PROVIDING HIGH PERFORMANCE VISUAL SIMULATION AT LOW COST, REVISITED

Allen E. Snow
Richard N. Moon
Evans & Sutherland Computer Corporation
Salt Lake City, Utah

ABSTRACT

In the process of transitioning from the theoretical to the actual, things often change greatly. This is also the case in the move from system design, to system implementation and finally the actual performance characterization. In a follow-up to last year's paper "Providing High Performance Visual Simulation At Low Cost," which described the system design and architecture, this paper will discuss what has been learned in the actual implementation and use of this system.

An evaluation of the low cost approach is presented, delineating the system capabilities and limitations under various training requirements. One aspect that will be discussed at length is the channelized system architecture and how this architecture responds to different types of visual imaging simulation including sensor channels, narrow fields of view, and combinations of imaging channels, all used in the same system.

INTRODUCTION

Last year, we reported on the design of a new visual system, known as SPX, which had been developed with the primary emphasis on providing high performance in the basic areas of visual simulation at the lowest possible cost.[1] That paper discussed some of the thinking which went into the design of the system, which capabilities were most important, and the challenge to implement them efficiently. We discussed how some capabilities, especially the "bells and whistles," were by-passed for the sake of cost.

At that time, the engineering prototype was just being debugged and only the first simple pictures had been observed. A year has now gone by, the hardware has been fully debugged, the software is in place, several models have been constructed, and we have observed from actual experience how the system will perform. We have been able to test the system capabilities against the requirements for several military and commercial applications, each with their special challenges. We would now like to report on some of the things we have learned and how the system measures up.

OVERVIEW OF SYSTEM ARCHITECTURE

First it would be well to review briefly the system architecture. Figure 1 shows a block diagram of the system. In this system, the general-purpose computer and its associated peripherals were replaced with a single-board multiprocessor card and a 5.25-inch Winchester disk. The Multiprocessor card feeds and controls up to eight channel processors through a common bus. Each channel processor is composed of three major sections. The Object Manager (OM) is a very small section but plays a major roll in the efficiency of the system. The Object Manager performs the data base management function by considering the entire data base structure and presenting to the Geometric Processor only those scene elements which are potentially visible, near the eyepoint, and contribute most significantly to the scene. The Geometric Processor (GP) performs the graphical computations such as translation, rotation, clipping and perspective division. The GP also performs many special effects such as VASI control; texture motion; and flashing, rotating and strobe lights. The Display Processor (DP) is the largest section. It processes the polygons, complete with antialiasing, texture, fog and landing light effects, into the raster structure and drives the display devices. The DP can drive calligraphic display devices in a hybrid raster and calligraphic mode, or it can drive a raster scan device.

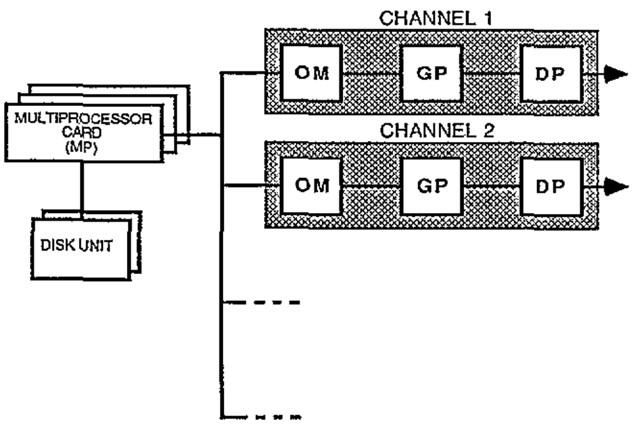

Figure 1: SPX System Block Diagram

One of the unique features of the SPX architecture is that, except for the common Multiprocessor card, the system hardware is totally channelized. Not only is there an independent GP for each channel, but the data base management section (OM) has been distributed across the channels. This was a new and innovative concept, and one which presented the most unknowns. We will devote a major portion of this paper to our findings relative to the channelized OM.

