HARDWARE STATEGIES FOR SCIENTIFIC VISUALIZATION

Chair: Robert Haber, University of Illinois at Urbana-Champaign

Panelists:

Tom Jermoluk, Silicon Graphics, Inc.
Thomas A. DeFanti, University of Illinois at Chicago
Lou Doctor, Raster Technologies, Inc.
Frank Moss, Stellar Computer, Inc.

HARDWARE STRATEGIES FOR SCIENTIFIC VISUALIZATION

ROBERT HABER: Welcome to the panel, "Hardware Strategies for Scientific Visualization." I'm Bob Haber from the National Center for Supercomputing Applications. And before we get started, I'll make one procedural announcement. The conference organizers would like to keep the stage area clear after the session is over, so they have arranged a breakout room. If you have questions for the speakers, you can meet them in Room 303. That will allow time for the quick setup they need for the next panel that follows 15 minutes after ours.

In fact, this panel and the following panel are companion sessions. They both relate to scientific visualization. This one focuses on hardware issues; and the next one will be complementary, focusing on software issues.

I'm going to begin by introducing the whole panel at this time. Our first speaker will be Frank Moss. He is the founder and vice president of Software Engineering at Stellar Computer. Prior to joining Stellar, Dr. Moss was vice president of Domain Engineering at Apollo Computer, Incorporated. He was responsible for the continued development of Apollo's operating system, UNIX, local area networks, heterogeneous interconnections and distributed processing architecture. Dr. Moss received his BSE degree in aerospace and mechanical sciences from Princeton University in 1971.

Our second speaker is a change from the printed program. Jim Clark will not be able to make it today and in his place, Tom Jermoluk from Silicon Graphics will be speaking. Tom is vice president and general manager of the Advanced Systems Division of Silicon Graphics. He is responsible for the development of Silicon Graphics' high-end workstations, including graphics and CPU hardware, as well as operating system and graphics software.

Previously, he was with Hewlett-Packard as head of hardware and software development for one of the new RISC computers of the HP precision architecture. He received a BSEE degree from Virginia Tech in 1978 and an MS in '79 from Virginia Tech.

The third speaker will be Lou Doctor. He is president and co- founder of Raster Technologies. He founded Raster Tech in 1981 with Jay Torborg, who is now vice president of research and technology. Prior to founding Raster, Lou was a staff member at the Center for Interactive Computer Graphics at RPI; where he was involved in research in advanced 3-D computer graphics algorithms for computer-aided design.

Then, finally, someone who is probably familiar to most of you, is Tom DeFanti Tom is an internationally recognized expert in computer graphics, having pioneered the development of an interactive graphics programing language for his dissertation at Ohio State. He came to the University of Illinois and founded the Circle Graphics Habitat in Chicago.

His early efforts have evolved over time and, today, the third generation of his programing language, RT-1, is the basis for all the interactive installations and the display of the exhibit, The Interactive Image, which is here at SIGGRAPH in the art show. He also co-founded the Electronic Visualization Lab, a joint effort of the College of Engineering and the School of Art and Design at the U of I. I think he also should be credited for turning

SIGGRAPH from a fairly conventional show into a complete zoo during his time in charge; and we all thank him for that. I think.

Before having the panelists start, I'm going to introduce three theme questions which I warned the speakers about prior to the panel, and give up a little bit of my neutrality by responding somewhat to each of them. That might give each of the speakers something else to respond to that they weren't expecting.

So the first theme issue is, "What will the overall hardware environment be for scientific visualization in three years?" I have chosen that time as a period that will not be so far away that we'll just be doing Buck Rogers evaluations, but also as a period that is beyond the current product line so our vendors on the panel can't get too commercial.

I think one of the issues here is, "Which classes of machine will be in the mix?" There's a number of different machine types that are now being hyped, I'd say a little bit, by the vendors, and maybe during this panel we can start to sort them out. Also, other issues we'll be looking at are mass storage, performance, networking and how will functionality be distributed among the different types of machines.

On this issue, one of the questions that we've been facing at the University of Illinois Supercomputer Center is, "What is the future of the PC?" And I think to answer that question, "What is a PC?", is the first question you have to ask. You can answer that question in a number of different ways; and depending on how you answer it, you'll either say that it will not be part of the mix in a few years from now or it will be part of the mix in visualization.

If you define it in terms of cost, as say, under $10,000 and something that's owned by a single individual, I say, yes, that will certainly be there. If you say it's distinguished by the type of operating system and software environment, I think the answer is no. I'm sure people will disagree with this, but I would at least prefer not to see it there; because having to deal with multiple operating systems is a problem.

The PC architecture I think also will not be very much distinguished from a workstation by that time. So other than cost and ownership, I predict that PCs as a class of machine are not going to be distinguished from workstations.

Super workstations is a type of machine that's been introduced for the firtst time this year, and I know the panelists are going to get into that. So I'm not going to make any further comments.

One of the things that you want to be looking at in workstation architecture - since I think there are some changes coming in this year - are bus designs. We're getting away from VME. Components will all be going into parallel, whether CPU, graphics engines or frame buffers. And the frame buffers, of course, will have quite a bit more memory associated with them. That, I don't think, is too controversial.

I think we'll be moving away from the sort of isolated environment around supercomputers, middle-tier machines and workstations and PCs, where they're fairly separate in function, to a more integrated approach like you see on the slide. You're going to have very high bandwidth networks to make that feasible. We'll be installing a 100 megabyte per second network at Illinois this year. There will be segregation of the traffic between layers of the newtork.

You'll also start to see integrated graphics continue to creep up from the PC and workstation level into the middle tier machines, and eventually, I'm going to see if I can bang on

Cray & Fujitsu and get them to integrate graphics engines and frame buffers directly into their machines. Really, the need for integrated graphics is greater as you go up than as you go down. It's just history that's responsible for what we have at this point.

Mass storage, is one thing that's currently the most out of balance in our computing environments. For example, if you look at the real prices of the super workstations when you put on a reasonable amount of disk, this point becomes clear. Right now we're talking about gigabytes of mass store as what is needed for local workstations in supercomputing environments, and terabytes on central machines now - and workstations say in a few years from now. The cost is far above and the transfer rates are way below what they need to be.

The second issue that I'd like the panelists to address is, "What will be the balance between hardware and software implementations of visualization functions in the next three years?" In the graphics world, we've seen the introduction of a lot of hardware accelerators to execute the compute intensive part of the graphics rendering algorithms in real time or in at least balanced time with respect to the rest of the process.

There is a tradeoff there. If you put something into hardware, you usually give up some flexibility. So I think we'll see a continuing debate as time goes on between which functions are put onto the general purpose CPU and which ones are going to be implemented in hardware. I'd like to see what the panelists think about that issue, where that's going. Also, identify where the key bottlenecks are going to be.

Part of the response to this a look at what role computer graphics plays in a much braoder framework. These two slides show one diagram of the scientific visualization process, where traditional computer graphics is really just the last step, the rendering step. That's down here.

There's quite a bit of compute intensive work that goes on in data enrichment and mapping your data into some kind of visual representation. Those operations are often many times more compute-intensive than the original Cray simulation, and also more compute-intensive than the rendering operation. So it seems that some of these steps might require some hardware acceleration in the future as well. To get there, we'll need to have a well- defined structure for this visualization process.

And as we go forward toward having interactive steering of scientific simulation - so you're not just interacting with your graphics, but really interacting with the model, the physical model - then this much bigger process is going to have to be addressed. It won't do us any good to have millions of polygons per second rendering rates if we can't do our simulation time steps in less than 10 minutes.

I think we're going to have to look at the whole scientific simulation process to see where we're going to need hardware accelerators to eliminate bottlenecks. The trend is going to be toward a much broader view of scientific visualization than we've had up to now.

These slides indicate some things to keep in mind with regard to what I was just saying. There is one time scale that normalizes everything else in this process. If we're talking about real-time animation from a simulation, the animation time per frame is essentially fixed, somewhere between one fifteenth and one thirtieth of a second. We should compare the time it takes to do simulation updtaes or to transmit data across a network or a bus for rendering operations to animation time; and make sure that all of these are in balance.

The third issue is, "What are the market projections for scientific visualization hardware systems over the next three years?" Visualization is a fairly new field, at least in terms of its publicity and the amount of interest that it's been getting in the last couple of years. The question for the vendors is, "How much market is really out there and can it support the amount of development of new technology that's required?" What types of computer systems will be the main elements of the mix? Should they be putting their efforts into super workstations or more modest systems - or looking at departmental machines? How will that pan out?

Finally, I'll sum up with a discussion of the NCSA Rivers project. This is an R&D effort I'm heading at the University of Illinois which is involved with both hardware and software development. Here are some of the things we're doing in the hardware end. We're looking at very high-performance graphics systems, but also very high-performance network systems to tie them together. And we are very interested in hardware-related software standards for workstations that support things like PHIGS+. We see standards as very important to ensure that there will be healthy visualization market. Standards will allow the very large costs involved in developing visualization software tools to be amortized among a larger community.

So we are looking at systems that support X Windows and NeWS and PHIGS+, well-integrated with the hardware. If the hardware vendors don't support those standards with their high-performance engines, I think we're going to do the whole country a tremendous disservice.

All this brings us to the fun part. Finally, after waiting for 12 or 15 years, we are seeing raster graphics move into the real- time interactive environment that we've had since the mid-seventies in vector graphics. This allows the scientist to move around in the data, to really be at home with it. In the case of volumetric data sets, I don't think there is any way you can really understand the whole volume of data from a representation in a single frame. You need to be able to feel around in the volume in some sense. And the real time capabilities we now see are key to doing that.

This is some crack propagation work that I did. It wasn't until very recently that we could afford to fly around it. There were views I just never would get to see because of the time involved in generating them.

So if we could roll the videotape, there's a brief piece here that gives you a flavor of what that's like. This was actually recorded last Sunday on the exhibition floor here. This was taped in real time - you'll see the interactive devices. I should credit Polly Baker, who did the NeWS programing, Ray Idaszak, who did a lot of the PHIGS+ programing, and Steve Chall, who also did a lot of work on these systems.

Here you see animation controls for stop motion, forward and reverse, stepping through step by step. To give you some reference, when we were doing this a few years ago, it took about three minutes per frame. So the speedup here is on the order of two or three thousand times what we could do a few years ago. Now you'll see the interactive controls for panning and rotating and looking at this data from any position. This gives the flavor of the kind of environment that I think scientists will be working in in the future.

Okay, we can stop there. Finally, the last point I would make is that all of this needs to go out across the network. We'd like to bring this kind of work out beyond the campus level; and this means that we need a national network that can sustain the kind of data rates we're going to be experimenting with this year on the local level. The rates are going to have to be up in the realm of 100 megabytes per second, sustained, for a single user. I think we need a national infrastructure program, like we had with highways in the 1950s, to build a very high-performance research network for the country for the nineties that will give US science and industry a competitive edge in the international market.

I'll quit there and move on to our next speaker, who is Frank Moss from Stellar Computer

FRANK MOSS: Thank you very much, Bob. Hype. You haven't seen anything yet. Well, thank you. As a software person, or at least having been typecast as that, I would like to begin by projecting that the directions and strategies in hardware for visualization will not be driven by technology. Rather, they will be driven by this new class of applications that Bob has discussed, whereby users intimately observe and interact with complex 3-D models. They dynamically change in time to reflect underlying physical and mathematical principles in response to user input.

I've heard this called by several names, visual simulation. Randy Smith of Xerox PARC has called it realization as opposed to visualization, and I think that's a neat enough term. You see many early examples of that on the SIGGRAPH floor this week.

Now realization applications, if I can call them that for short just for this talk, will require a new metric of hardware or system performance that goes well beyond the MIPS, MEGAFLOPS and MEGAPOLYS of various types and stripes that we've been talking about. The fixed ideas. Consider the general structure of these applications as shown on the left of your screen.

Now the underlying dynamic analysis will be performed by a numerically intensive computation, abbreviated, NIC. An example of this is the Navier-Stokes equation of computational fluid dynamics. The NIC generates dynamically changing values of raw data, for example, the values of pressure at points in the 3-D grid. This raw data must be converted to viewable graphical renderings by a mapping function, abbreviated as MAP on your screen. In our CFD example, the discrete points of pressure must be mapped from a 3-D grid to isosurfaces to make them usable and viewable.

Finally, the primitives are dynamically rendered as viewable images to the user. The user can interact with either the rendering, with the mapping, or with the numerically intensive compute. This is the model that we see.

By analogy to animation, you can think of the notion of a realization frame, which is composed of a single step of a numerically intensive compute, the mapping, the rendering, and the all-important data transfers between them. I'm suggesting, just for the sake of argument, that you can conceive of trying to maximize the number of realizations per second. This gives you the levels of dynamics and the interactivity that will be desired over a broad range of applications which have different mixes of computation, of mapping functions, and of graphics.

Now given a desire to maximize RPS, I'd like to suggest three hardware system strategies. Actually, they're system strategies because they involve software as well. I believe that these three strategies will emerge first in the form of a new class of computer system, or a new member of the technical computing hierarchy, which has come to be known as a graphics supercomputer. You've seen several here at SIGGRAPH this week. They'll be distinguished, as I said before, not only by MIPS, MEGAFLOPS and MEGAPOLYS, but by these three architectural characteristics which I believe will become more and more common in systems of all types as we move along.

Let me look at these briefly, one at a time. First, let's deal with the question of bottlenecks in the visualization pipeline or the realization pipeline, and the issue of bandwidth. On the left of your screen you see the function distribution as it typically appears today. The numerically intensive compute and mapping on the supercomputer connected, typically by an ethernet or eventually by higher-bandwidth networks, to a graphics workstation which typically runs, perhaps, a piece of the mapping, but primarily, the rendering of the display. The user interacts with the rendering and occasionally interacts with the function on the supercomputer.

Network bandwidth, more or less, is one megabyte per second throughput. Now over the next several years, those bandwidths will increase, at least by an order of magnitude with FDDI and by several orders of magnitude with special-purpose, high-bandwidth channel type networks like HSC and Ultra-Net. But I contend that with growth in the power of supercomputers and the power of graphics, that networks will remain a relative bottleneck in the dynamic operation of the realization pipeline.

So I believe that yet another order of magnitude increase will be required, and that I believe, will be delivered as internal bandwidth in very, very high-performance systems. Pieces of the numerically intensive computation, denoted here by NIC double prime, will be offloaded from the supercomputer to graphics supercomputer kinds of systems; and it will be tightly integrated, with a bandwidth exceeding a gigabyte per second. Over the next three years, I see those internal bandwidths increasing even more, and having access to even larger memories and larger caches which greatly increase the flexibility and the efficiency of data transfer between the various steps of the realization pipeline. I think this is a very, very important step and I think it will be reflected in MAP a generation later within networks.

Now, the second strategy or trend that I'd like to talk about is that of trying to meet a price point. These systems are expensive. The capabilities are complex. And we ask the question, how can we reduce costs and drive them down? We all want to do that. Well, I believe that an important notion is to recognize the commonality between the needs of supercomputing, what I've called NIC and MAP here, and the need for high performance, high functionality, 3-D graphics rendering. We can reduce cost by sharing these capabilities.

On the left, I've depicted by this Venn diagram the capabilities that are typically devoted in a supercomputer to a numerically intensive compute and to the mapping. And on the right, I've depicted those that are dedicated in a graphics workstation to rendering.

Now the trend will be toward more general-purpose functional units, indicated in the center by the intersection, and away from specialized special-purpose hardware whenever possible or whenever appropriate. For example, at the intersection we see vector floating point processing. Typically used for large-scale numerical computation, this can constitute the front end of a graphics rendering pipeline and perform functions like translation, clipping and lighting computations.

Large memories, which are typical to supercomputers, can be used to contain virtual pixel maps of arbitrary depth and of arbitrary size and eliminate the need for very large costly frame buffers, etc. There are a number of things that can be shared in these systems and it will give rise to the three advantages which I've listed. You can have a lever between computation and graphics and allocate your resources in your application according to its needs at the time.

