ENHANCING THE COMPUTER GENERATED ILLUSION

Neal L. Mayer and Michael A. Cosman
Evans & Sutherland Computer Corporation
Salt Lake City, Utah

ABSTRACT

The usefulness of the image produced by a CIG system is not well characterized by just the number of edges or surfaces displayed, but is a strong function of the effectiveness with which the scene details provide visual cues. This paper presents some examples of what can be achieved using hardware capabilities and modeling techniques to enhance the ability of the CIG to present useful scene detail.

Previous CIG systems changed scene details when their image size was small enough so as not to be distracting to the observer. Recently introduced system capabilities allow scene details to evolve in a more continuous, smooth, and independent manner. By using these capabilities, details need not be included in the scene until they are of visual importance. A hierarchical management structure is utilized to provide efficient data base culling and level of detail control. This enables data bases with thousands of square miles, many levels of detail, and thousands of surfaces per square mile to be processed efficiently by the image generator. Such data bases, rich in two- and three-dimensional textural features, can be efficiently produced by using automated generation procedures that require a minimum of modeler effort.

INTRODUCTION

The imagery produced by a real-time computer image generator (CIG) is, in truth, a very sophisticated illusion. The participant in a simulation exercise should feel, think, and react as if what he is seeing is real, when in fact it is not. As the range of training, engineering, and test environments broadens and the demand on training effectiveness increases, so does the demand on the CIG.

The problem for CIG systems has always been to produce the "most effective image" possible within a set of constraints. The typical measure used to define the "most effective image" has been the number of edges or surfaces that can be *displayed*. This has proven to be an inadequate metric in specifying the capability of a CIG system. The effectiveness of the illusion is more dependent on the system's ability to provide sufficient cues for a particular task, whether it be high altitude navigation or low altitude weapons delivery. This ability can be measured in terms of the number of displayed edges or surfaces, the number and density of cues, the concentration of cues where they are most important, and the system's ability to manage the data base in a manner that is both non-distracting to the observer and efficient in presenting scene detail.

To enhance the computer-generated illusion, an effective combination of modeling strategies and image generator capabilities is required. The development of the techniques presented in this paper was done on an Evans & Sutherland CT5A image generator. This paper will introduce some of the system capabilities, and the modeling strategies which have been developed to more fully utilize these capabilities.

HARDWARE FEATURES

This section will outline the structure of a CT5A data base and then discuss some of the features of the CIG's cell processor, which does the initial processing of the data base.

Data Base Structure

A CT5A data base is a collection of *scene elements that has* been organized into a hierarchical structure to allow efficient processing by the image generator. Scene elements consist of planar surfaces, called polygons, and lights. Objects in a data base are made up of a collection of these surfaces and lights. Usually an object is thought of as a single entity such as a tree, bush, or rock. By organizing surfaces of various sizes and shapes the modeler can create complex items, such as aircraft and ships. The modeler can determine the visual attributes of an object by specifying the color and shading characteristics of the surfaces that comprise the object. CT5A also allows objects to be made transparent, and several levels of transparency are provided.

The hierarchical structure of the

data base can be likened to a tree with main branches that diverge into smaller branches and eventually down to the individual leaves. The CT5A image generator traces through the data base tree and applies a series of culling tests to determine which portions of the entire data base should be considered for further processing and possible display. These culling tests are performed to eliminate, as early as possible, and in pieces as large as possible, those portions of the data base that will not appear in any display channel. These culling tests include level of detail (only the simplest representation of an object necessary for a given viewing distance is used) and field of view testing (does the item impinge on any display channel). An example of this data base tree tracing will illustrate this concept.

Imagine a data base which consists of the entire continental United States. If the simulated eyepoint is positioned so as to be looking directly at the Evans & Sutherland building in Salt Lake City, Utah, then the image generator must trace through the data base tree for the whole United States and decide which surfaces, out of the billions possible, should be displayed to represent this scene. At the beginning of the tree processing the entire continental United States is represented by what is known as a cell. A cell is a volume in 3 space which has a high and a low level of detail (LOD), i.e., a complex and a simple representation. Each representation may either be an object, which is output directly to the geometric processing function for eventual display, or a collection of cells. This collection of cells, called a mesh, forms another branch of the data base tree which is further processed by the image generator. Each cell also has an associated transition range, which tells the image generator which LOD to process, depending on the eyepoint's distance from that cell.

Since the simulated eyepoint is within the cell for the United States, the high LOD option is pursued and all processing on the low LOD stops. The high LOD for this United States' cell may consist of a mesh of 48 cells, where each cell represents a state. The culling tests are applied to these 48 cells to determine that only the cell for the state of Utah is within the specified transition range and needs further processing, while the other 47 states need no longer be considered. The cell for Utah also has a low and a high level of detail with an associated transition range. The high LOD is processed further by the image generator since the eyepoint is within the specified transition range. This high LOD is made up of several counties in the state, with each county being a cell. Again the culling tests on these county cells are applied to find that only Salt Lake County needs further processing.