Another major feature of the system architecture is its *modularity*. Each section is capable of being configured through several increments from low capacity and capabilities to high capacity and capabilities.

FLEXIBILITY

One of the major goals of the SPX system was flexibility. The product was designed to cover a wide range of configurations and applications. During the short time we have had the opportunity to test the system against many user requirements, we have been impressed with its flexibility.

The modeling, diagnostics and real-time software have been designed to be as general as possible, so that the same software package can be used for all configurations, and the user is not unnecessarily burdened with the details of his particular system configuration. The real-time system software, for example, makes use of a Site Configuration File, a special user-definable data file, which specifies his special user parameters. In fact, nearly every program parameter which can be adjusted is accessible through the Site Configuration File.

Not only is outward system configuration flexible, but the actual programming of hardware by development engineers is surprisingly easy. Most hardware algorithms are microprogrammable. Of particular note are the Geometric Processor and Object Manager sections. The Arithmetic Processor (AP) card, which makes up the basic computing element in each of these sections, contains a writeable control store which is loaded during system initialization. During the system debug phase, many changes, adjustments, and enhancements were made to both the GP and OM sections without a single change in hardware implementation. As an example, the clipping algorithm was improved after our initial debug — a change which resulted in a 10% savings in processing time. Also a new feature allowing pseudorandom placement of point lights and a random modulation of the brightness of individual lights in straight light strings was added. Figure 2 shows an example of the realistic effects produced by randomly placed lights in the background and random intensity lights on the runway.

Another example of system programmability was manifest in the early stages of GP debug. Several Arithmetic Processor cards operate in parallel to provide total system throughput. The original design implemented a double-buffered memory at the output of each Arithmetic Processor card. Early tests showed that double buffering was inadequate, and AP cards which processed objects with many polygons interfered with AP cards which processed only a few polygons. Initially, object size was limited to twenty polygons in order to minimize this interference, but modelers were hampered by this limitation. With some fairly simple microcode changes, the double buffers were changed to FIFO (First In First Out) buffers, object sizes were increased, and AP card interference was eliminated.

Total GP improvements in microcode resulted in a 25% reduction in the number of AP cards originally estimated.

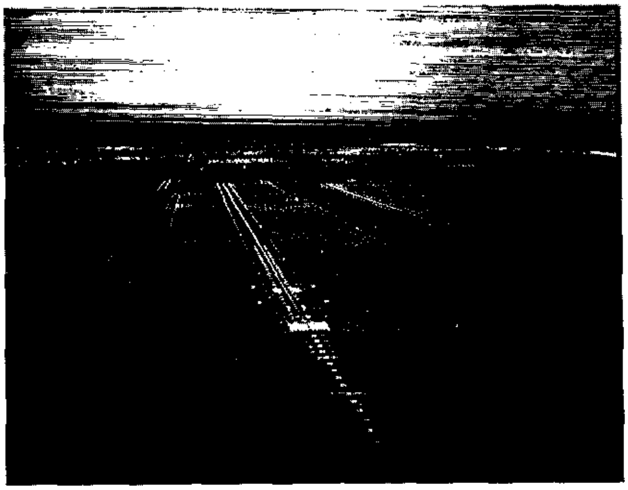

Figure 2: Random Placement and Intensity of Light Points

CHANNELIZED OBJECT MANAGEMENT

As mentioned in the system overview, except for the Multiprocessor Unit, SPX has a completely channelized architecture. This is of special interest in the OM section. In general, the Object Manager's task is to traverse the hierarchical data base tree structure, performing priority processing, level-of-detail calculations, and field-of-view testing at each data base node. For an OM common to the system, the field-of-view test at each node would be accomplished multiple times, once for each channel in the system. The data base traversal process continues down each branch of the tree until a node is not contained in any of the channel views. Once a node is reached which is out of the field of view of all channels, the OM stops further processing on that branch. In a system where each channel has its own OM, each OM begins traversing the same data base, independently performing the same calculations as a common system OM but doing only one field of view test at each node. Eventually each channelized OM traverses a smaller portion of the data base tree. However, there is some amount of overhead since the channels duplicate effort in the top nodes of the tree. As can be seen in Figure 3, this overhead results from the parts of the data base tree that all three channels must traverse.