You may dynamically change the allocation between NIC, MAP and rendering, as the application proceeds, or among a number of different applications. We can apply the well-known advantages of virtual memory, to graphics: the ability to have arbitrarily large images, the ability to have application access to pixels for functions like image processing - all transparent to the application programer.

And finally, by bringing some of what were previously special- purpose, lower-level hardware functions into a general-purpose programing environment, we can expedite development. First development by vendors, then development by users.

For example, the ability to move the mapping function, today performed primarily on the compute piece of the machine, into the pixel processor, or into the vector floating point processor shared with the pixel processor. There are great opportunities for advancing performance in that way, and for advanced rendering techniques, such as photo-realistic rendering and volumetric rendering.

And finally, my third trend, if you will, is the trend toward a new meaning of parallelism. Parallelism, to date, has been limited to breaking up the loops of a numerically intensive computation into smaller pieces, and running them in a fine-grain fashion on parallel processors. Well, I believe that with the advent of general-purpose multiple processing units in graphics supercomputers and other systems, that we'll be able to greatly increase the rate at which we can execute the realization pipeline.

At the top, we see a serial execution of the pipeline. I've normalized that to realizations per second of one. In the center strip we see what would happen if you allocate different functions of the pipeline to different streams of execution synchronized in a way that reduces your time to execution and increases your throughput by greater than a factor of two. Again, we are exploiting multiple functional units, not only for numerically intensive compute, but for graphics functions. At the bottom, I've illustrated how you can increase your performance by even greater factors, if you pay attention to a finer-grain level of parallelism within each of the functions (which in a sense has always been applied to numerically intensive compute). Here, the theoretical optimum of three is reached.

So I really do believe that the concept of parallelism will become a much broader, a much more general notion over the next two or three years.

Now I mentioned before that MIPS, MEGAFLOPS and MEGAPOLYS are not the entire story; that you need architectural features that allow applications to tap into those resources, to get the levels of performance needed for dynamic visualization. But for those of you who like to play the numbers, I've offered my version of a prediction that has a little bit to do with where we think we're going at Stellar, but also with where we see the technology and the applications driving things.

I've shown price versus year projected out to 1992; and you can see that in three years, I see the price of high-end systems moving down through the learning curve. I've demonstrated the numbers I have there. I've illustrated the numbers, 200 MIPS, 80 MEGAFLOPS LINPACK, and significant dynamic rendering capabilities for photo-realistic rendering. I also see the kinds of capabilities that we offer today appearing in mid-range machines of half the price, beginning at about 1990.

So, I'm told to sum up and I've got the yellow light here just in time for the last slide. And my message is that MIPS, MEGAFLOPS and MEGAPOLYS are not the bottom line. Unique architectural features will appear and, indeed, they'll have to be exploited by capabilities in software, analogous to optimizing compilers, that make it easy for scientists and engineers to exploit and tap into this great level of power and to write dynamic visualization for realization applications. So that's my talk. Thank you very much.

HABER: Thank you, Frank. The next speaker is Tom Jermoluk from Silicon Graphics.

TOM JERMOLUK: Good morning. I guess that I'm the specialized part of the talk this morning, as opposed to the general-purpose part of the talk. I'd like to start out with a little bit of a look at the market, though. You'll all be pleased to know that we do believe the market will continue to exist in 1992.

I can't have a graphics talk without a pretty picture. This is "Directions in Visualization," so I thought that I'd show you where we would like to go with visualization. Some of the market surveys that we have about the size of the three-dimensional visualization market out over the next few years show that we are on the underside of the knee of the curve. I think this technology is exploding. Various people that you see here, Dataquest and Alex Brown, etc., have estimated this market to be growing fairly exponentially over the next few years.

Where this market will lead in terms of users, though, is the more interesting question. I believe that we've come from an era of the power user, the era of the person who's on the leading edge of technology - the animation users, the visual simulation users - and we're moving much more into the era of the application users. When you can walk into a Chrysler or a Ford or a GM or a Boeing or a Lockheed, people who are actually out there doing real work with systems on every engineer's desk, that's when you're getting into the application era, and that's what we believe to be approaching today from many of the companies involved with this technology.

The more interesting one in the future will be the consumers. When and if 3-D visualization gets into the consumer market, that of course will significantly change the way that most people think about dealing and interacting with computers. I think a large part of what scares off some of the consumer market today is the lack of an interactive realistic interface for these systems.

The sorts of things which denote the users that we see today, though, is a question of interest. The power user typically has no concern but the absolute highest power that they can achieve on the system and the most functionality they can get. They rarely care about standards. They rarely care about how it looks. They don't care about how much power it takes or how much money it costs. They simply have a job that's too large to be done in any other way.

The more advanced users do care about those things that I mentioned. They're looking to do a job over a long term. They are probably people who are not writing applications, but they are modifying things for the rest of a community within their company to use; and they have a certain need for broad product lines.

They want the high end, they want the low end; so they can distribute their software. They want flexibility in the system, to be able to provide many different functions and tackle lots of different tasks in their day-to-day work. And compatibility across the line is very important, as well as compatibility with other workstations and large computer vendors.

Then the application user is primarily motivated by price. There's a lot of price elasticity in that particular marketplace. Ease of use. They don't particularly care to deal with complex graphics languages or UNIX interfaces. They want simple turnkey applications. And they're very interested in how many packages are available, much like the early PC users were. Some of these characteristics of compatibility, networking, single vendor, are things that large companies look for when they're buying across a product line.

This is sort of a visual of where we see the bulk of this market. For every workstation categorized by the top of the pyramid, say, $75,000 and up, I believe that there's about 30 workstations that will go with it for the application users. So about 10% of the market is in super workstations and by far the majority is in what we'd refer to in the future as personal 3-D workstations.

I'll talk a little bit about technology. I think that Frank had a slide much like this. The upper one shows the direction that we will go in. We have a multiple CPU architecture, a dedicated graphics subsystem and a display subsystem, and that's as contrasted to a general-purpose computer which would have multiple processors hooked to a display system.

He's right in that the link between those two, as between Crays and graphics workstations, has traditionally been a network, some kind of networking communications capability. Now the link is migrating more toward a bus, some sort of high-speed bus hooked as a memory interconnect between the processor and the display portion of the subsystem.

So that's a very interesting trend. A number of people are taking that particular thing. I don't believe that there's necessarily any pluses or minuses to either one of these. It's more a question of focus, where do you particularly see some of these workstations being used.

It's our belief that the graphics subsystem is a vital piece for the interactivity of 3-D, which is the primary market that we're interested in. Because in fact, the rates that are achieved by the specialized hardware and the graphics subsystem aren't attainable on the more general-purpose CPUs. In addition, if you're using the bandwidth on your general CPUs for the analysis operations, you're not drawing graphics. So you're either doing one or the other, as opposed to the interactive scenario where you want to be doing some analysis on the data while you're actually interacting with the graphical representation.

An example of this is a calculation of the number of MEGAFLOPS required to do a polygonal transformation and viewing. If you took 100,000 of these four-sided polygons and did the appropriate calculations required to display them - you have the transformation of the vertices and the normals, the lighting, the clipping, the projection for perspective and the viewport mapping - that's about 46.5 MEGAFLOPS.

We believe that it requires a rather specialized architecture to be able to sustain that sort of a rate. And if you're sustaining that rate -- if you are attempting to sustain that rate on your general-purpose side, then those MEGAFLOPS aren't available for the analytical calculations that you'd like to have going on.

This is the sort of program where you'd like to interact with your analysis and your graphics, mapping of the stress points on a piston carved from a cylinder block. You'd like to be able to move that around in real time, which obviously, I can't do here. But you'd like to move that around and move the cutting plane through, watching where the stress points are in that piston, doing the analysis on the host and the graphics display with the graphics subsystem.

Where is it going? Oh, I'm sorry. A little out of order. This is another representation. I'm in agreement with Frank on this point in that graphics data expands exponentially. The data in portion here is described as 100 surface representation patches, each with 16 control points representing 9.6 kilobytes of data. The data out, in actuality, the data that your eye sees, is represented by about 20,000 polygons of 100 pixels each, with over 16 million bytes of information. So the problem is, where do you do the expansion of that data?

If you do it on the general purpose side, that means that the data transfer requirements go up significantly over the bus that might also connect you with your memory or your I/O system. That takes away bandwidth from the analysis calculations once again. You'd like to have that down in a more specialized area that can handle that bus traffic totally in parallel with the rest of your processing.

So, where is it going? As I look out over the next couple of years, among the things that we see users asking for, more and more realism is very high on the list. Phong shading and other advanced lighting techniques. Texture and environment mapping is very important for the realism aspects. Anti- aliasing. There's some technology reasons why monitors aren't going to a lot higher res very quickly, and to compensate for that, many people are interested in anti-aliasing techniques. I think you'll see those a lot more prevalent over the next couple of years. And certainly, surface representations to, once again, cut down on the amount of data transfer required, if nothing else.

More and more performance, smaller polygons, larger numbers of polygons. As your requirements for realism go up, the size of your polygon goes down with respect to the lighting models and other aspects of the graphical computations.

Lower clear times. The interactivity that Bob referred to is very important. You want fast screen clear times. You want to be able to interact and move those objects on the screen without wasting a lot of your drawing time, and you want a lot lower overhead for your interaction. More hardware for window manager support. More hardware for interaction with the operating system.

And finally, more and more integration. Probably this is most important. As we move into the application-ease-of-use area, these are, I believe, the keys to the successful company. To integrate things very successfully with the window managers, to have X with a nice 3-D representation, NeWS with a good 3-D representation is very important. To have your operating system fully support and understand the requirements of the graphics subsystem, the schedulings required.

And finally, of course, to network these things together very successfully. Because, in fact, there will always be a need for several levels of these different types of computers in any significant application. Thank you.

HABER: Thank you, Tom. The next speaker will be Lou Doctor from Raster Technologies and ALLIANT.

LOU DOCTOR: And now for something completely different. What I wanted to talk about today is actually quite a twist, in contrast to what was talked about by the first panel members. So, rather than telling you that workstations are getting faster and more powerful and that you'll be able to do a lot of local computation, I'm going to try in my talk to spur a little bit of thought as to what could be - as opposed to what seems to be happening - to give you a bit of an indication as to where we're going in the longer term.

I think everybody is in agreement that today's systems are really sort of too slow. They are too slow in just about every respect. A hundred times better computation, a hundred times better graphics. If you could have any system you wanted, you would clearly take systems that are a lot more powerful now and make them available to users.

But the other thing is that they're too expensive. It's pretty obvious that not every scientist and engineer can afford a $100,000 and up workstation, and even when they get to $50,000 and up, they're still going to be too expensive. The third element is that they're also too impractical.

I'm speaking as a vendor into this market; and I find it ironic that when you decide to spend 10 times as much for a high-end workstation as you spend for your $10,000 and below

PC or Mac, the high-end workstation is almost completely useless to do anything but high-end graphics. It can't run all the utility software and all of the things that scientists and engineers do 90% of the time.

So we've looked at that problem and we came to the conclusion that the high-end workstation market is going to be a market that basically is represented by what I would call islands of computation. If a user is literally isolated, maybe in an island in Hawaii or something, maybe they need a high-end workstation because they can't talk to anyone and they can't take advantage of other people around them in the organization and the buying power of being able to put some of their dollars together.

I think that what's going to happen is that you're going to see all different kinds of workstations emerging. You've got these commercial 386 based ones that are coming out with follow-ons and those are clearly being migrated into what I'll call the commercial and office market. I think everybody would agree that they're not that interesting for scientific visualization.

Then you've got a whole family of RISC based architectures for CASE with 68000 instruction sets, or maybe RISC instruction sets. Those really don't have the graphics horsepower, and they aren't that interesting either for scientific visualization.

You've got this market that I think Tom was talking about for what I'll call the production CAD user. That's the person that sits in front of their screen all day long and does nothing but graphics. They need sub-one- second response time; and they need good local 3-D capability. That clearly is a growth market.

Companies like IBM, Computervision and Prime are shipping tens of thousands of workstations into that market every year and selling them to the automotive companies that Tom had talked about. So there is clearly a market there; and that market is clearly growing.

But now I've taken another twist; and I've defined a new class of workstation that I contend is going to emerge as the dominate way of doing scientific visualization over this three-year period that Bob asked us to talk about. That's basically using cheap workstations that have essentially a live video capability. And I'll talk more about that in a minute.

So I think that what is going to happen is that we will very much see a computing hierarchy, just like Bob had talked about and shown on the slide of his architecture. We think that companies or organizations will gang together and pool all of their dollars and make what will emerge as a strategic decision to buy a supercomputer.

I think people are realizing that the long-range implications of supercomputing are so tremendous that the most innovative companies and the companies that have that vision are buying supercomputers. So you'll see one or more clustered in a group for an entire company.

Now I've thrown out a new notion which applies when you have a work group or a department that has to use machines. I've invented this new machine type which I'll call a visual server. But a visual server is a machine that is connected to a supercomputing network at very high speed, but basically gives a department or work group a local capability for both graphics and MEGAFLOPS that is far in excess of what each user could afford to buy for themselves.

Essentially, a lot of the work that we've done, now that we've merged together with ALLIANT, is in trying to look at this marketplace for multi-user servers that have very large physical memory systems, all of the characteristics that we talked about here previously. But they have specialized processors for graphics, I/O and computation. That's the nature of a multi-user system, and that's why it varies quite a bit from what you would design for a workstation architecture.

I think that the networks are going to advance, not just in their ability to handle digital data, because digital data is only one form of the data. But I think that the networks will advance to the point where off-the-shelf video technology can be usable for delivering pictures to people sitting at multi-media work stations.

So you literally could sit in your office on a Mac three years from now, pop open a window and get pictures from a remote server or supercomputer. I see that as being the direction that the market is going for the vast majority of systems for scientific visualization, because 90% of what scientists do does not require the graphics and visualization power of a $100,000 workstation. So that's my key point of controversy.

So I think that is what we'll do. We've defined this new class of machine, a visualization server in some fashion. What are the characteristics of it? Well, number one, it interfaces a lot better to existing supercomputers and extremely high-speed networks.

I think Bob described this 100 gigabit network, or trying to move these extremely large data files around and load balance between supercomputers on a network. It's very, very clear that you've got to have that. But it also seems clear to me that those network nodes are not going to be practical to go into your Mac or workstation directly. In fact, you need impedance matching computers between the high speed networks that move data around in supercomputers and the sort of PC or low-end workstation you're going to have on your desk.

I think that this notion of what I'm calling visualization on demand as a time sharing resource is really an extension of the idea of being able to move pictures around rather freely. So it is a kind of an odd twist, but it's not so far-fetched that it doesn't make a lot of sense for a lot of applications.

In effect, what we've said is, we kind of shared this vision. We decided to take a company that had the graphics to build a super workstation, and essentially take that super workstation graphics and stick it onto a server class machine - where each processor inside of, say, an ALLIANT machine or any one of these other server machines, is more powerful than you can buy in the most expensive super workstation. And when you have a departmental resource that has, perhaps, 10 times that power, you're really giving every user on-demand access to much more performance than would have been practical otherwise.

So the idea is that basically we demonstrated this capability by hooking up a few screens that were essentially direct connected. We designed a system where the memory system of the server was the memory system that you build graphics structures in. And we've done a lot of work on the idea of how you build an architecture that can support multiple users. We've announced a system that supports from one to 16 visual channels off of your server; where every visual channel in this product has the look and feel of a workstation. So you can use things like fiber optic links to pull these screens away from the server and give them a capability to be brought into somebody's office. So it's sort of an odd twist again.

In effect, we've replaced ethernet by the memory system of the server for building graphics structures; and then we basically ship video to users, as opposed to trying to ship digital data.

So I think that what we see is that, essentially, these leading workstation vendors are going to have this multi-media capability. It doesn't take much to extrapolate what you can see now from companies such as Apple and Sun to imagine that you'll be able to have workstations in which you can pop windows. And I think you're going to have some local graphics capability also. It's pretty clear that there is going to be some level of graphics that you want to be able to generate yourself.