Each of the cities in Salt Lake County may be considered a cell with its own high and low level of detail. Since the eyepoint is in Salt Lake City, the high level of detail for the city is processed further while the lower levels of detail for some of the nearby cities may also be processed. The Salt Lake City cell may have several subcells for sections of the city, and again the culling tests would choose the East Bench area for processing at its high LOD, while other portions of the city might have their lower levels of detail represented depending on their distance from the eyepoint. Tracing through the data base tree in this manner continues until the particular scene that is being observed is culled out of the rest of the features in the East Bench portion of the city.

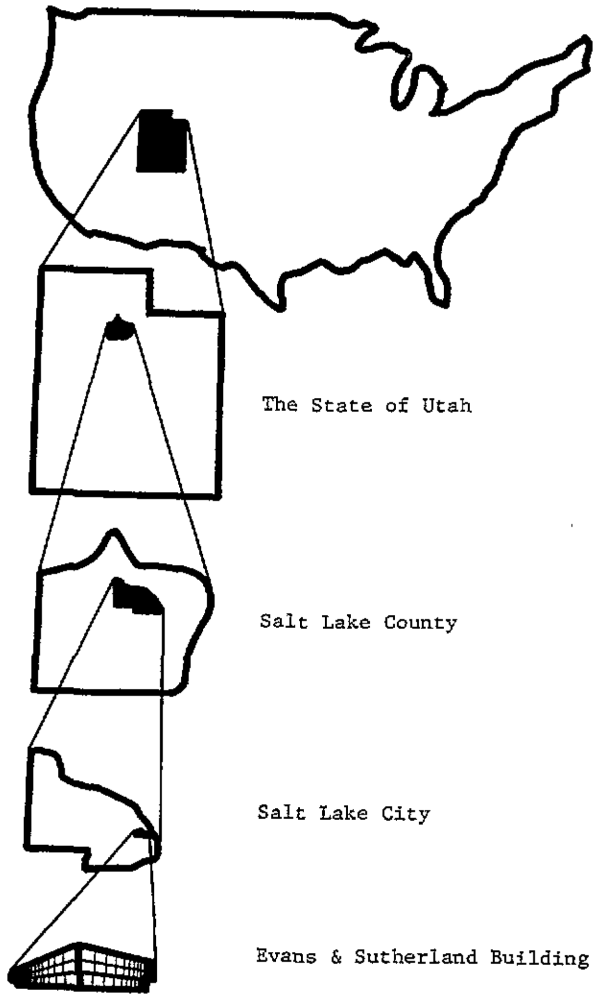

Figure 1
A Hierarchical Data Base

In this manner the image generator has, with very few decisions, culled through the entire United States data base to determine exactly which surfaces need to be processed to display the desired building and nearby details. This type of hierarchical structure allows the CT5A hardware to trace through an extremely complex data base tree very rapidly and efficiently.

The Cell Processor

The cell processor is that portion of the image generator which traces through the data base tree, once every field, to determine which set of objects should be considered for further processing. Several features in the cell processor enhance its ability to perform this function.

Perspective Foreshortening Correction. When a primarily flat scene detail is viewed at a very shallow glancing angle, its effective screen subtense may be much less than its nominal perspective size based on range alone. This perspective foreshortening means that the displayed visual significance of the item will be generally less, sometimes much less, than if it were always viewed straight-on. CT5A provides a method of accounting for the perspective foreshortening of items by correcting the computed transition range to delay introduction of an object into the active scene to provide for equivalent visual impact at the switch. This process helps avoid the compression of details at the horizon, and their attendant buildup, and further frees IG resources for use on more visually significant scene details.

Fade Level Of Detail. Fade level of detail makes the transition between levels of detail smoother and less distracting to the observer. When this is used, the transition is done gradually. Using the transparency feature, the incoming object starts out transparent and becomes more opaque, while the outgoing object starts out opaque and becomes more transparent. This gradual switching between levels of detail allows the transition to take place much closer to the observer without being distracting.

Load Management. The CT5A employs a hardware-based load management scheme which takes advantage of the fact that small modifications in LOD transition ranges throughout a data base can have a considerable effect in altering the image generator's system load, without having a significant effect on the visual scene. The transition ranges on each cell in the data base are modified to keep the image generator running at or below the nominal time. Modifications to the transition ranges are done in real time and are small enough to reduce the system load when necessary without becoming visually apparent. Since the load management is an automatic process, the modeler does not need to be concerned about creating a data base that will never overload the system, he only needs to have a data base whose average complexity is within the system's limits.