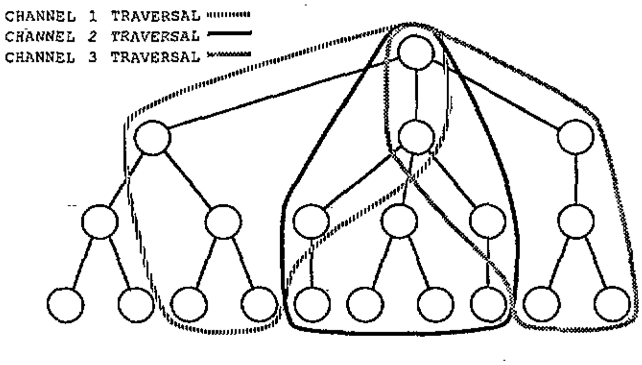

Figure 3: Tree Traversal of Channelized OMs

To determine the amount of overhead which might be expected, we performed an experiment. Using 52 degrees as a typical channel field of view, we widened the field of view of a single channel of the SPX system to three times 52, or 156 degrees, to simulate a single system OM processing data for a three channel system. Comparing this simulated single system OM with the three channelized OMs in a three channel system, it was determined that each of the three channels processed about 50% of the total nodes verses 33% as would be expected in the ideal case. This does not mean, however, that each channelized OM processed 50% of the system load. In our single-system OM simulation, the field-of-view testing was performed only once at each node instead of three times — once for each of the three channels. If this factor is accounted for in our simulated system OM, then each of the three channelized OMs would perform about 39% of the processing load of the single system OM. Even though the channelized OMs traverse less of the data base tree than the single system OM, all three channelized OMs process some common parts of the data base tree. This accounts for the 17% increase over the ideal 33% of the system load processed by each channelized OM. This overhead was actually smaller than what we expected and certainly justifies the channelized architecture. Had we designed a common OM, it would have required an additional backpanel per system. Also it would have had to have been very modular in order to grow with channel additions. Both of these requirements would probably have added more than 17% to the cost.

ADAPTABILITY TO SPECIAL APPLICATIONS

In the last section, we discussed the performance of the distributed Object Management hardware. Here we will discuss some of the other exciting possibilities which result from having independent Object Management sections in each channel.

Often there are requirements for a visual system to have the various channels of a visual system compute the image from multiple eyepoint positions. Multiple eyepoints might be required to meet many different training needs. For example, two or more observers might view a scene from different places in a simulation device where it is important to provide the correct scene from each observer's perspective. Sensor devices are most often placed on an aircraft at a point displaced several feet from the pilot's eye, and the sensor scene should differ from the visual scene appropriately. Other simulator configurations desire to share image generator capabilities between more than one cockpit.

The SPX image generator, with its channelized OM sections, is particularly well suited for handling multiple eyepoint requirements. As the OM traces the data base tree structure, it forms translation vectors, accumulates composite rotational matrices, makes level-of-detail decisions, and forms priority relationships based on separating planes. Many of these computations are dependent on the eyepoint position. Since each channel has its own OM, the tree can be traced uniquely in each channel for an independent eye position. If the OM were common to all channels, and a single traversal of the tree were associated with all channels, multiple eyepoint positions would cause a serious problem. In this case, in order to handle multiple eyepoints, multiple passes would need to be made through the tree and system capacity would be reduced.

We have found it convenient to divide the multiple eyepoint requirements into two categories. The first category we have termed "offset eyepoint," where the eyepoints are separated by only a few feet. With small separations, the active data base (the data base which has been scrolled off disk and resides in environment memory) can be common to both eyepoints. Offset eyepoint capability essentially comes "for free" in SPX, since each channel will determine priority and proper scene perspective independently.