But for that really peak demanding visualization requirement that you only use once in a while, it's actually more practical to have those images and those image data bases rendered and managed by a multi-user server, because of the nature of the problem. The problem still seems to be getting more expensive and more powerful, again, taking us toward multi-user systems.

I think that the kinds of pictures that people will pop up in their windows are going to be essentially live video that comes off of these remote servers and also things that they may have in their office, like videotape and disk and cable TV services. So they're going to have an ability to get a lot more usefulness from their PCs, or super Macs, if you will. I think at the same time, they'll have access on demand to the most powerful visualization resources. And that's really a key to my observation about the marketplace.

So what are the characteristics of this sort of an approach? You get the benefits of what I'm calling low cost per seat, and you have all of the usefulness of the existing kind of PC to run all the spread sheet and planning and word processing software that people do most of the time, at least in the science and engineering area. We're able to take advantage of this large installed base of PCs and Macs and whatever to tap into these visual resources.

I think that window systems are going to be sophisticated enough to handle this notion of mixing together, for example, a live video capability along with digital data. I think you're very much going to be able to optimize both the supercomputer resources, which are going to remain centralized at least for the next 50 years, and the scientist resources, which are going to be decentralized, where people do most of their work in their office sitting in front of their PC or workstation.

While I'm not by any means forecasting gloom and doom for the high-end workstation market, I think it's going to emerge as a very specialized market - only for circumstances where you can't tap into these server resources. I think that there are also going to be applications that demand a level of real-time performance at the workstation, such as the military and the defense area, that are simply going to demand local graphics accelerators. But I don't see it becoming the mainstream. I see an approach like this becoming the mainstream, and I see us doing a lot of work together with ALLIANT to make that happen.

I'm getting the boot here. So again, I think that's my talk and I'll cut it off right there. That was my last line anyway.

HABER: Thank you very much, Lou. I'm being a little bit of a dictator here to keep us on schedule. The last speaker is Tom Defanti, who will take up the "Everyman's" point of view.

TOM DEFANTI: The people you just heard from, of course, are more or less from what I call the conspicuous consumption point of view of computing. I've always -- most of you who know me, know that I kind of came out of video games. So I have a little different approach to all of this; and I've been very interested in getting this stuff out to people who are quite poor.

One way to get 3-D visualization to people is what we have on the corner of the stage over there. It's called a phscalogram; and I hope if you haven't seen one before in the art show or in one of the 3-D sessions, that you come up and look at it. It's a way to involve people who don't happen to have $100,000 workstations to see things in three dimensions in a public space.

In order to compete with these guys - Stellar and other astronomical-type projections, I had to go five years into the future. I think that in five years in the future, we're going to have PCs on our desks.

Now unfortunately, I used the large letters when I was typing this slide in last week. I used the large letters "PC" and I actually meant to use the small letters "pc"; because the large letters PC really mean IBM PC and clones. I also noticed this week that the word "artist" has become a trademark of somebody; and I don't know how to react to that in my future talks.

But I think of PCs as being personal computers and this -- I only have one slide. I was able to get all of my thoughts onto one slide. -- is essentially what I like about current PCs. I suspect that things that we call PCs in the future, regardless of whether they have a 16 bit architecture, will look like this.

Another thing is that one button I forgot to make this time would read, "I use the same hardware as Jim Blinn." I have in fact for many years used PDP-11s and VAX's; and Jim's currently using PCs with Vista boards in them. So one can do scientific visualization on very modest equipment.

Okay. The things I like about PCs are that they're networkable and you can plug them in whenever you want. You can plug them into graphics servers, which were actually described in the (NSF) visualization report. One of the nice things, though, is you can unplug them. We have the experience of unplugging our Silicon Graphics and Hewlett-Packard high-end workstations, and they complain a lot when you unplug the -- have you ever done that, unplug the ethernet? They sit there and complain at you. It's really annoying.

Configurable. I like the fact that there are two zillion people who make PCs; and when I get mad at one vendor I can get one from somewhere else. Every part of it is multi-sourced. You can mix and match it. If you don't like a particular board that's available, if you just wait a couple of months and keep reading the back of Byte magazine, you'll find one that does what you want. So that's kind of nice. You don't have to put pressure on a single hardware manufacturer to do what you want them to do.

And in science, that's really tough. If you go into any big chemistry building, for instance, you'll find out that they blow their own glass and they have a machine shop with multi-access milling machines. The reason why they have this is because you can't get people to build the kind of stuff you want. You have to do it yourself. So the configurability is extremely important.

So is the hackability. PCs, as they exist right now, are very easy to hack. There are all sorts of high school students and young college students who have these things at home; and you can actually let them just have them. They're cheap enough that they can just have them to hack at.

This is not so true with UNIX. It's relatively difficult to find people who will admit to being able to write UNIX drivers. And anytime you want to hook some whacko device or do something slightly different -- which is really what science is about -- into your computer, it's very difficult in UNIX. It's not as bad as it is to try and hook something into an IBM mainframe or something, but it's really tough.

PCs actually make pretty good interface units. Really, if you look at even the current 640K space, that's not a bad space for human interaction. I mean, just to talk to the human, 640K isn't bad. Everything else you do with boards that you plug in. I currently have a PC that has 640K address space and 14 megabytes of graphics space on two boards. So that's pretty serious. It's very much like the old LINC-8s, which were essentially the first laboratory machines, and the PDP-11s, where people just plug things in.

I realize I'm getting a little away from the general concept of visualization for computational only, where you're dealing with people who are doing all their science on large computers, to people who are mixing real-world things in there. But I see those guys as scientists, too.

Recordable. Listen to this. Everybody listen, all right? If it ain't recordable, it ain't science. It doesn't count. And that's real serious. If you want to get scientists' ears, you've got to give them a way of preserving it. You don't get away with saying, "Well gee, I did some nice science last week but I forgot to take notes." It doesn't work. There was a Journal of Irreproducible Science. We now have a Journal of Unrecordable Science.

In order to be recordable, it's got to be legit NTSC video. People call everything video; but it 's got to be legit NTSC video. And most of the stuff out there isn't. Digital people simply don't understand that they can read documents and figure out how to do this stuff.

Another reason to go for recordability -- I mean, people say, "Well, I don't need to record it. I can just replay it and see it in high res," and all that sort of stuff. How many people out there really feel like they could, if they went back to the labs now, run every program that they were able to run three years ago? Anybody?

How many people feel like they could play every videotape they played three years ago? No problem, right? So, 15 years ago? I can play all the videotapes I made 15 years ago. I can't even find the hardware that I was using back then. So if we want to communicate this stuff to the future, which is what civilization is about, we probably ought to record some of it.

I like the fact that PCs are portable. It really opens a whole lot of ways of getting these things out to people. You can ship the disk, throw the thing in UPS -- it's not the kind of thing that you want to do with an ALLIANT, I would guess, right? -- and have it show up someplace. And boy, there's a lot of people out there that can take things out of boxes. You just have your auto boot thing, you plug it in and it comes up and does what it's supposed to do. You can take it to the cottage, that kind of stuff. That's wonderful.

The last big point is affordability. The model that I've kept in my mind for many, many years has been the three-year-old Buick. The "I want to have systems that are right at the entry level" is right about what it costs to buy a three-year-old Electra 225. That's about $10,000. That's the kind of money that we in universities, and I suspect scientists in labs and things, consider to be commodities.

A junior professor can go into his department chairman and say, "I need $10,000 to buy a visualization thing," and he'll probably get it at the end of the year when there's some money left over. If he walks in and says, "I need $40,000," then it's, "Well, write a grant." It's a big difference.

I was real impressed one time about 10 years ago when I went to Livermore and saw they had PDP-11s in general stores. You could just go write a requisition and one came down the line to you. That's what I like. I like the idea that these things are effectively -- in fact, in our lab, they're virtually disposable. We don't fix them. We just stick them in the corner and scavenge. It's not worth fixing them. They cost too much. Nobody knows how to fix them anyway. Crockett and Tubbs may need a Ferrari, but I think that scientists need a three-year-old Buick.

Thank you.

HABER: Well, I guess that shows hardware types don't have to be hard-nosed. I thank the speakers for a series of very interesting presentations. We have about half an hour left for questions from the audience. If you'd come up to the microphones, we'll take your questions.

Q: I'd like to know whether you're going to be addressing image processing, besides just drawing polygons --

HABER: Excuse me. Would each questioner please state their name and their affiliation?

Q: I'm Bill Herbert from the University of Wisconsin.

HABER: Thank you. So who on the panel would like to take that, the question about addressing image processing?

A:Lou Doctor: I think everybody is pretty much aware of how important handling image data and being able to process it is. I mean, I think companies like PIXAR have shown people that images don't have to be flat anymore. So I think that what you've got now is this merging together of architectures to be able to handle both pixel data and geometry. I think everybody is also anticipating 3-D voxel data sets for the next step. But I would be very surprised if any of the other three hardware panelists are not doing a lot of work in the imaging area, and image processing area as well.

A:Frank Moss: I'll just add to that. I would agree with Lou. I would say that one of the key problems is that image processing has been extremely expensive until now. In our approach, we're going to try and achieve that within the context of a graphics rendering system by exploring concepts like virtual pixel maps that give applications access to pixels in order to provide special-purpose image processing; but with the same idea of trying to avoid special-purpose hardware for that capability. It won't meet every need. There will still be special-purpose image systems. But, again, I think the trend will be toward trying to amortize the cost of hardware and systems.

HABER: Question from over here?

Q: Art Olsen. Molecular Biology, Scripps Clinic. You talked about the computational hardware. Hardware is the general title. I'd like to ask the panel in general about input devices which are certainly part of the scientific visualization process. In particular, the highly interactive part. And I'd like to know what your opinions are of both current input devices and what direction you see them going in the next year. Just as an opinion, I think we've taken several steps backwards in input devices. A mouse is about the worst thing that I can think of for interacting with my data.

A:Lou Doctor: I'll take that one. We've actually been doing a lot of things, both with -- although the next worst thing is a knob box. But these knob boxes do give you at least a few different variables to handle. Actually, in our booth at the show, we had a six-axis torque ball. It's a product that came out of Germany; and it is essentially software plug-compatible with the knob box - which is sort of a neat feature - and it even can be daisy chained. So we'd like to be able to see an environment where scientists can plug together their input model and, potentially, even have a multitude of these devices in front of them to grab. The six-axis torque ball is a nice thing because it's very intuitive. We also saw the data glove and things, but we weren't running that in our booth.

A:Tom DeFanti: I'd like to add that if you really want to be inspired, Art, I think you ought to spend some time in Myron Kruger's space in the art show. Myron has a thing that -- it's actually hard wired -- he's never even put it to PC boards -- that will track your finger. If you put up five fingers, it erases the screen. It's got a little critter with 150 rules. Myron has done this as an art piece. I think, with substantial support, like a little bit of money, he could actually turn this thing into something that people could use and program as a good interactive device. Of course, it involves having a video camera on yourself, but I think it would be kind of nice to have.

A:Bob Haber: I also think that this is all going to become a lot more important in the scientific domain. Engineers, for a long time, have had to deal with very complex geometries and model building prior to simulation. From what I've seen in the pure science applications, they're either dealing with very simple domain geometries to begin with, or they can generate them from a relatively small amount of information. I think as time goes on, the scientists will also have to tackle much more difficult input problems. I think it's an important question. Okay, next?

Q: Olin Lathrup, Cognivision. I thought it was very interesting to see the disparity between Lou's vision and the two gentlemen from the high- end workstation companies. One way of resolving that, which I didn't see any mention of at all, is to look at the relative prices of the various subsystems that go into a high-end workstation. The memory subsystem, the CPU, the mass storage and the graphics. I would think that the expense of the subsystems, the ones that are more likely to be shared in a back room than the cheaper ones that could probably go on people's desks...but I didn't hear any comments on that. I'd like to hear some comments on where you think the prices of the individual subsystems are going and which ones you think will dominate.

A:Tom Jermoluk: I can answer for our particular workstation. Our vision is that the graphics subsystem is still the majority of our cost within the workstation because it's the focus of our effort. But that's sort of misleading because it depends on what entry point you get into any particular product line. As you add more and more processor onto the host side to tackle the tougher applications, the image processing applications and computational fluid dynamics and other things, then you'll start to see the compute side of the workstation go up in price relative to the graphics performance. So much of that is dependent on what the application user needs and where they get into the line. I think that's why it's important to have a broad range of product available to meet the breadth of those applications.

A:Lou Doctor: The other thing that we've seen is that there are two kinds of people out there right now. There are ones that are going to buy computational servers, and there are ones that are not. The ones that are going to buy computational servers are generally looking at a mix of high-end and low-end workstations, as well as their computational server. Since they have to do their visualization on something today, they are buying high-end workstations to do it. The alternative approach that I was talking about, where you take video right off of your server, you essentially can spend the same amount of money you would have spent for several high-end workstations for shared access; and you get the server for free. Because for that same amount of money, everybody can have access over the network to the computational resource and some number of screens can be made available as a direct connect capability. If you were going to buy a server anyway, then it certainly makes a lot of sense to plug graphics screens on to it; and then you get the graphics for free. So one way or the other, I think there's a nice approach there by trying to connect multi-channel video to high-end servers.

A:Frank Moss: Well, I'll answer the question a little differently than it was asked. It was asked as, what would be the relative cost of the components. But I think the fundamental question here is, what is the role of the client-server model, and what role machines such as Stellar's are playing in that model, versus Lou's vision. First of all, I think one of the things that we find, just to go right to the bottom line, is that a significant number of Stellar systems are sold as compute servers. One of the reasons for that is, again, if you're able to solve the problem of avoiding special-purpose hardware, then the price/performance for a graphics supercomputer system can be very, very effective for a compute application. In fact, in many cases, we find that our systems are competitive, or more than competitive, with similarly priced systems that provide compute only. So, indeed, I think an important trend will be to see these systems not as single-user systems, but as hubs of visualization environments that will provide connectivity to supercomputers. There will also be servers of various types to networks of PCs, lower cost PCs and workstations; which incidentally, in response to Tom's comments, are very much a part of the picture we didn't discuss today, but I think will be tied into a total environment.

A:Bob Haber: I think it will be very interesting to see how these new supercomputing workstations really pan out in the marketplace. When you listen to the vendors, sometimes they call them workstations and sometimes they call them servers. If you look at the size of the boxes, they look more like servers, I think. So it will be interesting to see how that really pans out. A last point on this question is, I think you can't look just at the individual systems and their prices because in all this high- performance hardware world, the chief issue often turns out to be communications. So you have to worry about the communications issues between them. You can't just take the individual pieces.

We'll go on to the next question over here.

Q: Marcelli Wein, National Research Council of Canada. If I interpret, first of all, that this is a session on architecture strategies, then I could -- I want to talk about window systems, X Windows in particular, and observe that the architecture that you're describing -- and there is some commonality among the architectures suggested -- that the mental model of the architecture that the X Windows people carry is at best irrelevant, and at worst, could be a significant setback to computing in the US. If you look at the architecture, at the relationships you visualize between the numeric server and graphics server, none of them fit the current model. So what's going to happen, do you think?

A:Lou Doctor: I don't completely agree that it doesn't fit very well into the X Windows model, because I think the -- all along, at least in our view of things, the window system is the control structure and today's networks are more or less the command and control backbone of the system. For example, we've done a lot of work in implementing high-performance graphics and window systems combined, as have the other high-end super workstation vendors. In all cases, the window system and the graphics subsystem exist more or less side by side and peacefully coexist, meaning the window system tells the graphics system when it's doing something that it needs to know about. And I think the same thing will hold true with these multi-media systems, that you'll actually be able to run in the context of a Macintosh or X Windows environment pretty transparently.