MODELING TECHNIQUES

The characterization of what constitutes a useful image has always been a very difficult task. Many considerations must be taken into account which are highly dependent on the training task, the image generator's capabilities, and the subjective biases of those using the system. There are, however, several objective measures that can aid in determining the usefulness of a data base.

1. How many scene elements can be displayed in each channel, and in the aggregate channels? This number, which is hardware-dependent, has typically been the measure of a CIG's capability, yet important issues arise about how those scene elements are used to provide cues.

2. What is the density of scene elements or objects in the modeled environment? The density of objects in the scene indicates the frequency of cues and determines whether sufficient information can be presented to perform a particular task. Training tasks, such as nap of the earth helicopter flying, may require some type of ground object every 25 feet, while medium altitude flying might require one every 2000 feet. Some tactical missions require appropriate cues throughout a large altitude range, implying a capability to make a graceful transition to additional detail at lower altitudes without exceeding the system's capacity at the higher altitudes. The user of a simulator must determine the density of cues needed for the task and then specify an image generator and modeled data base which can support that density.

3. What is the average density of scene elements or objects in the image plane? Because of scene compression near the horizon, a data base that has a uniform distribution of objects in model space will not be uniformly distributed in the image plane, but will tend to have increased density with distance from the eye. This tends to "crowd" the objects in a very narrow band near the horizon, at the expense of sparser detail closer to the observer. Methods that can create a nearly constant density of objects in the image plane will allow much more visually effective use of scene detail.

4. Are objects provided at various sizes so that cues are available at several ranges from the eye? The eye has a tendency to ignore visual cues from objects which are either too large or too small. Whatever the actual size of an object, there is a *range* of distances in which the object will have the proper perspective size to be of visual significance, and outside of which the object will be increasingly ignored. This implies that the data base must provide cues which are of various sizes so that those objects, which are larger and spaced less frequently, are mixed in with small items which are spaced closer together. This provides a fairly constant spatial frequency of cues for the participant in a visual simulation.

Several modeling techniques have been developed to increase the effective scene content of a data base based on the criteria set forth above. These methods can provide ways of analytically designing and then producing a data base which will satisfy a broad range of training tasks.

Level Of Detail

The modeler uses a level of detail strategy to create several representations of an object, each with a different number of scene elements. For example, an airplane might have three levels of detail, i.e., three representations, one with 440 surfaces for the highest detail, another with 140 surfaces, and the lowest detail may only be 32 surfaces. The image generator chooses which representation to display based on the distance from the eyepoint to the airplane. In creating the data base, the modeler specifies that the airplane will first appear as the lowest level of detail at 3.25 nm., then be replaced by the middle level of detail at 0.5 nm., and the highest detail representation will be shown at 0.1 nm.

The modeler tries to specify transitions to take place as close as possible without making the replacement distracting, and thus saves the image generator as much processing as possible. The distance at which the transition takes place is based on such criteria as: 1) the perspective size of the object, 2) the relative contrast between the object and its background, 3) the shape of the object's silhouette, and 4) the relative geometric differences between the levels of detail which are switching. By using simpler representations for objects at greater distances, the image generator has greater capacity to display more objects, thus increasing the number of visual cues that can be displayed.

The level of detail strategy applies not only to individual objects but also to every cell in the data base. Each cell in the data base has a low and a high option pointer which can be considered the low and high levels of detail for that cell. This allows an entire region to have a low and a high *detail representation, as well as each* portion of the region. The real power of the strategy comes from the hierarchical nature of the cell. By creating a hierarchy of cells, the modeler can create a tree with a high and low level of detail, then a group of trees with a high and low level, then a forest with a high and low option, *and even an entire* large section of forested terrain with both options.

An example of this can be seen in Figure 6. Represented in the desert scene are six levels of detail. The first level of detail consists of the basic desert floor with a few mesas. Nearer to the eye, buttes are added to form the second level of detail and complete the large objects. The third level of detail is formed by adding sand dunes around the mesas and buttes. These dunes are added to the scene at a distance which makes their silhouette visually significant. The fourth level of detail is to add the cacti on and around the dunes and buttes. The fifth level of detail adds the bushes to the scene. Rocks and pebbles make up the sixth level. This six level-of-detail design of the desert applies the level-of-detail ideas to the desert as a whole, as well as to each palette item in the desert, and makes the scene very detailed near the eye but fairly simple far away where the small items are not important.

Fade Level Of Detail

With the hardware fade level-of-detail capability, the modeler has greater leverage in specifying the transitions between representations of an object. Previous strategies for level of detail switching required a sudden replacement of one representation for another. This required the switch to take place when the perspective sizes of the objects, or their relative differences, were sufficiently small so as not to be distracting to the observer. This was typically at a much greater distance than that at which the resulting details were useful. Thus a significant amount of detail had to be processed, not because it was useful but solely to help keep its introduction from becoming objectionable.