The second category we call "independent eyepoint". This refers to widely separated eyepoints where the active data base might be entirely unique for each eyepoint. The base-line SPX software and microcode perform the data base retrieval from disk based on criteria common to all channels. The OM of channel 0 dedicates a small portion of its time spread over several fields to determine which data base elements are required to be in environment memory. Since this function is performed in only one of the channels, we can see that independent eyepoints require a slightly different approach. However, the architecture is such that a few simple changes would allow for independent eyepoint operation.

Another application which has presented a new challenge to image generators is that of sensor simulation. Often there is a need for both out-the-window visual scenes plus sensor images of some portions of the same scene. The most cost effective solution would be for a single image generator to provide both types of scenes using a common data base. One channel might be required to produce monochrome IR or LLTV images while the others provide full color visual scenes. In SPX, not only can color maps be loaded independently to each channel processor, but the model can contain two color descriptions for each polygon. Then since the channels operate independently, designated Geometric Processors can select the alternate color descriptions when directed to do so.

Most often, sensor channels require fields of view much smaller than the visual channels. This is generally no problem as far as scene perspective is concerned, since the viewport description is contained in a simple rotational matrix. However, level-of-detail management presents an interesting problem. Objects which can be seen at great distances through the sensor channels should appear at their highest level of detail. For the visual channels, such distant objects should be displayed only at the lower level of detail. If the object manager were common to all channels, it would have difficulty doing both things at the same time. SPX, with its channel-independent object managers, has no difficulty adjusting the level-of-detail transition ranges appropriately for channels with widely varying fields of view.

FADE LEVEL OF DETAIL

The initial design of the SPX system allowed for the changing of the detail of objects based on their range from the eyepoint. This changing of the detail of objects creates scenes that have objects in high detail when they are close to the eyepoint, objects of lower detail that are farther away from the eye, and objects that are not displayed at all when they are very far away. For complex models, many detailed representations may be used. Transition between any two of the various representations of an object were designed to be discrete as specified by a transition range described in the model. This abrupt change from one representation to another proved to be very distracting, especially when the change occurred close to the eyepoint. Unfortunately, moving the transition ranges far enough away from the eyepoint to remove the distraction increases the polygon count of the system. With limitations on polygon count, the scene density has to be reduced to keep the polygon count within limits and yet the transition ranges need to be far enough away to avoid the distraction of representation changes.

We had expected these results and had considered implementing fade level of detail. However, we decided not to implement it because the estimated cost was too high. Once the system capabilities were better understood, we reconsidered implementing fade level of detail using the built-in transparency capability. Fade level of detail allows for the increase of scene density by shortening the transition ranges on the objects and allowing for more complex objects close to the eyepoint. This fade level of detail was implemented by using a percentage of the transition range to fade between the object representations. In this fade region, two representations of the object are displayed at the appropriate transparency level. At the far edge of the fade region, the lower detailed object is displayed with a low transparency level and the higher detailed object is displayed with a high transparency level. As the eyepoint gets closer to the object, the transparency level of the lower detailed object increases and the transparency level of the higher detailed object decreases until the lower detailed object completely disappears and the higher detailed object is completely opaque. As the OM traverses the data base tree, it uses a range calculation to determine which representations of an object to send to the GP along with the transparency level of the representations.

A small test was conducted to verify the effectiveness of fade level of detail in the SPX system and to evaluate the additional load which might be encountered by the various sections of the channel processors. Two tests were made, one used fade level of detail and the other did not. A test model was created using a house with two representations (one more complex for close viewing and one less complex for distant viewing), and the house was replicated at equal distances to create a data base with 4096 houses in a square of 64 houses on each side. The more complex representation (high level of detail) of the house contained between 13 and 15 viewable polygons (depending on the viewing angle) and the less complex representation (low level of detail) of the house contained three viewable polygons.