A:Bob Haber: Was the gist of the question about how the X Windows client-server model is distributed across the network? Was that what you were getting at? Yeah. I think that is a legitimate question. I've been giving some thought to distributed processing environments for this kind of work, and I haven't reached any firm conclusions, but one of the questions I'm beating on is really that kind of client-server split, at the X Windows or even at the PHIGS+ level, is that split across the network the right level of granularity for a well-designed software architecture. Maybe the next panel will get into this. I think that it may end up that you're doing X Windows; but you're doing it on a single processor and that there is data transfer going on at a higher level between the modules in your software architecture. I think that's an interesting question. I'm not 100% sure what the answer is at this point.

A:Frank Moss: I'd like to respond, if I may. Yeah, I'd like to echo what Bob just said. I don't think that the current X Windows model necessarily stipulates whether the systems are configured in client-server configurations over networks or within a single system. However, when you have both your compute piece of the process and your graphical, user interface process in the same system, you don't want to bottleneck the performance. Today, the X Windows system specifies a Berkeley-like 4.2 remote procedure call mechanism between the client and the server. I can tell you that our experience has been that it's quite straightforward to replace that connection, without impacting application programs, by a shared memory kind of protocol between the client and the server, and you can get high performance. In the Stellar system, for example, the software is never a bottleneck to the hardware, even though we're performing through X Windows and PHIGS. So it need not be a bottleneck when you're in the same system.

A:Tom Jermoluk: I guess we have a little bit of disagreement with that, in that our notion is that even a shared memory protocol between the graphics process and the server process when you're on a single combined workstation is limiting. Limiting in the way that you can do graphics, as well as in the performance. So what we think of as each X process on the system has access to the hardware environment and that it in fact can control all the input and output and the normal window manager functions of the system. So the key there is to provide a hardware architecture such that it's totally transparent to that particular client at the time the rest of the mechanical control that the server would have done. That model extends well across the network, so that you can actually open the window into the supercomputer, or the window into the PC or whatever other sort of system you have holding on. And I think that that will be the more interesting area for the 3-D people, at least, in the X and NeWS area to try and standardize, as how those models appear across the system.

Q: My name is Mike Kaplan and I'm at Ardent Computer. I don't want to raise a lot of controversy or anything here, but then I seem more like throwing the baby into the pit of worms. My question is specifically for Lou Doctor. His vision of the future, where you have centralized computing, very expensive resources and then slightly less expensive, but pretty expensive resources which do all these specialized things, and then you have scientists who are -- and these resources are all controlled by some central governing authority of some sort, many governing authorities. And then you have users who are allocated these kind of things on their desks that tap into it. It sounds an awful lot like time sharing to me. And my question is simply, if time sharing was such a desirable thing, why did workstations catch on?

A:Lou Doctor: The obvious difference to me between my model and time sharing is the fact that you can disconnect the network and have a perfectly usable stand-alone system in my model. Because the fact is that you're really only relying on the network, if you will, or the remote resources for this peak computational throughput, the on-demand requirements. And the fact is that, within these organizations, I mean like it or not, go to the largest national laboratories and walk into every office and you see a Macintosh, and walk into the computer center and you see Crays. I realize that the counter-cultural view might be to distribute all of this processing power to every user and get rid of the centralized resource; but people that work at large companies and labs that are doing scientific visualization are using supercomputers to do them. I personally don't believe that a workstation is going to deliver the power of a supercomputer for the next 100 years.

Q: My name is Rob Wolff from Apple Computer. A couple of comments, I guess. Let me be somewhat consonant with Lou's vision in the sense that there will always be a requirement to do something that's beyond the capabilities of your own $10,000 workstation. We've seen that for years. However, the power of one of the long-term objectives -- and hopefully, this growth curve will increase -- is to continually bring power to the people. All right, so we're old, right? Some of you guys weren't even around. That's the one last night, it wasn't even important, we used to talk about this. But it's obviously important to bring more and more power to the desktop, and we're seeing this. This is a requirement. If you look at the distribution of time that a scientist spends doing various activities, very little of it is spent doing batch processing with a Cray or using high-end computers. When the capability is needed, though, you should have it on demand without having to go and make some major effort to march off to a computing center or whatever. But let me take issue with something that DeFanti said with regard to the hackability of the MS/DOS environment. What I've seen over the last 10 years -- fortunately, I grew up in a physics environment where we did theoretical physics and we didn't worry about which boards worked with which software and whether or not we could software hack the particular system and so on, and we spent a lot of our time doing physics. Although I strongly believe in an open architecture system where you can tap into the creativities that exist in the general industrial, commercial and university population; it's very important to ensure that the open architecture doesn't end up causing the scientist who just wants to do their science and doesn't want to blow a light bulb or a beaker or something like that. They don't want to spend their time hacking hardware and hacking software. If you then extend the concept of distributed computing to the academic domain and realize that the end result of research is teaching, you don't want to have your students hacking this either. You want to have your students learning. In a very simple environment they can take their computer, their personal visualization workstation, rather than PC, and put it on their desk and be able to do academic computing and tap into the larger resources and be able to learn in an environment without having to worry about whether or not their software and hardware will work together.

A:Tom DeFanti: I'll answer that. It seems to me that you've failed to form that in the shape of a question. Next contestant, please.

Q: I never form questions. You know that.

A:Tom DeFanti: That's a good question. I don't like doing archaeological digs on computers either. I spend an awful lot of time figuring out what people really had in mind when they designed these boards, since they're not allowed to document it for ...

Q: That's because you have MS/DOS, because you have an MS/DOS machine.

ATom DeFanti: Right. Well, I spend a lot of time doing that. And I don't like that except for the fact that it does give us a way of getting there. And what people who come into my lab do is, they say, "Well, how do you do that?" And I hand them a stack of purchase orders and they go and Xerox them and send them in and get something that plugs together and works. I think that science itself is extremely idiosyncratic, and in fact I sometimes refer to it as the fractal nature of science. Scientists, by definition, are hanging out in those places like you get in a Mandelbrot set where you keep zooming in and it's still there. And I don't think that anybody can, except through extreme flexibility, attempt to address that market. If you give somebody an application that's already done, well the next week, they're going to want to do something different and of course you can't give them the source code because that's proprietary. So what are they going to do now? Unless they're the military, they can't get you to change your code. And that's a real problem. So I really think that scientists need - - many scientists, those who are on the edge, need to -- which is all of them, really -- need to be able to get at this stuff. The same argument people say about the programing rub. They say scientists shouldn't have to program. But I've been checking around here and Japan, unlike engineers who use packages, 90% of the people who work with supercomputers do their own programing. I don't know whether that means that's the only way to get there, or what.

Q:Rob Wolff: Well, it's almost -- on my computer, that we're in a transitional phase. For example, we don't question the algorithms for sine, cosine, exponential and so on, that come with the math packages that we buy. And at some point in time -- I mean, I like sort of the mathematica approach where I can get to the algorithms that are used for particular simulations or modeling ... equations or whatever.

A:Bob Haber: I think we're drifting away into the software panel here.

Q:Rob Wolff: Right ...

A:Bob Haber: So maybe we should save this discussion for them.

Q: David Mathews Morgan, University of Georgia. I agree with Tom DeFanti about the idea that if it's not recordable, then it's not really science. And I realize that NTSC video is Tom's way of recording. I'd like to get a feeling from the panel as to what they see all the various hard copy devices, where they fit in, especially in the university environment where you can't afford to have every person have his own, say, hardcopy film recorder or even a $7,000 color thermal recorder. I'd like to know where they see that fits in, especially in the university environment.

A:Lou Doctor: Not to be in this visual server rut, but one of the advantages I think of a centralized resource -- Mike Kaplan's probably cringing -- is the idea that people can share things like output devices and videotape resources. We've seen a lot of progress in the color hardcopier case and in the ability to make videotapes reasonably easy and cost effectively. But I think that there is also a need for being able to generate these large image and animation data bases and have those data bases be available on demand as well. So that if you did an animation last year, you might want to actually have it archived back and be able to play it back. Or likewise, if you want to have a videotape or film made, you need to have a resource you can tap on the network to have that done for you. So again, while it might seem like gloom and doom to think about a centralized resource that someone is actually managing and keeping running for you, I think the reality of the situation is that everyone is not going to have a videotape recorder and a hardcopy unit and a Dicomed in their office. They're going to have to rely on a resource providing that output capability for them.

A:Tom DeFanti: Or they could bring in their home little hand-held video recorder and do it that way. No, no. You can plug those things in and it puts out video output. Plug it in and make your videotape.

A:Bob Haber: I'd make two comments. One is that I hope we're not stuck with NTSC forever because it's not such a great medium, and that we might well see digital methods of storing image data as the technology improves that is affordable and you could have in your office with cheap playback systems just like your stereo.

I forgot what my other point is, so I'll go on to the next question.

A:Tom DeFanti: Before you go on, though, I'd like to point out that one advantage of the approach of having a video server is that it's video. You just plug it in and record it. So that's a nice idea, too.

Q: Michael Herman, Apricot in Canada. I have some background in seismic data processing and have thought about this video workstation, live video terminal before. My question to Lou, I guess, is how little computing power can we put in that local workstation if we are shipping video? I would think we would get away with even less power than a 286 might be required in something like that.

A:Lou Doctor: The question was, how cheap can you make it if you're just going to have it be kind of a stand-alone sort of video node or something. I mean, I think I'll probably actually side with Rob on this one, that what we're calling a PC is getting more and more powerful, probably for good reason. I mean, there are still things that I do on my PC that take a long time that I'd like to see sped up. So that when you're at that level of cost performance, though, you're really riding the exact right place on the technology curve with microprocessor chips and reducing memory prices. Because I think what we're seeing, and what seems pretty self-evident, is that while the cost per MEGAFLOP and cost per MIP is going down; the cost of the memory system and the cost of a system that has high bandwidth is going up or staying the same. So that when people try to build these high-end workstations and have to design the memory systems and power and cooling and cache systems to feed these RISC CPUs, I think they have real problems getting the cost down. I think when you're designing in the under $10,000 class, there's just a lot of things you can do using PC technology to get a lot of performance in there, and excellent price/performance. I guess the only other comment I'd like to make is that comments about ... disappearing as a time sharing system may be inappropriate. Really, what we're talking about is lengthening that video cable that comes out of the back of the CPU and shortening that ethernet cable that connects the CPU back to the central resources. So you could imagine essentially 100 PCs in a rack-mounted configuration distributing video out, and really, time sharing would be an inappropriate comment there.

A:Tom Jermoluk: It's an interesting comment, but remember, there's a couple of challenges left to go there. When you want to talk about video, you're talking about lots of bandwidth and lots of memory. NTSC at 30 hertz is about 16 megabytes a second. So if you want to play a few seconds of animation, that's a lot of data bus bandwidth and memory that you'd better have in your system to accomplish that. So I don't think you'll see -- you'll see a lot of interactivity going on there, you might be able to preview single scenes across the network after waiting for a couple of seconds, but you're not going to be dealing with any actual preview of animation or motion. So there's some barriers there yet before you get down into taking live video or things like that down into the lower cost workstations. So I agree a little bit with Lou in the server side of it. That's sort of how we envision that same part of the strategy and that you'll have the genlockable capability and the video recorder capability and the RGB printer capability hooked to some node shared among a lot of people. And our challenge is to make that as easy to use and transparent as possible.

A:Lou Doctor: I have a device at home that receives 100 channels of live video all in real time. It wasn't really that expensive. So I think that what you have to do is, you have to kind of use a little bit of imagination here and what I'll call a leap of faith to put two technologies that are both very well established and moving very quickly, together. I think that while digital data might be perfectly appropriate for shipping data files around, this device that I have at home has tremendous technology in it for receiving live video and being able to display it for me. I think that that's where you have to put these two critical technologies together. I disagree that you want to move live video through the memory system of the device all the time. It's totally inappropriate to move animation rate pictures through the memory system of a server or potentially even through the memory system of the workstation. What you want to do is simply have those live video pictures appear on the screen. So I think we can simplify the problem by taking advantage of another multi-billion dollar industry for commercial television technology.

A:Tom DeFanti: Dan Sandin likes to point out to people that Peoria, has more bandwidth in its cable TV stations than the NSFnet is proposing.

A:Bob Haber: Yeah. I think this will have to be the last question.

Q: Steven Hunt from Imperial College, London. I'd like to pick up on something that Thomas DeFanti said. If we're not taking notes, we're not doing science. We're also not doing science if we can't publish. That may be a bad thing, but we have to publish. I have in mind problems to do with libraries, archiving, referencing. Something has been said about video but I have to say that I'm from a country where we don't use NTSC. We have a better system. And if you change from NTSC to something better in the future, you're going to have the problems we have now. All your archive will now be inaccessible to you. How are we going to publish -- if it needs a scientific supercomputer to display the data and to understand it, how do we publish the results?

A:Tom DeFanti: There are digital devices that go quite nicely between NTSC and PAL and they are routinely used and they're getting cheaper and cheaper. So I don't think that's a serious problem. In many places in Europe, it's really straightforward to find triple standard systems. Of course, the French think they have a better system, too. So I don't find that to be a problem. NTSC video isn't great. The preferred method that we've been recording lately is on BetaCam, which allows you to record essentially an RGB through a simple transcoder, and that's very nice. Because when you play the stuff back on RGB video projectors -- and most of them are these days -- or monitors -- most of which take in RGB -- the stuff looks pretty much as good as it did originally. So that's a mechanism that I'm currently recommending, so you don't get the NTSC zippers and the other kinds of problems that better systems have tried to replace.

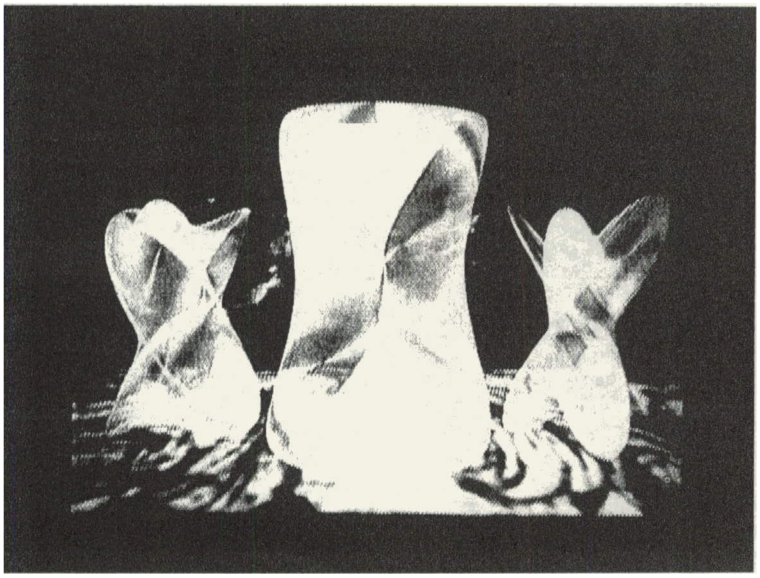
Q: The problem is a bit broader than that. I can't publish a video. The journals don't even accept stereo, usually.

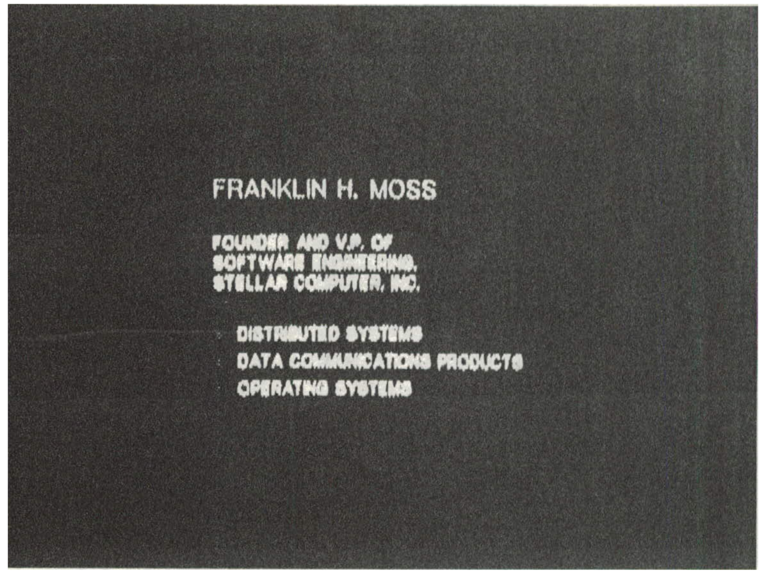
A:Tom Defanti: Well, that is a historical problem that has to do with people who say they can't allow people to publish on video. We even publish them in the SIGGRAPH Video Review and have something like 35 hours of stuff. And although it doesn't count as a journal, that largely has to do with that fact that ACM doesn't allow SIGs to have journals. And people in other fields, in medicine, are starting to publish refereed journals, like in surgery, using video. That argument made a lot of sense 10 years, 15 years ago when people didn't have video cassette recorders in their home, but people have them now. So the distribution and playback devices are out there. If your library doesn't have one, I think you ought to get on their case.