If fade level of detail has been selected by the modeler for the transition between an object's two levels of detail (LOD), then at the specified transition range the replacement LOD is brought into the active scene as a transparent object. As the distance from the eyepoint to the object changes the replacement LOD gradually becomes more opaque while the replaced LOD becomes more transparent. When the replacement LOD is fully opaque the replaced LOD is completely transparent and is removed from the active scene.

Fade level of detail makes the changes between objects more subtle and consequently less distracting to the observer, and thus allows the transitions to take place closer to the observer. The subtlety also enables the use of fewer levels of detail, with greater geometric differences between levels, yet still without distraction at transition. Fifty percent reductions in the transition range are not unusual. This means that simpler representations are more often displayed, which conserves system resources and allows more objects to be displayed.

Level Of Scale

A visually effective data base must provide a selection of visual cues of differing sizes and spacing. Each particular visual cue has a range over which it contributes significantly to the image; beyond this range its decreasing size makes it less and less valuable, and the modeling strategy should remove it from the scene when its value drops below some threshold. To extend the scene richness beyond this range, other visual cues are needed which are substantially larger (hence continue to have visual significance at the further ranges), and are spaced farther apart. Several additional leves of scale may be provided, each with a particular range over which it contributes significance, until a continuity of detail to the maximum required range is achieved. The largest levels of scale, which are introduced into the scene at the longest ranges, are not removed from the scene as the observer gets close to them, but their greater spacing keeps their contributions from overloading the image generator. Each level of scale can be designed to require a percentage of the overall image generator resource, and then several levels added together to achieve the required depth of scene without image generator overolad. The desert shown in Figure 6 uses six such levels of scale, and achieves scene richness from very near the observer out to ten miles. The apparent scene density remains high over a wide range of flight altitudes, and the emergence of the smaller levels of scale provides valuable velocity, altitude and time-to-impact cues.

Computational Methods

Planning and executing the design of a data base to accomplish specific objectives within given image generator constraints has traditionally been a difficult and iterative task. The next few sections of this paper indicate how such planning and estimating can be successfully accomplished during the design phase, and how the data base structure makes it feasible to modularize the modeling process. In performing these computations, several approximations are made concerning the data base and the methods used in deriving the results. The data bases used in these examples are assumed to be sufficiently flat that the surface area of the terrain can be approximated by a plane and planar area formulas can be used. When the area under consideration extends from the eyepoint out to a particular distance, the area will be computed as a sector of a circle. If the area under consideration is bounded by two nonzero distances from the eyepoint, then the area will be computed as a sector of an annulus. The average ground area per object is computed as the total area covered by the objects divided by the number of objects in that area. The average spacing between objects is simply the square root of the average ground area per object.

The hierarchical nature of the data base tree allows the modeler to construct the data base in modules which can then be concatenated to form the whole data base. The impact of each module on the image generator's capacity can be computed independently from the other modules in the data base. As the modules are joined to form larger modules, the result on system load is additive. This allows the modeler to start with the specifications for a data base and systematically budget the image generator's resources among various types of detail to produce the desired data base.

The arithmetic used in deriving the results of the next few sections is simple and straightforward. Sufficient information is provided so the reader can verify the numbers if desired.

Scene Element Densities In Model Space

Computing scene element densities gives an indication of the number of visual cues presented in a scene as well as their average spacing. If the data base is assumed to lie essentially in the ground plane, then scene element density means the number of scene elements per square unit when the highest level of detail for objects in the area is presented, e.g., surfaces per square nautical mile (sqnm). This is usually computed as:

$$\text{Density} = \frac{\text{No. of objects}}{\text{sqnm}} * \frac{\text{No. of surfaces}}{\text{object}}$$

Because of the limited capability of an image generator to display scene elements, the allowable density of objects in a data base is constrained by the total number of scene elements which can be processed and the simulator's field of view. In the following examples the visual system is assumed to have a 120-degree horizontal field of view.

The above relationship gives the modeler four parameters which can be adjusted to meet the needs of the simulation problem: number of objects, the area which is covered by the highest level of detail of these objects, the number of surfaces per object, and the density. Any three of them necessarily define the fourth.

