughter*] This probably has applications beyond spaghetti. One can imagine, you know, dropping ropes into piles on the floor, or string, what have you. It could even lead to models of protein folding and so forth.

So that's my challenge to the computer graphics community.

And the final unsolved problem...



Click for text

[*laughter / applause*]

[10] Finding a use for it. This is actually a real problem in the business side of things because so far we have games and simulations that are driving the hardware business on the low-end of things, but that's not enough users to justify the cost of making these things. Even though the graphics chips for doing real-time 3D are getting down into the twenty-dollar range, the companies that are making them are going out of business. So we need figure out some way that business people can get some benefit from this, communications can get some benefit from it, visualization and so forth for real-time 3D that will make lots more

people want to buy these things.

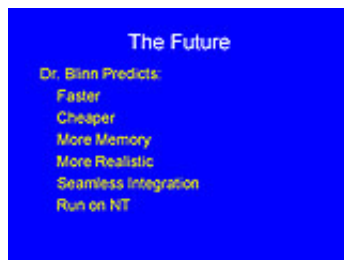
Consider user interfaces. One of the things that I've seen that occurs to me that I don't like is the way system state is represented visually on the screen. Settings you have in your program or settings you have configuring your computer, you click around, there is the window, the window points to the other window, and there's this list and menu so forth. It seems like a better way of doing it would be some sort of three dimensional shape, kind of like a car engine, where you can see the components in there, and you can pick this one, open it up, and see the thing inside there. Some sort of 3D user interface to represent system state or represent the program settings I think would be a great thing and would make it a lot easier to be able to find the way to change a particular setting you want. So this is perhaps a fertile ground for ways of milking benefit from the 3D thing.

So, here's Dr. Blinn's ten unsolved problems as of 1998.



Click for text

And so, well, none of these are going to be solved, you know.



Click for text

Now, no keynote address is complete without some discussions of the future. So, Dr. Blinn predicts...

In the future, computer graphics will be... faster. *[laughter / applause]*

In the future, computer graphics will be... cheaper.

In the future, computer graphics will... take more memory.

... it will be more realistic,...

... and, it will be seamlessly integrated with whatever it wants to integrate with,...

... and, of course, it will run on NT. *[laughter / applause]*

Now, there are those of you who might think that these predictions are

not daring enough. So, let me be a little more daring.



Click for text

I think that in the future... this is something that people have asked me for many years, and I used to think it was really dumb - "When are synthetic actors going to be indistinguishable from real actors?"

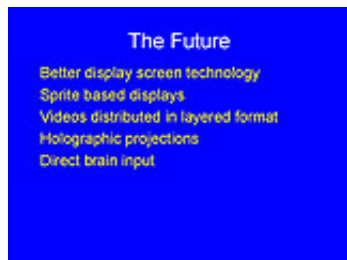
Actually, I think we're getting there. I think we're not quite there yet, but I would say by the year 2000, we'll be there...

... in April...

... 23rd...

... at 3:00pm. [*laughter / applause*]

What other things are going to happen in the future?



Click for text

Better display screen technology. They've been promising us flat panel wall screen displays since 1956 or something like that. But, you know, they're *never* going to do that. We're *never* going to have cheap flat panel wall screen displays.

Sprite based displays. This is the Talisman concept of having a list of images that are dynamically composited, so that the programmer just deals with the manipulation of a list and the hardware deals with compositing it together into one screen. Once you have that, it's possible to distribute videos directly in this format. This is kind of, part of the concept of MPEG-4. Rather than distribute video as final frames, you get much better compression when you distribute the layers individually, compress them individually.

We're going to have holographic projectors... and direct brain input... and, well, you might get the sense that I find it hard to take future predictions all that seriously some how, but [continuing on] ...



Click for text

We're going to replace the Hollywood backlots with complete synthetic things. Theme parks are going to be completely virtual.

2003, we won't need to come to SIGGRAPH anymore. [*laughter*]

And, pretty soon, anything that's not on the web won't exist. I do research on the web now, and, you know, if I can't find it on the web, I say, well, I guess it's not around, and then I realize, oh, there are actually libraries... [*laughter*] ... but... a few years, it won't be anymore.



Click for text

So, 25 years. What does this all mean?

We've created a new industry. We've taken something that was an academic curiosity back in 1974 and we've turned it into something that's so pervasive and so common that it shows up in practically every aspect of imaging. From film and movies to print medium... in fact it's sometimes cheaper to use computer graphics animation for things than live action. We saw a commercial on TV, once, where there's a moving van moving down the street and we realized - that's a computer simulation. I guess it was just cheaper to do it that way than to rent a real moving van.

We've also changed, I think, how technical conferences are run. SIGGRAPH has brought in the artistic community. We have art shows as well as film shows, and we have dance performances and what not. This is very unusual for a technical organization. The American Society of Physics probably doesn't do this. [*laughter*] But, I think we've changed the perception of the technical community as to what sorts of interesting things can go on at conferences, and I think that SIGGRAPH has been instrumental in making life more fun in the technical business.

We basically changed the world. We've made manipulating pictures, which is one of the most powerful communication medium we've known, easy and accessible and available to everyone.

And, as Jim Kajiya says, we now have new medium. Back in my

generation we grew up not being able to imagine what the world would be like without television. Nowadays, we have a generation of kids growing up not being able to imagine a world without computer graphics.



And, sort of to underscore that, I want to show a little project that I've been working on... on video. This is a video of my little boy watching his favorite movie. [*cues video of his son watching Toy Story*] It's a good thing it's such a good movie, let me tell you, because we see it two or three times a day. [*laughter*] And what I'm looking forward to is, in maybe three or four years, when he's old enough to understand the concept, I'll be able to say to him, "see all that little bumpy stuff on Mr. Potato's head? Your daddy figured out how to do that."

Thank you.

[*laughter / applause*]

Volunteer transcription by [John M. Fujii](#), SIGGRAPH 98
Copyright © 1998, ACM SIGGRAPH 98

How happy is the little stone
That rambles in the road alone,
And doesn't care about careers,
And exigencies never fears;
Whose coat of elemental brown
A passing universe put on;
And independent as the sun,
Associates or glows alone,
Fulfilling absolute decree
In casual simplicity.

Emily Dickinson, 1881



Dark green sweater



Light green sweater

[● main](#)

[● comments & questions](#)