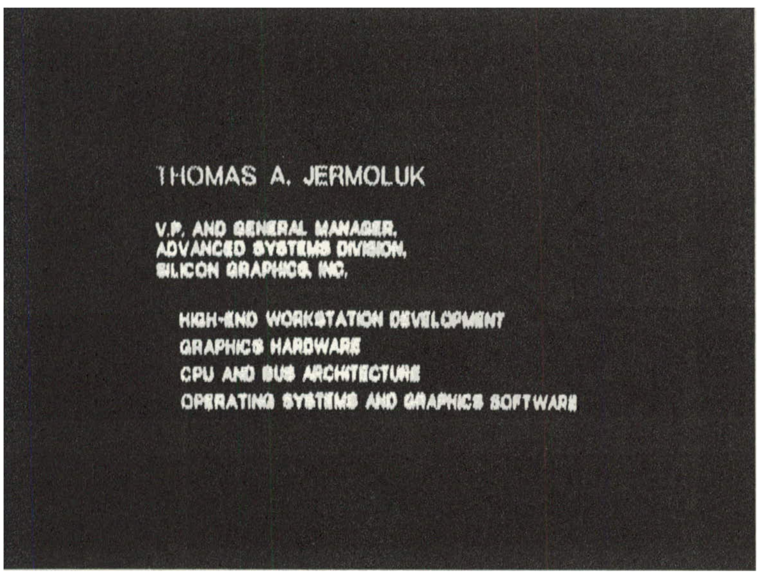
A:Lou Doctor: And lastly, I would just say two things. Any solution to this sort of visual networking problem has got to be an international solution. So it's pretty clear that we need contributions from Europe and Japan and some ability to work internationally. So I don't think -- when I say using commercial television technology, you should not necessarily read, "NTSC, poor picture quality, US only." You should read, "Large market. A lot of R&D investment. A huge installed base. And the ability to leverage very large quantities of product." So I think that you've got to solve all of those problems, picture quality and international distribution and the ability to use these networks worldwide. Those are all solvable problems. There are some extremely bright engineers that I think can solve that.

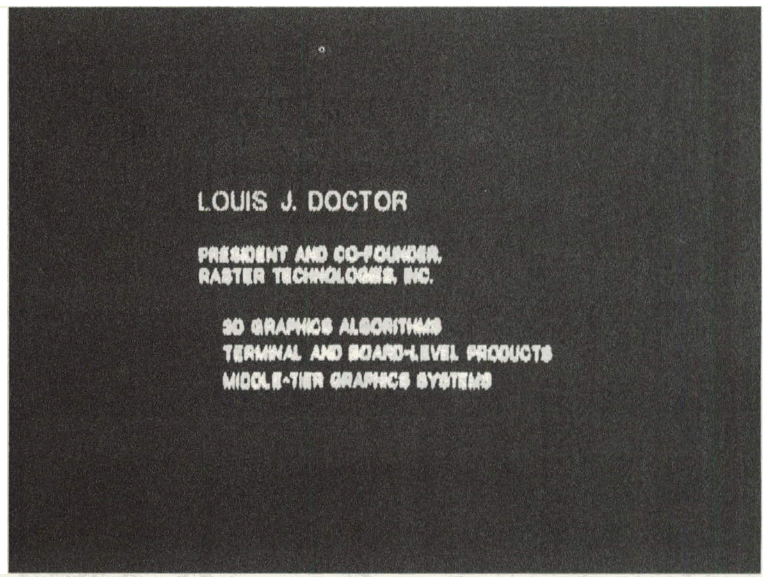
A:Bob Haber: I think for the scientists, our biggest challenge is more of a political one. Right now, it's still hard to convince a lot of the conference organizers - other than SIGGRAPH, which is entirely visual. You go to a typical scientific conference, you've really got to pound the table just to get a VCR. So we've got a big battle ahead of us, but we should fight it.

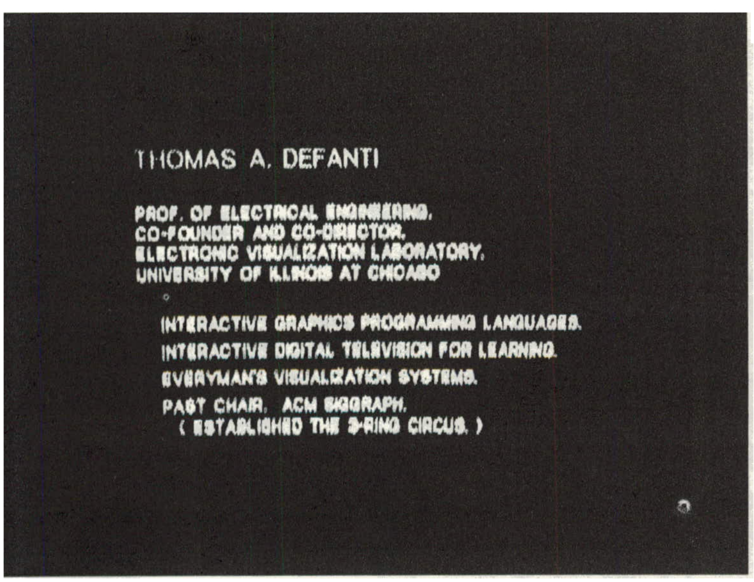
I think at this point we'll close the panel. And I thank the questioners for their good questions and the panel.

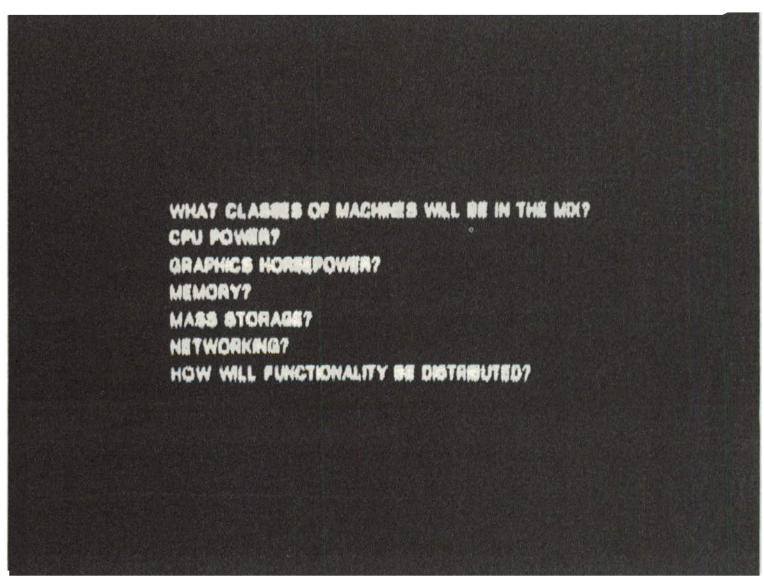




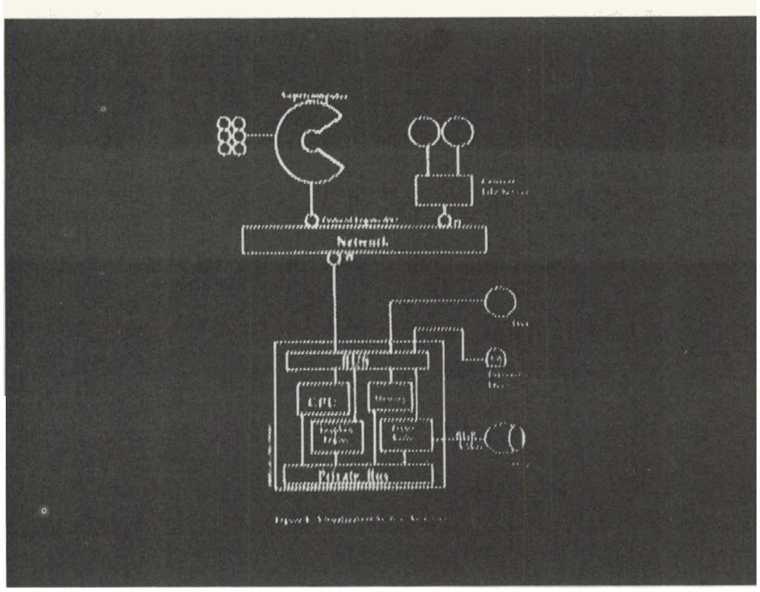


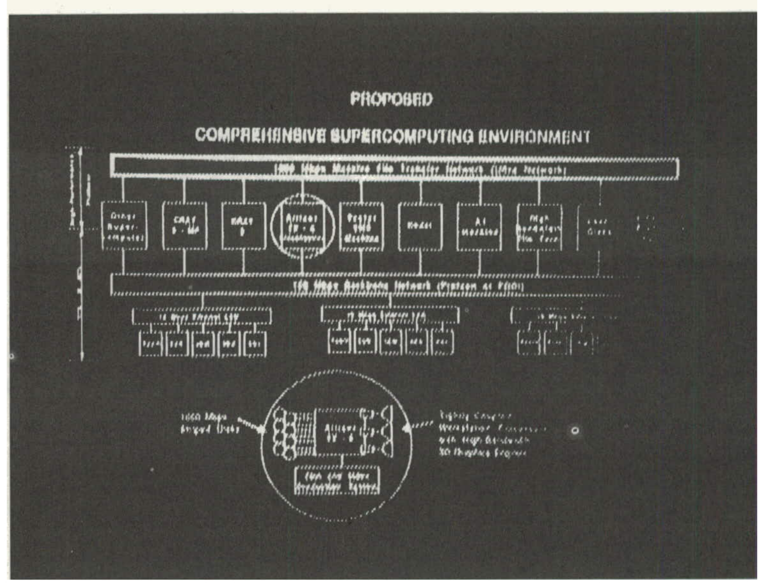






Robert Haber





MASS STORAGE DEVICES (DISKS)

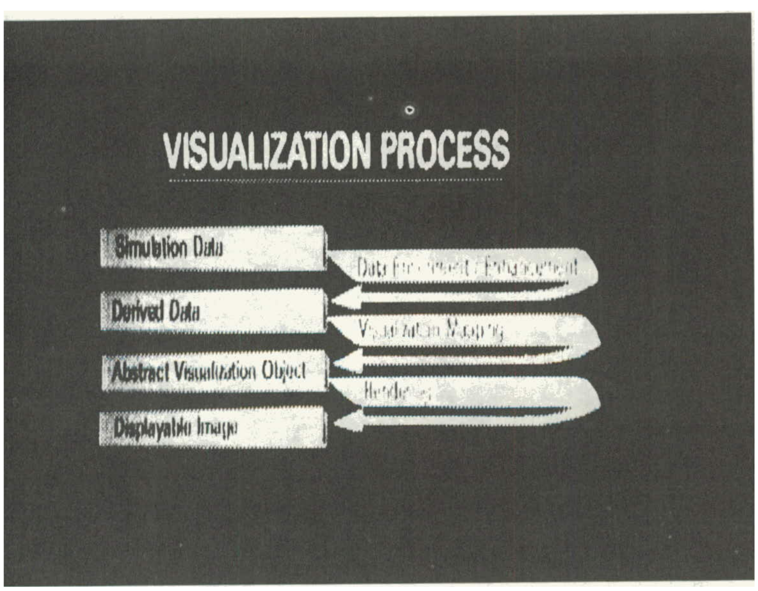
APPROACHES TO SPEED

- SOFTWARE STRIPING
- PARALLEL TRANSFER DRIVES
- DISK ARRAYS

GENERAL-PURPOSE PROCESSORS VS. SPECIAL-PURPOSE HARDWARE.

TRADE-OFFS BETWEEN PRICE PERFORMANCE AND FLEXIBILITY.

WHAT ARE THE KEY BOTTLENECKS IN THE VISUALIZATION PROCESS AND HOW WILL HARDWARE SOLUTIONS ADDRESS THEM?



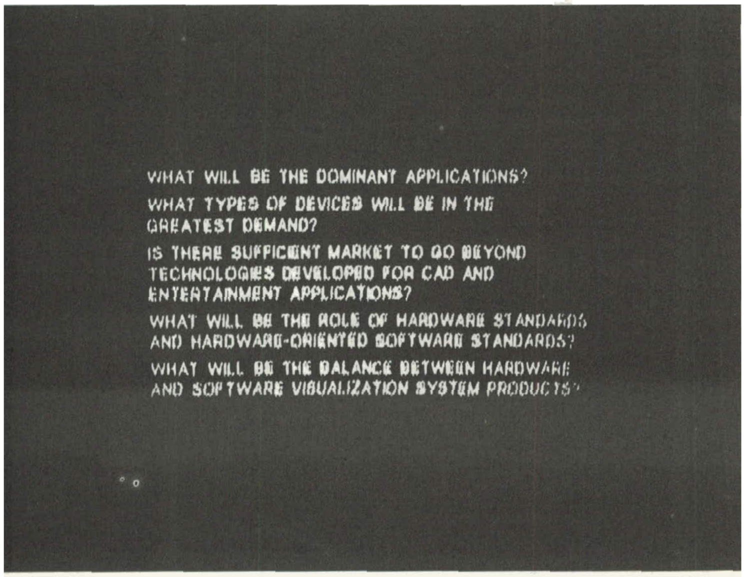
RUN-TIME MONITORING

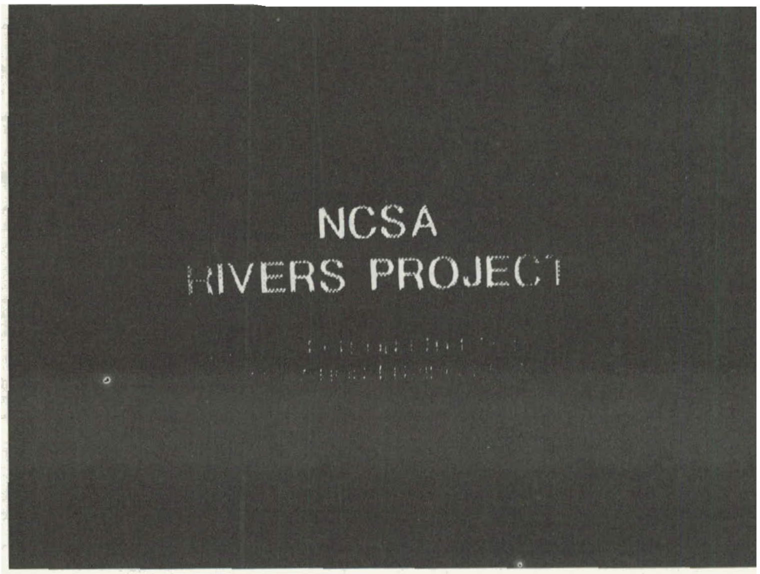
A) SLOW SIMULATION, STILL-FRAME VISUALIZATION
TAT + RAT .LE. SAT.