Typically a modeler will be faced with a problem where one or more of the parameters will be specified for him and he must create the data base to fulfill those specifications. A problem a modeler might encounter is to create a forest which uses an average of 1500 surfaces for the trees within the field of view. The next task the modeler would do is to design a tree or several types of trees which he would like to put in the forest. The modeler can make a tree which uses anywhere from several hundred surfaces to a simple three-sided pyramid in its highest LOD; the choice is based on the visual requirements for the simulation exercise. Some situations, such as nap of the earth helicopter flying, may require very detailed trees, while other applications for higher altitudes may be satisfied by simpler representations. The specifications for the data base should indicate the types of visual cues required. Suppose that in this example the forest will be flown over at a high velocity and only fairly simple representations for the trees are needed to give height and velocity cues. Based on this criteria the modeler creates a tree which has 10 surfaces. After some experimentation the modeler finds that if the tree is added to the active scene at a range of 5 nm., then the addition of the tree is not distracting to the observer.

The modeler has now specified all of the parameters in the density relationship. The area in a 120-degree field of view sector of a 5 nm radius circle is 26.2 sqnm with 1500 surfaces desired. This implies a density of 57 surfaces per sqnm or 6 trees per sqnm. This density means that the average ground area per tree is 6.44 million sqft. or an average spacing of 2,500 feet between trees. The modeler can then create his data base so that the placement of trees in the forest is an average of every 2,500 ft.

By using fade level of detail, the transition range at which the tree is added to the active scene can be decreased considerably. If in the above example the transition range were to be decreased to 1.5 nm., then the resultant forest would have an average surface density of 637 surfaces per sqnm. or 64 trees per sqnm. with the average spacing between trees being 762 ft. The tremendous leverage that is gained by decreasing the transition range is due to the fact that the total area displayed at a given LOD decreases proportionately to the square of the transition range.

Using multiple levels of detail also gives the modeler tremendous leverage, since representations requiring fewer surfaces can be used for the sections of the field of view which may cover large areas. Suppose the modeler finds that he can create a lower level of detail for the tree which uses only 3 surfaces. The modeler also changes the transition ranges so the low level of detail comes in at 1.5 nm. and the high level of detail comes in at 0.25 nm. Using these parameters, the modeler computes that his forest can now have a surface density of 1,992 surfaces per sqnm. or 199 trees per sqnm. with an average spacing between trees of 430 ft.

Table 1 summarizes the results of these three examples and shows the tremendous increases in surface densities realized when using multiple levels of detail and the fade level of detail option.

Tree Configuration	Average Spacing between trees (ft)	Polygon Density (surfaces /sqnm)	Tree Density (trees /sqnm)
1 level of detail, 5 nm. transition	2,500	57	6
1 level of detail 1.5 nm transition	762	637	64
2 levels of detail, 1.5 and 0.25 nm transitions	430	1,992	199

Table 1
Forest Tree Surface Densities

Image Plane Scene Element Densities

If the density of objects in model space is constant, as in the examples above, the density of objects in the image plane will increase with the distance from the eye. This is simply because the area in a given section of the image plane increases with that section's distance from the observer. This tends to concentrate the objects in a scene close to the horizon and leave fewer objects close to the eye. To efficiently use the displayed objects for visual cues, the observer would like to have the density of objects remain constant in the image plane so that the number of objects close to the horizon is nearly the same as it is close to the eye. This implies that the density of objects in model space must decrease rapidly with distance from the eye.

When computing the image plane density the modeler needs to know the vertical field of view configuration for a simulator. Figure 2 shows a possible configuration.

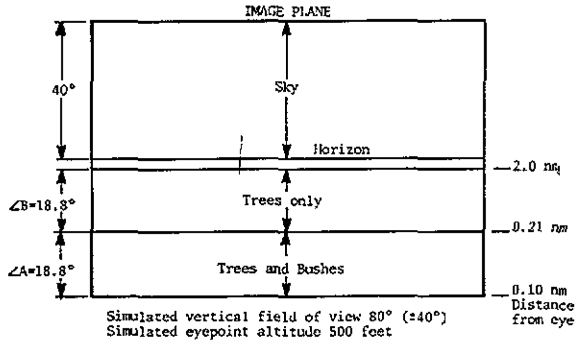

Figure 2
Vertical Field Of View Configuration

To achieve near constant image plane density, the modeler can employ some of the level of scale concepts and add smaller objects close to the eye at a greater density to provide more visual cues. Continuing with the example above of the forest, the modeler may wish to add bushes to the foreground half of the image plane section covered by trees, i.e., the section covered by angle A in Figure 2. Suppose that the modeler has created a bush which has a transition range to be brought into the active scene of 0.21 nm. The modeler then needs to compute the necessary density of bushes to have the same image plane density in the area with trees and bushes as in the area with trees only.