The first test was done using fade level of detail. In order to keep as close to 500 viewable polygons in each channel at any one time, the houses were placed 550 feet apart. The transition from null to the low level of detail was placed at 8000 feet and the transition from the low level of detail to the high level of detail was placed at 3500 feet. Using these distances, the OM took about nine milliseconds to process each field of data. In moving through the data base, the level-of-detail transitions were smooth, with little distraction.

The second test was made using a similar data base but without fading during level-of-detail transitions. Without fade level of detail and with the same transition distances used in the first test, the transitions between the levels were very distracting. In order to reduce the distraction to the level observed in the first test, the transition ranges had to be increased to 6000 feet for the change between the low level to the high, and 12000 feet for the change between the null and the low level. With this increase in transition ranges, the polygon count went up. To return to the 500 viewable polygon count, the houses had to be placed 875 feet apart. OM processing time dropped to 7.25 milliseconds per field.

Comparing these results, we experienced a 250% increase in scene density with only a 24% increase in processing time using fade level of detail. Figure 4 shows the increased density with fade level of detail. Figure 5 shows the scene without fade level of detail.

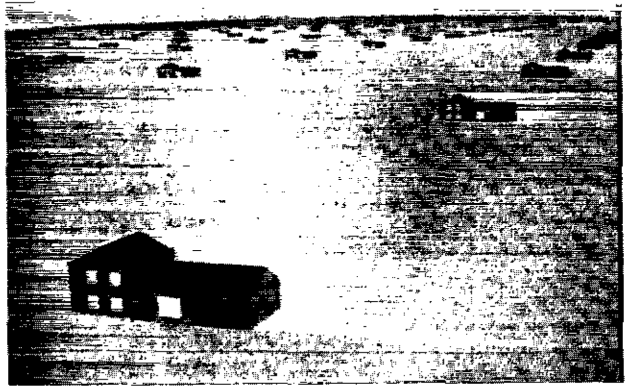

Figure 4: Test Model with Fade Level of Detail

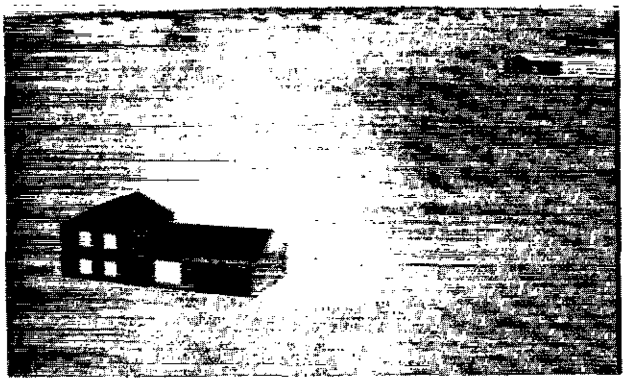

Figure 5: Test Model Without Fade Level of Detail

Another consideration in comparing the models is the DP processing time. The use of transparency in the houses to implement fade level of detail increases the DP frame time because those pixels affected by the transparency must be processed more than once. The processing time was found to increase by 3.5% for fade level of detail. The GP processing time stayed approximately the same since the number of polygons was held constant during both tests.

Scene density can be increased even further if more than two representations of the house are used, each being a small change from its neighbor's representation. This way the transition ranges can be even shorter. The tradeoff is a slight increase in processing time in the OM with more objects being output to the GP and more processing time in the DP as more transparency is utilized in the scene.

TEXTURE, THE OPTION WHICH CANNOT BE IGNORED

As texture becomes more and more common in visual systems, and as modelers develop more skill in its application, it becomes an essential part of realistic simulation. No longer can it be considered an option.