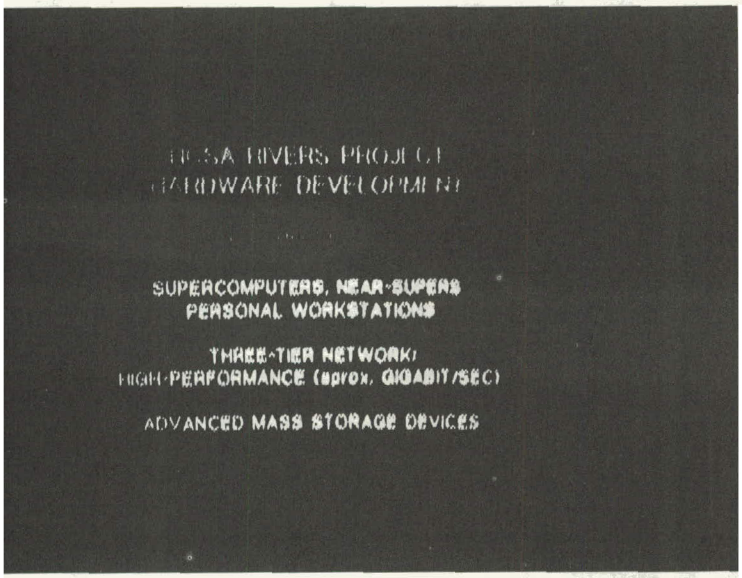
B) SLOW SIMULATION, ANIMATION
GLOBAL TAT < SAT.
LOCAL TAT + RAT .LE. 1.

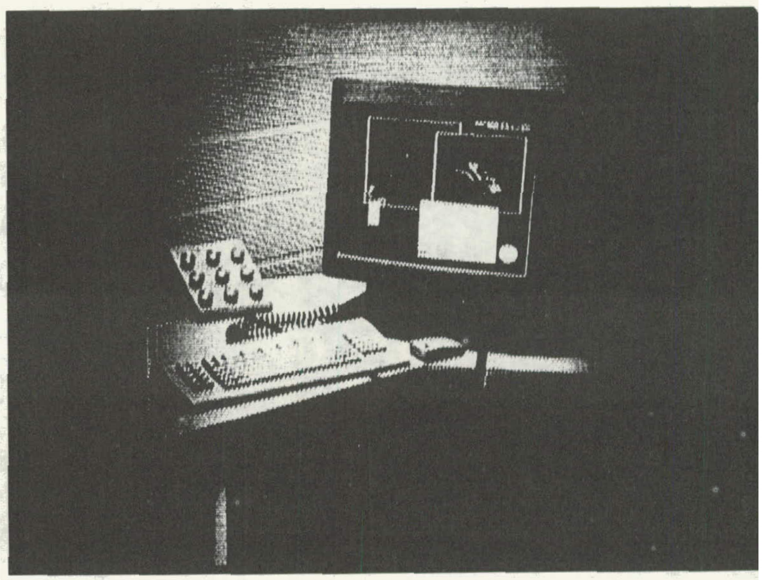
C) REAL-TIME SIMULATION AND ANIMATION
SAT + TAT + RAT .LE. 1.

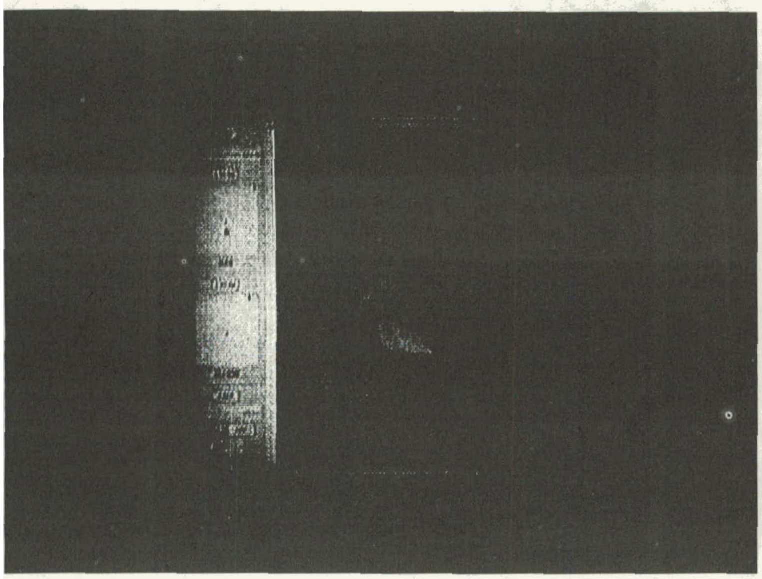
(ALSO APPLIES TO INTERACTIVE STEERING)