The density of trees in the 120-degree horizontal field of view was previously computed to be 199 trees per sqnm. The area covered by the annulus defined by angle B in Figure 2 is 4.1 sqnm., hence 825 trees are in that section of the field of view. The region defined by angle A in Figure 2 has an area of 0.04 sqnm. To achieve the same density of objects in the section of the image plane covered by angle A, there must also be 825 objects. In the region covered by angle A there are presently 7 trees, which means that the modeler needs to add 818 bushes to achieve the desired object density in the image plane. 818 bushes spread out over the area covered by angle A would be achieved by placing a bush on the average of every 40 ft.

With the addition of the bushes to the foreground, the modeler has achieved either a bush or a tree approximately every 40 ft. as compared to the 430 ft. with only the trees. If the modeler uses 4 surfaces for the bush, then he will have achieved a surface density in the forest of approximately 93,673 surfaces per sqnm.

The 93,673 surface density represents the total number of surfaces which comprise all of the bushes and trees when their highest level of detail is presented. Since the trees have a lower level of detail and not all the surfaces in the bushes and trees are front-faced, only about 4000 surfaces are actually displayed.

Another technique for achieving a near constant image plane density is to have the same type of objects brought in to the active scene at varying distances. In the forest example the modeler may choose not to use bushes, but rather have the trees brought into the scene at several transition ranges. He may have trees coming in at 2 nm., 1 nm., and 0.5 nm., each with increasing densities in model space as the distance from the eye decreases.

By trying to achieve a constant image plane density, the modeler can create a data base which has an abundance of three-dimensional cues near the eye where they are useful instead of concentrating them on the horizon. These examples, although simplified, show the basic principles. In practice the modeler can calculate the densities of several sections of the screen and use several different types of objects to achieve the desired densities.

Automated Generation Procedures

In the creation of high density data bases, the modeler is faced with the problem of how to create the large number of objects necessary. In the above example of the forest with bushes, if the modeler were to create each individual bush and tree in a 50 nm. x 50 nm. square of forest he would have to make about 57 million bushes and 500,000 trees. The need for automated creation of objects is clearly evident. It is also evident that the image generator must be able to reuse portions of the data base since it cannot hope to store the number of surfaces required for each individual tree and bush.

The modeling system used to create CT5A models has a procedural capability, which gives the modeler tremendous leverage in creating large numbers of similar objects. These procedures are somewhat similar to FORTRAN subroutines which have input parameters and particular outputs. A procedure which makes a tree can have input parameters of position, scale, rotation, and transition ranges between levels of detail. Once the procedure is written, it can be called many times with variations to the input parameters to create a large group of trees. After this group of trees has been made, it can be duplicated and used in many locations to create a whole forest.

The creation of terrain over large geographical areas is greatly aided by the use of procedures. Procedures can be written to create mountainous areas, hills, farm land, forests, swamps, oceans, and the like. If the modeler is constrained to create a section of terrain to a certain level of specificity he may do so and then continue the enhancement of the terrain by the use of procedures. A modeler may specify where the major peaks, valleys, and ridges are in a mountainous section and then let the procedures embellish the basic structure with smaller hills, minor peaks, and vegetation.

Procedures defined in the modeling system may, of course, call other procedures. This capability gives even more leverage to the modeler. A procedure may be written to do a runway by calling other procedures to do VASI's, runway lights, strobes, stripes, numbers, oil marks, tire marks, and expansion joint cracks. In this manner a modeler may build an entire runway in only a few minutes by specifying such things as runway heading, length, and type.

The large number of surfaces, which must be stored and processed in a high density data base, also requires the image generator to be able to reuse portions of the data base. The CT5A has an instancing capability which allows portions of a data base to be stored in memory once, but used many times by relocation within the data base. The modeler uses this instancing capability by creating a mesh for a particular portion of the data base and then indicating that copies of this mesh should be placed at various locations in the visual environment. The original mesh is placed in the image generator's visual environment memory and then all copies access the same information from memory and apply a modeler-specified translation to position the mesh where desired.

When creating the forest, the modeler may create a group of 575 bushes and 5 trees as a mesh. Using the instancing capability, the modeler may then place this group of bushes and trees at various locations to create portions of the forest or perhaps the entire forest. This small section of forest could be used in the data base as needed. In this manner the modeler only has to model a small section of the data base to get effectively large sections, while freeing the image generator's memory to hold more information concerning other portions of the visual environment.

EXAMPLES

This section shows several examples of data bases which have been created to demonstrate the modeling strategies discussed above. These data bases are all designed to run on a six-channel CT5A with a 120-degree horizontal field of view and to stay within the system's nominal capacity.

Figure 3: Hilly forest -- This hilly forest employs a single seven-surface object for the trees, using fade level of detail for the transition from null to a single tree. The trees are brought into the scene at distances ranging from 1.5 nm. to 0.5 nm. to maintain a constant image plane density. No level of scale concepts are employed, hence the lack of cues for higher altitudes or for below tree top flying. The spacing between the trees is approximately 191 ft., creating a surface density of approximately 7,000 surfaces per sqnm.