A conservative approach was taken in the addition of texture to SPX. The plan called for simple intensity modulation texture to be applied to polygons of any orientation. Eight on-line maps were included in the design, and up to two maps could be combined on any given polygon. As a last moment addition to the texture prototype design, color modulation and transparency modulation capability were added. Color modulation texture allows the texture values to modulate between two adjacent colors in the color table. For transparency modulation texture, the texture value modifies the transparency value of the polygon. These two new texture features have been met with a great deal of enthusiasm. As can be seen from the ship's wake in Figure 6, transparent texture can add irregular patterns which help immensely in adding realism. Color modulated texture provides white clouds on blue sky, brown and green fields on the ground, and many other realistic effects.

In going to the production configuration, larger memories were added to the texture map cards, providing 32 on-line patterns. As a further capability, texture patterns can be scrolled off disk to replace patterns which have become inactive. Thus, an unlimited number of patterns can be made available.

There is an additional point of interest relative to the use of texture. Due to cost considerations, smooth shading and color blending were not implemented in SPX. We have found that in many cases where fixed shading and color blending have historically been used, the proper application of texture can produce nearly identical results.

DISPLAY PROCESSOR PERFORMANCE

The performance of the Display Processors has either met or exceeded our expectations in all areas. Image quality is exceptional. Edge antialiasing equals the performance standard set by Evans & Sutherland's top-of-the-line CT6 product. And of course, calligraphic light points out-perform raster lights by at least an order of magnitude. With clear, crisp pictures, about the only element of distraction is that of field tracking. Much has been done to reduce the effects of this problem, but the only way to eliminate field tracking completely is to draw the scene using non-interlaced raster — currently a step beyond the cost constraints of SPX.

We have been able to meet the pixel resolution goals which we established for the various configurations. In fact, we have been operating our in-house systems at about 11% higher resolution than anticipated. We have also been encouraged by the system's capacity to process transparent polygons and transparency modulated texture. The display area covered by transparencies does have limits; however, the modelers find that most requirements for the use of transparencies fall well within these limits during normal viewing scenarios.

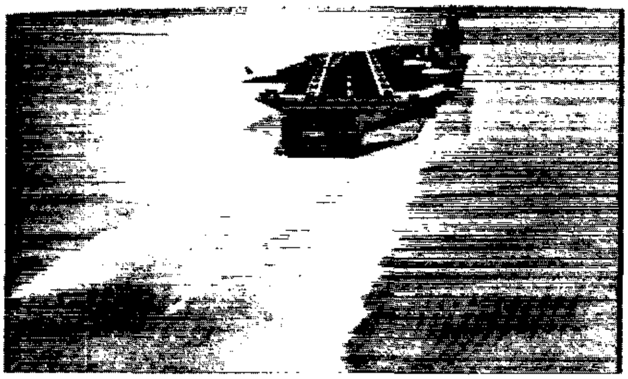

Figure 6: Transparency Texture For Ship's Wake

CONCLUSION

After a year of actual operation, the SPX system has been found to meet or exceed design expectation in almost all areas. Where problems were discovered, the flexibility and programmability of the system allowed rapid and simple solutions to those problems. Distributing the Object Manager across the channels has not only proven to be effective, but has offered a wealth of configuration possibilities. Evaluations of the system against published commercial and military requirements have shown it to have the correct mix of capabilities with a price more in line with increasingly stringent budgets.

REFERENCE

1. Moon, Richard N., "Providing High Performance Visual Simulation At Low Cost," Proceedings from the 7th IITEC Conference, November 1985.

ABOUT THE AUTHORS

Mr. Allen E. Snow has been with Evans & Sutherland for the past 6 years. He has been involved with Field Service and more recently in development of hardware and software for the NOVOVIEW visual systems. He currently develops hardware, software, and IC designs for the SPX visual system. Mr. Snow received his BS in Electrical Engineering from Brigham Young University at Provo, Utah.

Mr. Richard N. Moon has been with Evans & Sutherland for the past 12 years. He has been actively involved in the development of both hardware and software for the NOVOVIEW visual systems. He currently manages the SPX development effort. Mr. Moon received his MS and BS in Electrical Engineering from Brigham Young University at Provo, Utah.