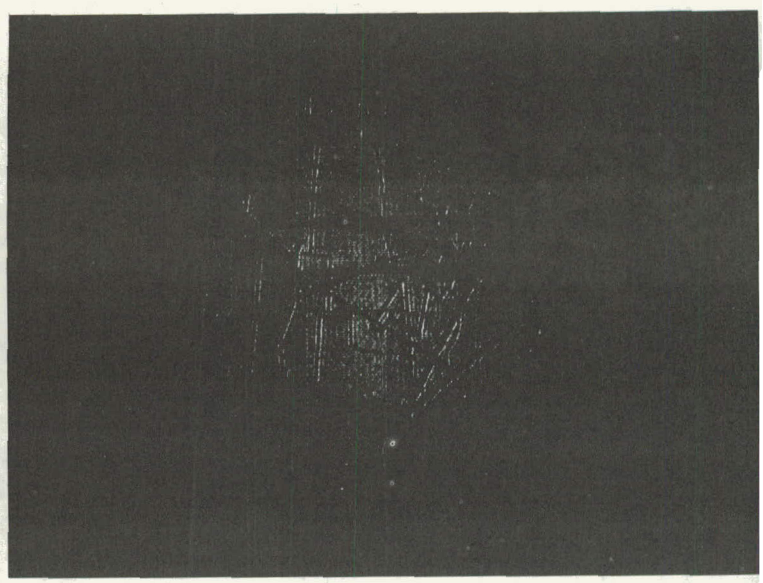




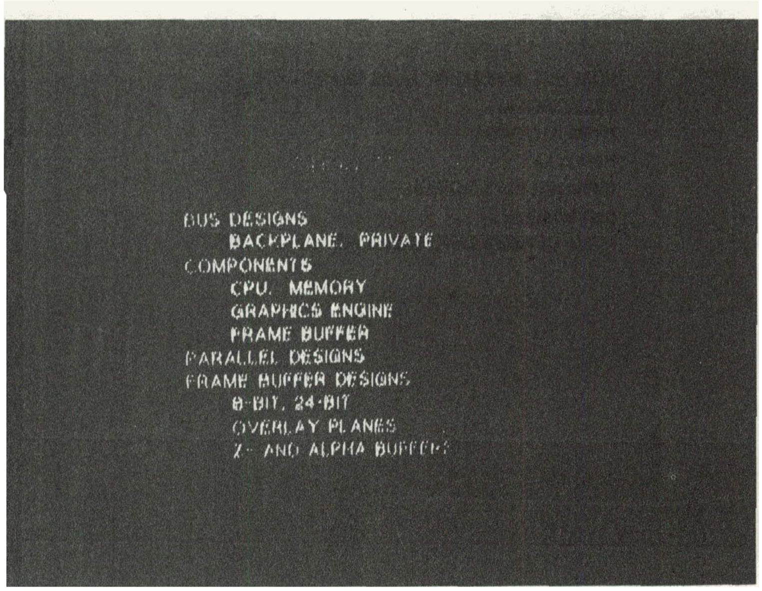


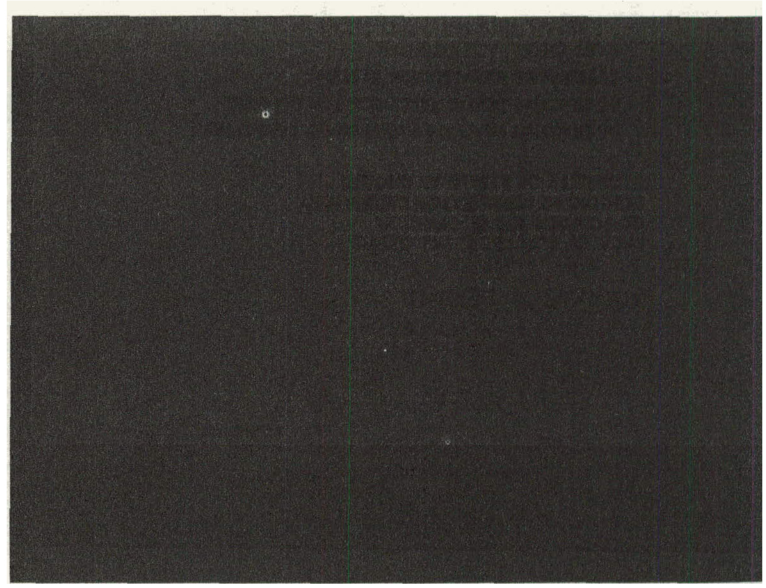


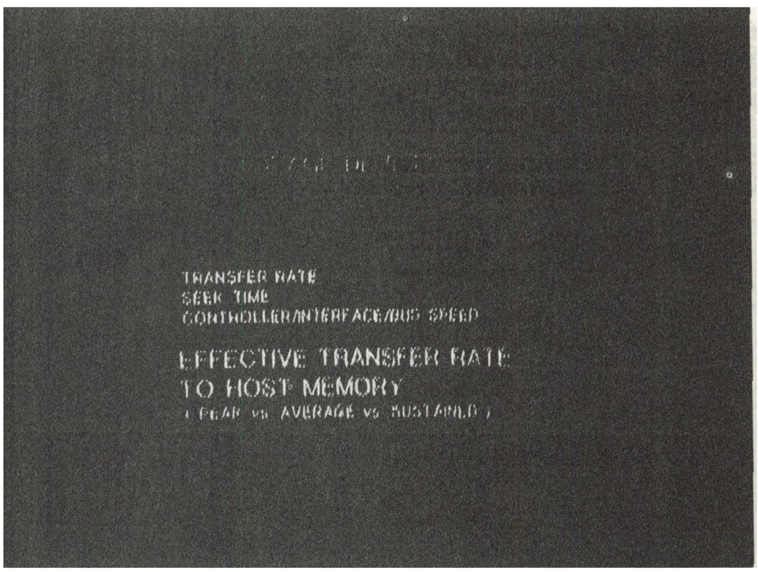


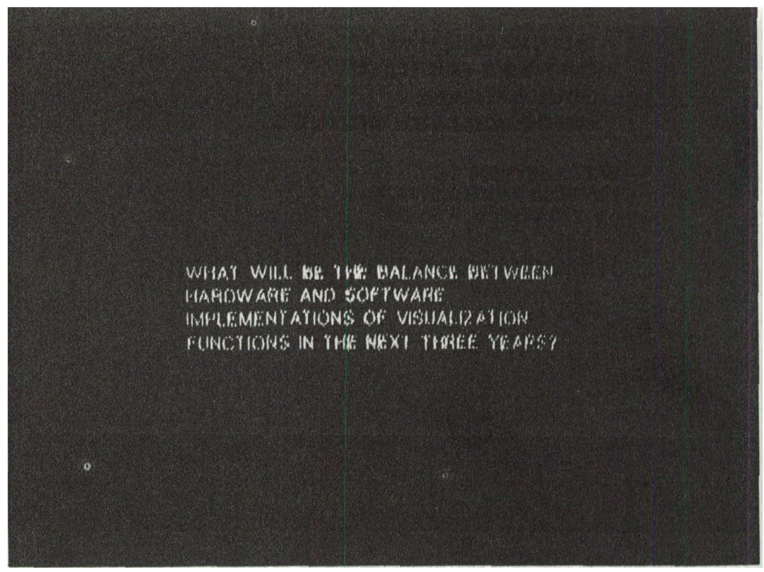


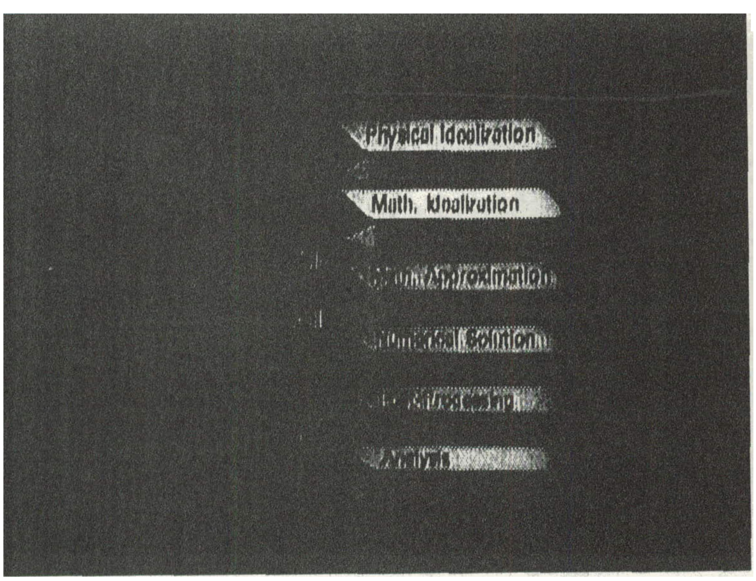
Robert Haber

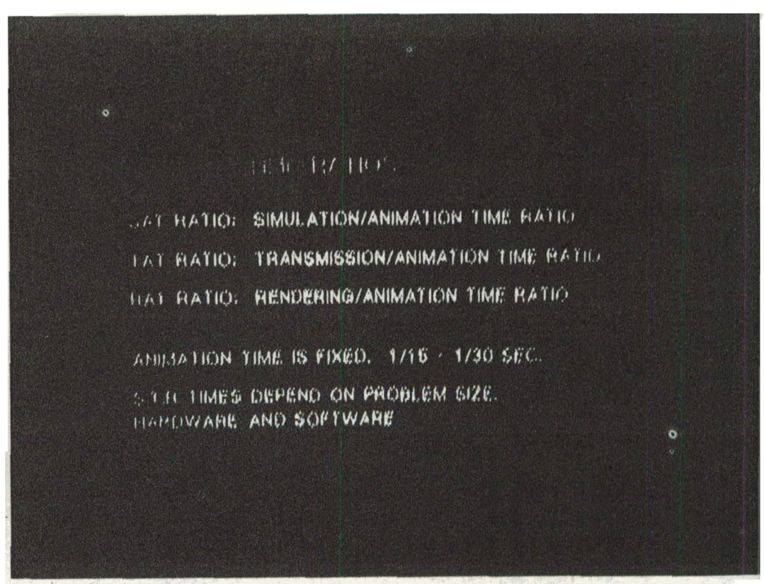




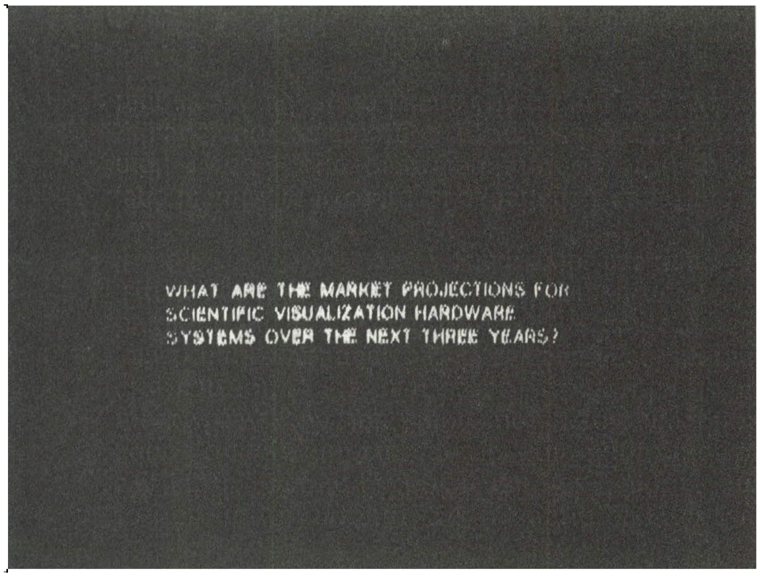


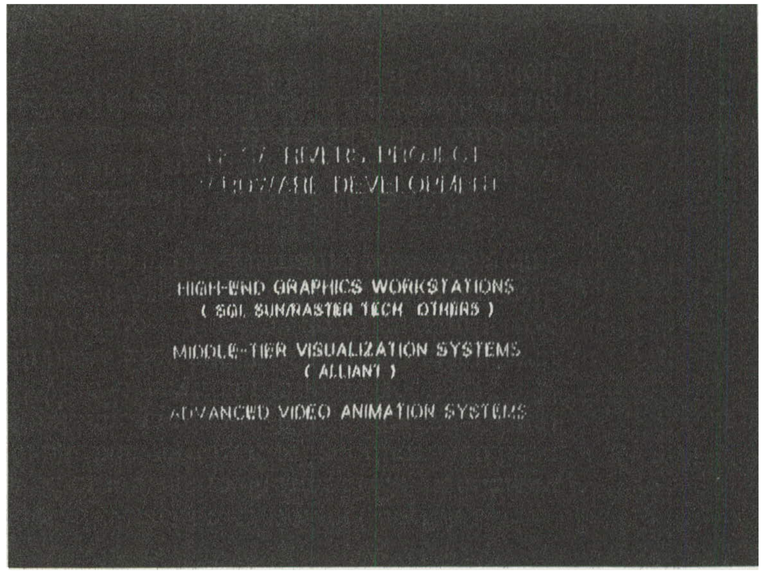


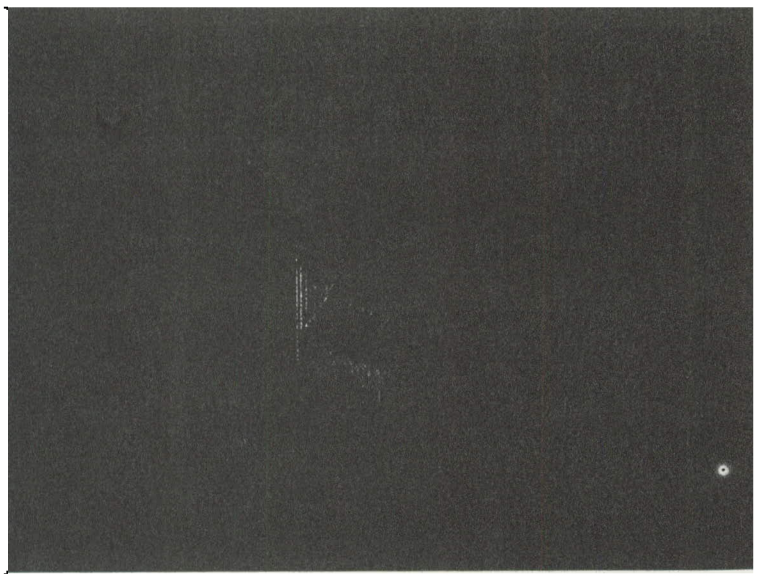


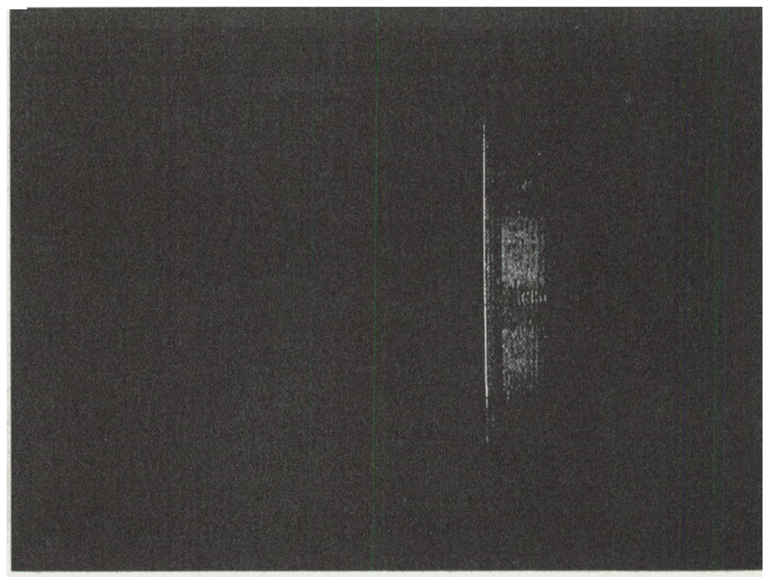


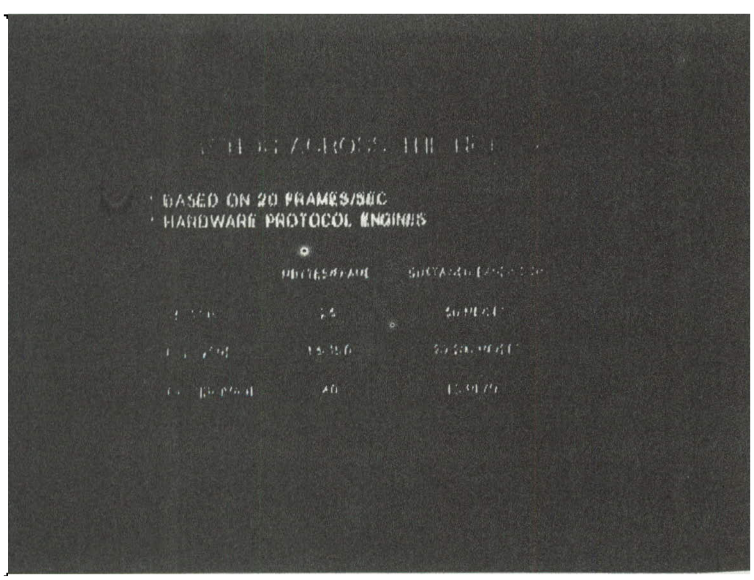
Robert Haber











Robert Haber

Hardware Strategies for Scientific Visualization

Siggraph '88

Louis Doctor
President

Today's Systems Do Not Solve Scientific Visualization Problem

- Too Slow
  - Systems Need to Be >100x Faster
- Too Expensive
  - Most Users Can't Pay $100K
- Too Impractical
  - The Most Expensive Systems Have the Least Software

Several Classes of Workstation Will Evolve

- Commercial
  - 386, X86 Based - Not Interesting for SV
- CASE, Desktop Publishing
  - RISC or 68xxx Based - Not Interesting
- CAD/CAE
  - RISC+FP or Equiv - More Interesting
- Multi-Media
  - Cheap CPU + Video - Most Interesting

Future Systems Will Consist Of:

- Supercomputers
  - 1 or More for Entire Company
- Mini-Super Visual Servers
  - 1 Per Department
- Multi-Media Workstations
  - 1 Per User
- Network
  - Designed for Data and Pictures

Visualization Servers Are a New Idea

- Interface Better With Supercomputers
- Let You Run More Software at Your Desk
- Render Images Faster Than Any Workstation

Today We Can Demonstrate A Visual Server

- Raster and Alliant Have Merged
- Both Product Lines Continue
- Visual Servers Drive Screens Today
- The "Look and Feel" of a Workstation

Multi-Media Workstations Will Come From Today's Leading Workstation Vendors

- Have Live Video Windows
- Have Frame-Grab Capability
- Have Some Limited Local Graphics

ALLIANT

Live Video Input Will Come From

- Video Tape/Disk
- Visualization Servers

ALLIANT

Advantages Over High End Workstations Are Many

- Low Cost-Per-Seat
- Access Resources Through Network and Window System
- Optimize Supercomputer and Scientist Resources

ALLIANT

In the Future -

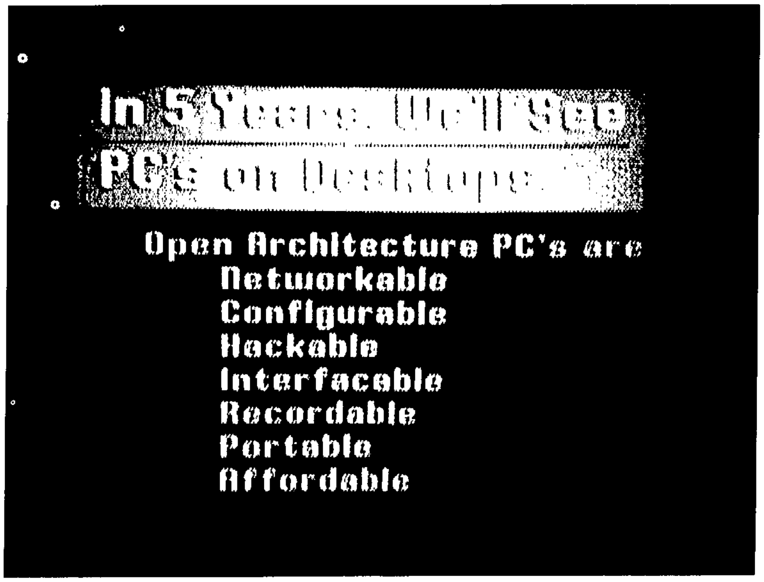
- Networks to Drive Multi-Media Workstations
- Better 3D Workstations for Local Computation
- More Focus on Database Issues

ALLIANT

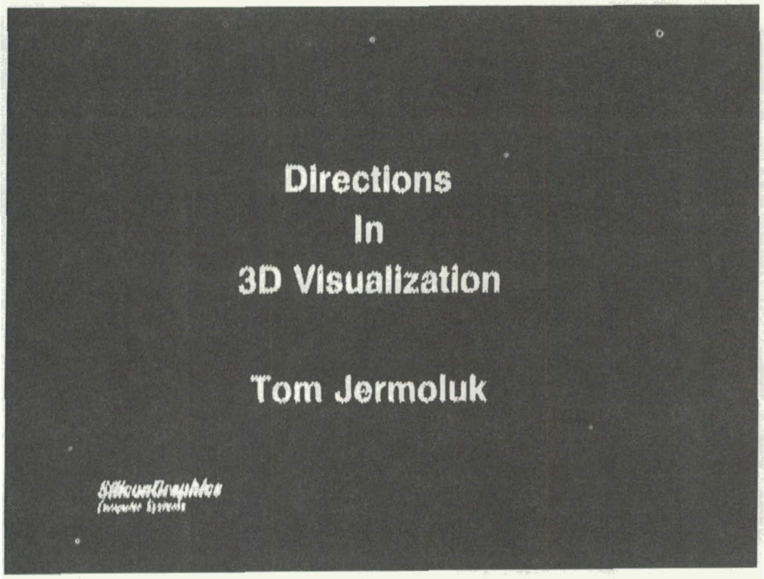
ALLIANT

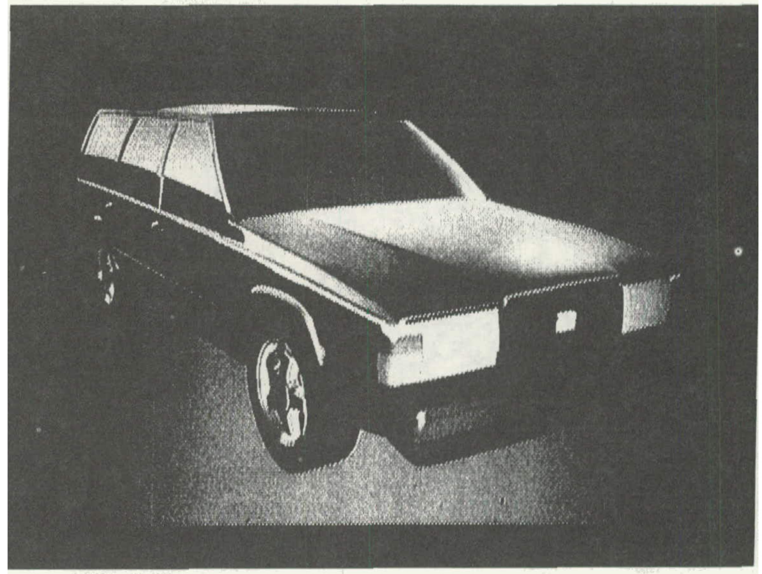
ALLIANT

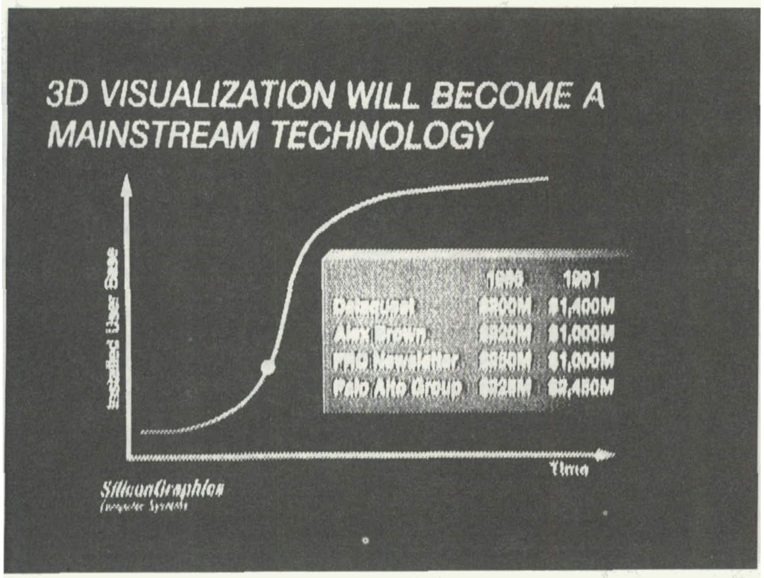
Lou Doctor

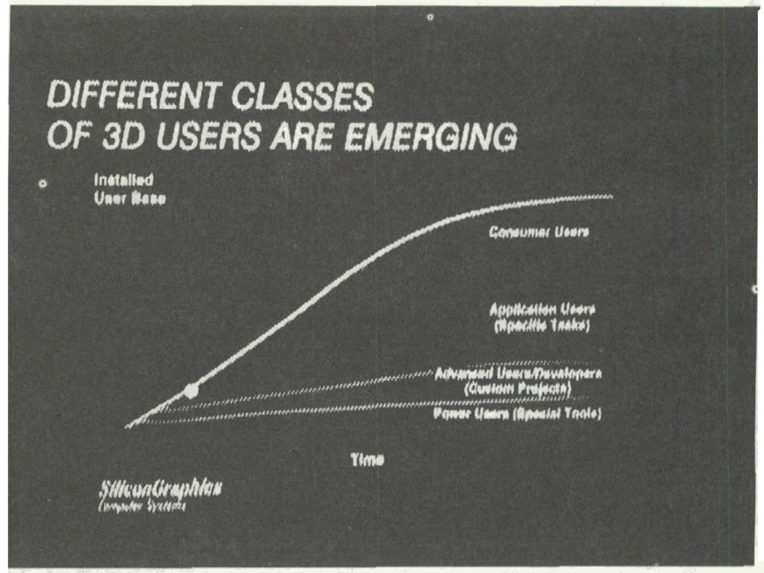


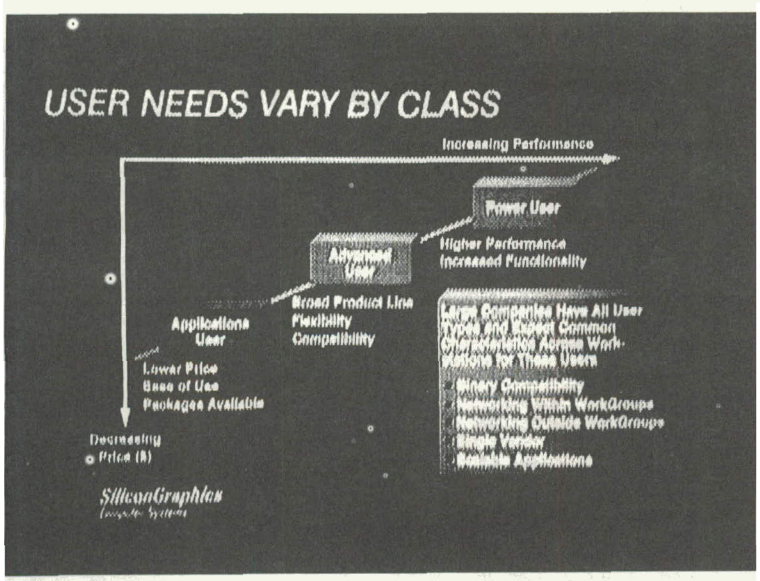
Thomas DeFant

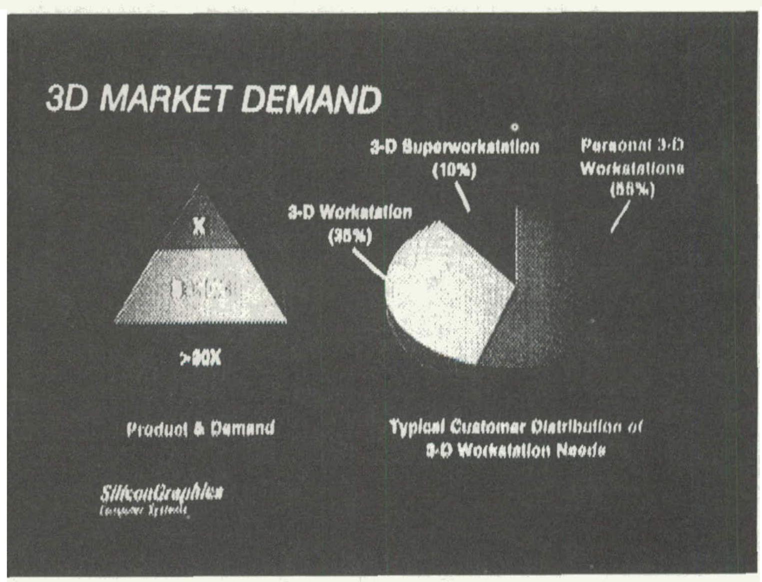

Directions
In
3D Visualization
Tom Jermoluk
SiliconGraphics




3D VISUALIZATION WILL BECOME A
MAINSTREAM TECHNOLOGY
Installed User Base
1991
Dataquest $1,400M
Alex Brown $1,000M
$1,000M
Palo Alto Group $2,450M
Time
SiliconGraphics