Figure 4: Flat Forest -- The flat forest employs two levels of detail for the trees and employs fade level of detail for the transitions. Constant image plane density is achieved by varying the distance at which the trees are brought into the scene. The trees and the abstract ground patterns create two levels of scale and enable flight below tree top level. The surface density achieved here is approximately 130,000 surfaces per sqnm. with a spacing between the trees of approximately 50 feet.

Figure 5: Ocean and Clouds -- Textural features like ocean waves and clouds are possible using several levels of detail and several levels of scale on the color patterns. The procedures used in creating the ocean and clouds are similar and required the modeler to specify only 28 surfaces for the patterns. The ocean and clouds have been designed to use only 34% of the system channel capacity, thus reserving sufficient capacity for ships and aircraft.

Figure 6: Desert -- The desert represents the most thorough application of the modeling techniques. Six levels of scale are used to create a nearly constant image plane density and to support flight at a broad range of altitudes. Each item in the desert is

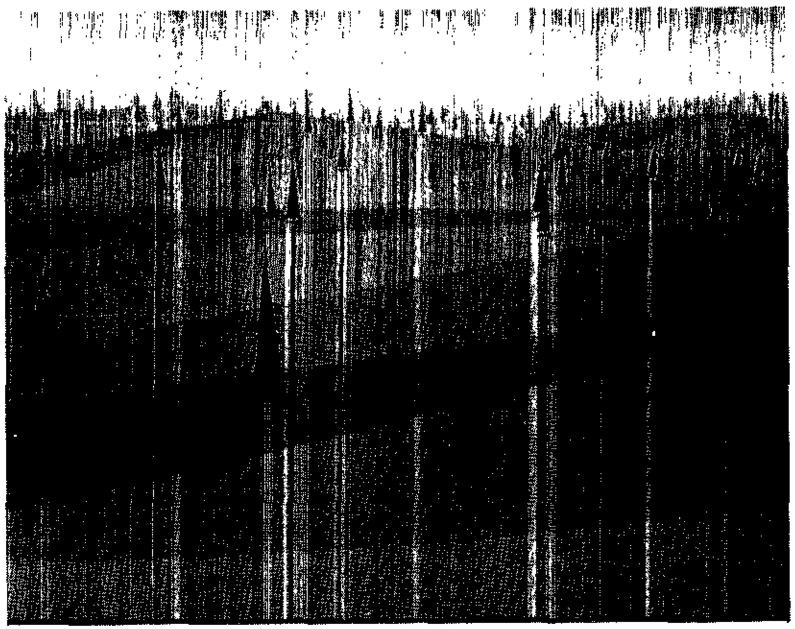

Figure 3 Hilly Forest

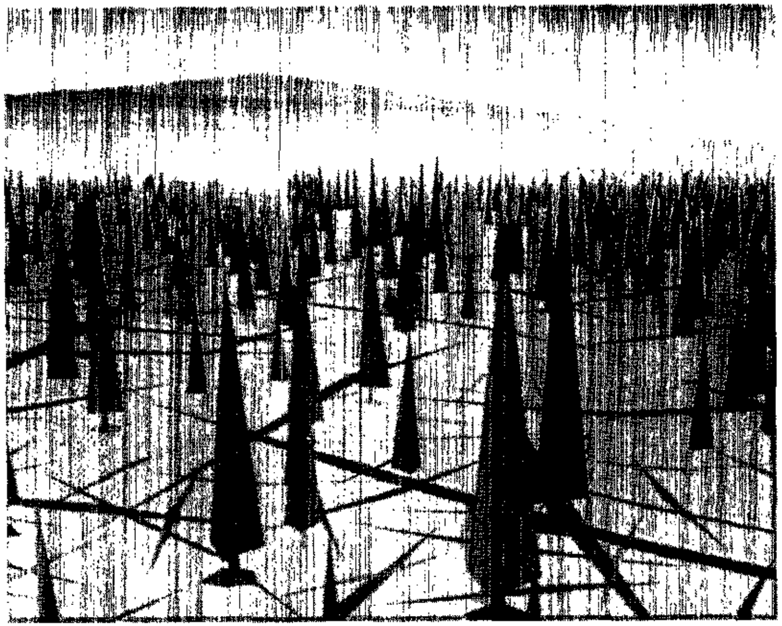

Figure 4 Flat Forest

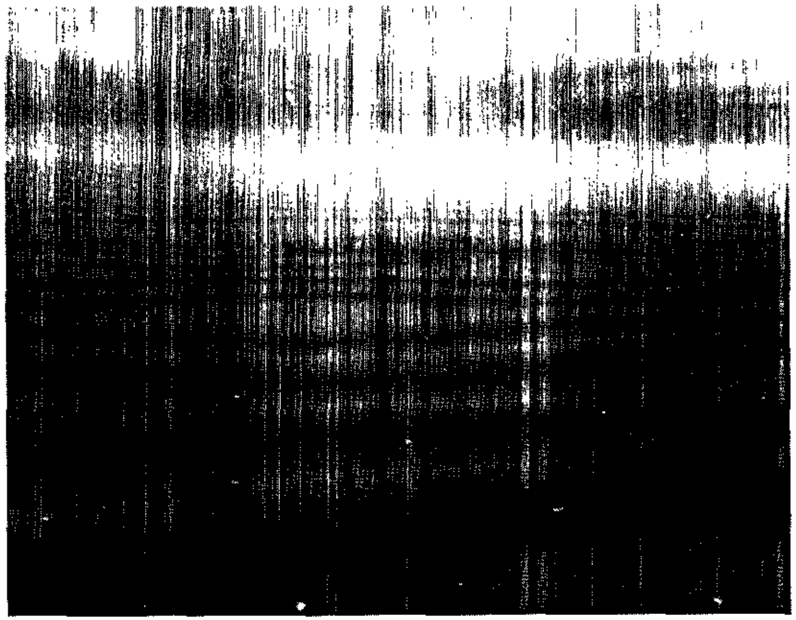

Figure 5 Ocean and Clouds

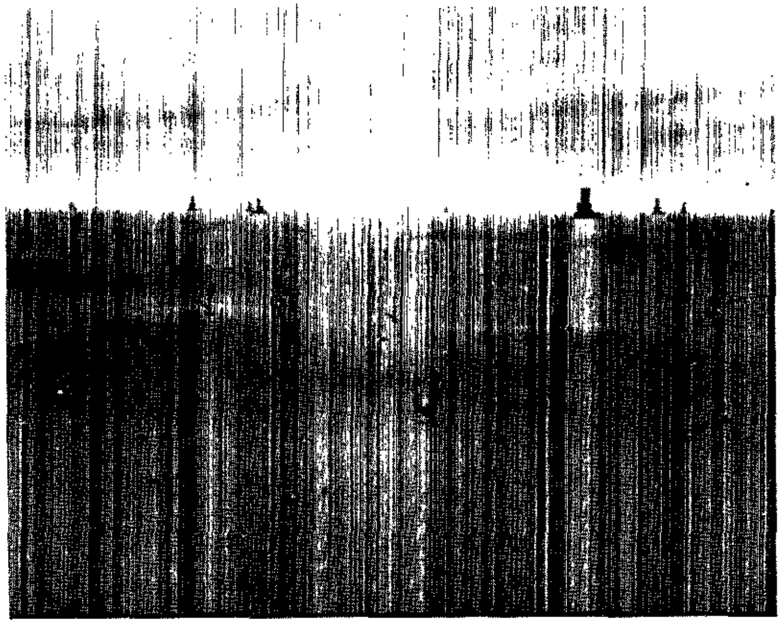

Figure 6 Desert

modeled at 2 to 4 levels of detail, using fade level of detail for the transitions. The surface density achieved is approximately 87,000 surfaces per sqnm. over 100 sqnm. and yet, using the procedural capabilities, the modeler only had to model approximately 400 surfaces. Using the hardware instancing, the entire desert requires only 4000 surfaces in the image generator's memory.

CONCLUSIONS

An effective combination of the CT5A hardware capabilities and modeling strategies enables the modeler to create and display data bases which greatly increase the usefulness of the CIG image. The image generator's ability to process large amounts of information, and make a multitude of decisions concerning such things as level of detail, load management, perspective foreshortening correction, and data base culling in real time, enables the modeling strategies to be employed.

It is our belief that by applying these modeling techniques, along with the necessary hardware support, visual data bases can be created which will support nap of the earth, contour flying, vertical takeoff and landing, and a host of other applications.

REFERENCES

1. Cosman, M. and R. Schumacker, "System Strategies to Optimize CIG Image Content." *Proceedings of the IMAGE II* Conference, June 1981.

2. Schumacker, Robert A., "A New Visual System Architecture." Proceedings from the 2nd. IITEC Conference, November 1980.

ABOUT THE AUTHORS

Mr. Neal L. Mayer has been involved in the development of modeling techniques and applications for CIG visual systems at Evans & Sutherland for the past one and a half years. Mr. Mayer holds a Masters degree in Mathematics from Brigham Young University.

Mr. Michael A. Cosman has provided technical support in the development of CIG visual systems at Evans & Sutherland for over 8 years, working primarily on image quality and system architecture issues. Mr. Cosman holds a Bachelors degree from Brigham Young University.