DIFFERENT CLASSES
OF 3D USERS ARE EMERGING
Installed
User Base
Consumer Users
Application Users
(Specific Tasks)
Advanced Users/Developers
(Custom Projects)
Power Users (Special Tools)
Time
SiliconGraphics

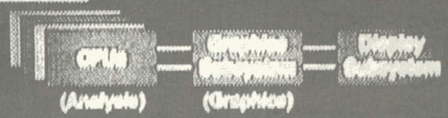

USER NEEDS VARY BY CLASS
Increasing Performance
Power User
Higher Performance
Increased Functionality
Advanced User
Broad Product Line
Flexibility
Compatibility
Applications User
Lower Price
Ease of Use
Packages Available
Decreasing Price ($)
Large Companies Have All User Types and Expect Common Characteristics Across Work-stations for These Users
Binary Compatibility
Networking Within WorkGroups
Networking Outside WorkGroups
Single Vendor
Scalable Applications
SiliconGraphics


3D MARKET DEMAND
3-D Superworkstation
(10%)
Personal 3-D
Workstations
(55%)
3-D Workstation
(35%)
X
Product & Demand
Typical Customer Distribution of
3-D Workstation Needs
SiliconGraphics

SPECIALIZED vs. GENERAL PURPOSE

General Purpose Hardware Trades-Off Graphics and Analysis

Specialized



General Purpose

(Analysis & Graphics)

SiliconGraphics
Computer Systems

SPECIALIZED vs. GENERAL PURPOSE

Interactive Graphics Require Intensive Floating Point Computation

Operation	MFLOPS
Vertex Transform	9.5
Normal Transform	6.0
Normal Normalization	9.5
Lighting	11.0
Clipping	2.5
Projection	4.5
Viewport	3.5
Total	46.5 MFLOPS

(100,000 4-Sided, Independent Polygons/Sec.)

SiliconGraphics
Computer Systems

SPECIALIZED vs. GENERAL PURPOSE

Graphics Computation

Well Understood Algorithms

Large Quantities of Small, Independent Data Items

Graphics Data Expand Exponentially

Data In	Data Out
100 Patches	20,000 Polygons
16 Control Points Each	100 Pixels Each
9.6K Bytes	16M Bytes

SiliconGraphics
Computer Systems

WHERE IS IT GOING?

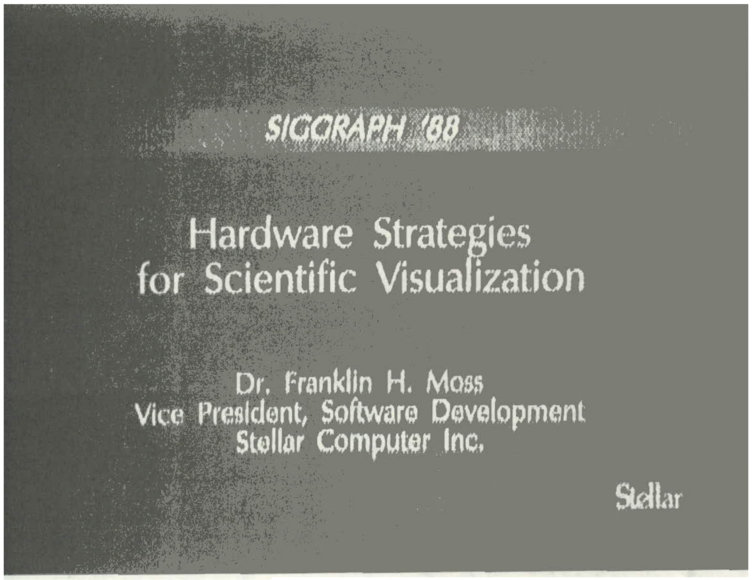
More and More Realism
- Phong Shading
- Texture/Environment Mapping
- Antialiasing
- Surface Representations

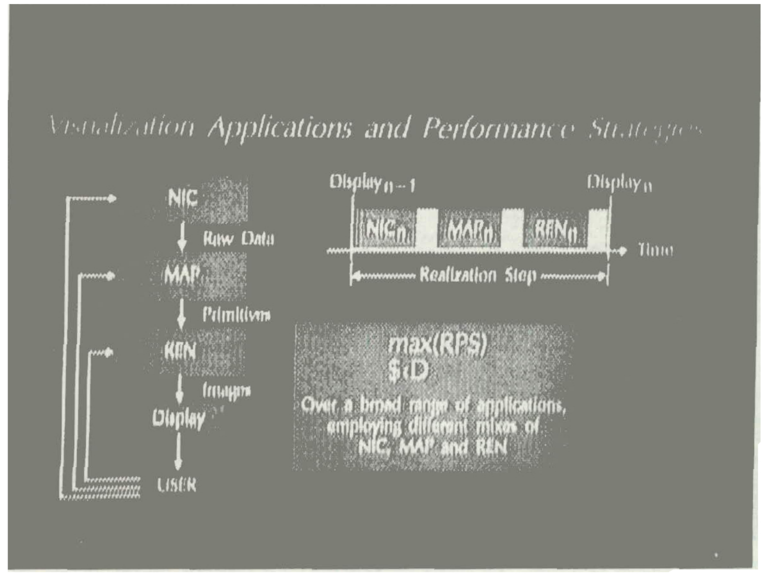
More and More Performance
- Smaller Polygons, Larger Numbers
- Lower Clear Times
- Lower Overhead for Interaction

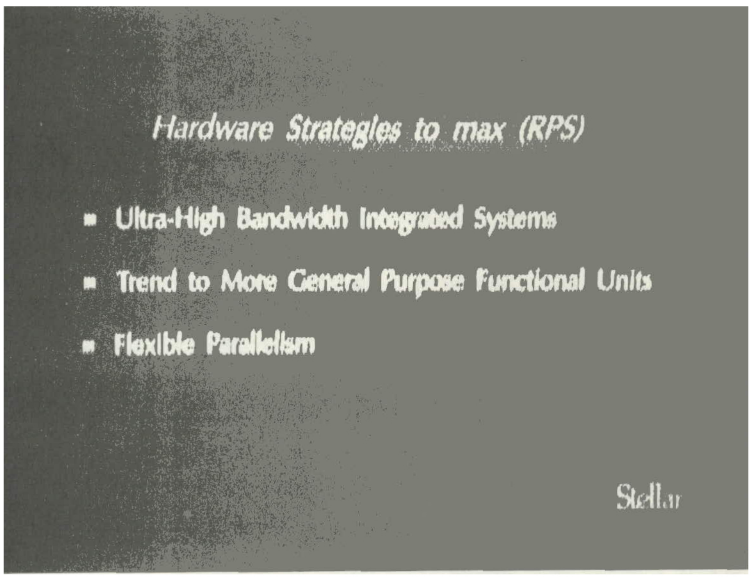
More and More Integration
- Window Managers
- Operating Systems
- Networking

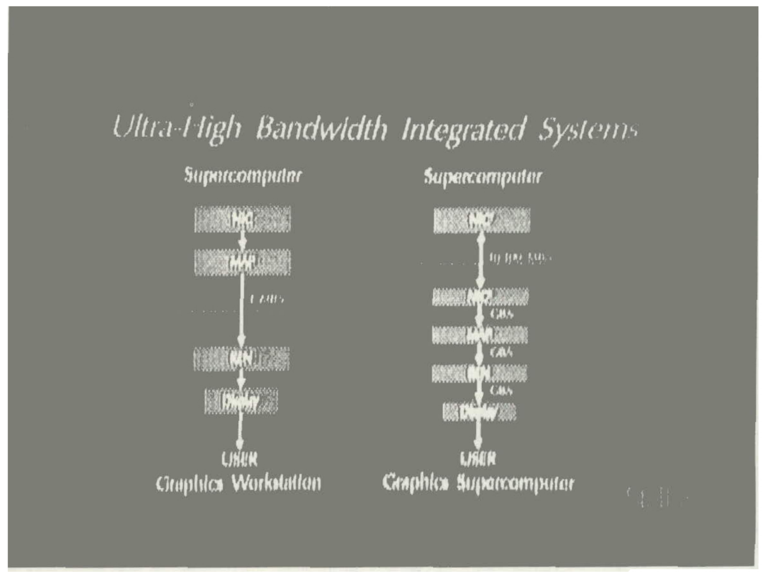
SiliconGraphics
Computer Systems

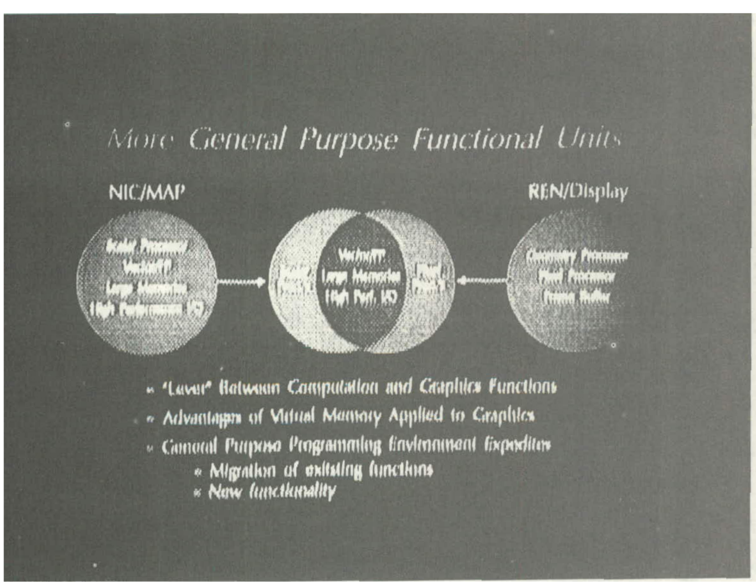
•Tom Jermoluk

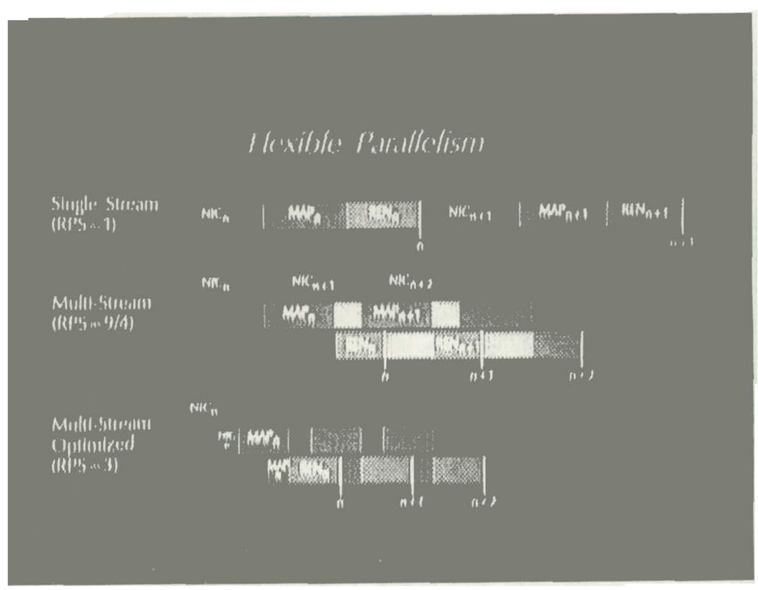

SIGGRAPH '88
Hardware Strategies
for Scientific Visualization
Dr. Franklin H. Moss
Vice President, Software Development
Stellar Computer Inc.
Stellar

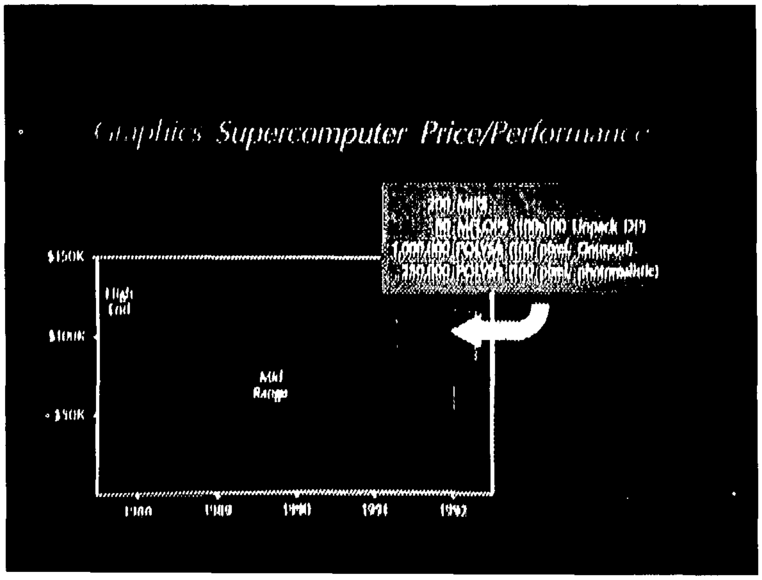

Visualization Applications and Performance Strategies
NIC
Raw Data
MAP
Primitives
REN
Images
Display
USER
Display n - 1
Display n
NIC n
MAP n
REN n
Time
Realization Step
max(RPS)
$/D
Over a broad range of applications,
employing different mixes of
NIC, MAP and REN


Hardware Strategies to max (RPS)
Ultra-High Bandwidth Integrated Systems
Trend to More General Purpose Functional Units
Flexible Parallelism
Stellar


Ultra-High Bandwidth Integrated Systems
Supercomputer
Supercomputer
NIC
MAP
REN
Display
USER
Graphics Workstation
NIC
MAP
REN
Display
USER
Graphics Supercomputer


More General Purpose Functional Units
NIC/MAP
REN/Display
• "Lever" Between Computation and Graphics Functions
• Advantages of Virtual Memory Applied to Graphics
• General Purpose Programming Environment Expedites
• Migration of existing functions
• New functionality


Flexible Parallelism
Single Stream
(RPS = 1)
Multi-Stream
(RPS = 9/4)
Multi-Stream
Optimized
(RPS = 3)



Summary

- MIPS, MFLOPS, MPOLYS not the bottom line
- Unique architectural features required to max (RPS)
- Analagous software advances required

Stellar