SPECIFYING LARGE CIG DATA BASES

John Robinson
Michael Cosman
Evans and Sutherland Computer Corporation
Salt Lake City, Utah

ABSTRACT

Computer image generators have reached a level of maturity in terms of the image realism they can achieve. The recent addition of advanced hardware texture to surfaces has provided unsurpassed realism, allowing the creation of much more effective and diversified visual cues. As a result, CIG systems can now be applied to a broader class of training missions.

Because the scene realism is now adequate for many training missions, the CIG data base content problem is becoming increasingly important. New training tasks such as mission rehearsal and nap-of-earth flight require data bases with an increasing geographic extent and density of three-dimensional objects. To meet the new requirements, many data base management and compression schemes are being devised and new hardware capabilities will be created to support them. The problem for the user then becomes one of determining which of these approaches meets his needs.

This paper will discuss and show through example the need for data base management and compression. In addition, it will discuss the trade-offs that occur because of compression schemes. Last and most important, *this paper will suggest several methods of specifying data base requirements that take into account* such parameters as geographic extent, specific content, density and correlation. The goal of the specifications is to allow the user to more readily understand the data base processing performance of a computer image generator and to reduce the confusion caused by the various management and compression schemes.

INTRODUCTION

The visual fidelity of scenes created by advanced computer image generators has recently reached a level of maturity that allows them to be used in a wide variety of training situations. Widespread use of texture and effective use of level-of-detail strategies have provided tremendous increases in the visual cues necessary for pilot training. More importantly, the scene complexity has improved to the point that real world areas can at least be recognized, if not exactly correlated.

But this scene fidelity has put more pressure on another problem in computer image generation, the problem of data base size. This comes about for two reasons. First, a greater density of objects in a given geographical area can be displayed. This is a result of the increased capacities of today's image generators and improved level-of-detail techniques that allow a greater number of objects to be displayed for the given capacity of the image generator. Second, many of the training missions require larger geographical areas to be modeled.

As a result, data bases are growing faster than the mass storage devices needed to contain them. New methods of managing and compressing the data are being used and will be developed to meet the new training requirements. Compression techniques can reduce not only the size or number of disks necessary for a given data base but can also reduce the complexity of creating, modifying, transferring and archiving it. While these techniques often impose some restrictions on the model, the ability to properly manage and compress the data base gives the designer the ability to overcome two of the physical constraints of the system, mass storage size and display capacity. This paper will not address any particular compression scheme but will attempt to enumerate the types of compression that will be used. In addition, it will discuss the tradeoffs that result from management and compression, suggest how these tradeoffs should affect the specification of the data base and suggest ways of evaluating the machines that run these schemes.

VISUAL ENVIRONMENT DATA STRUCTURES

The visual data base consists of two distinct types of data. The first is the scene elements which are the basic building blocks that the modeler uses to create the scene. The management data on the other hand does not directly affect the image being presented. It is used to structure the data base so that the scene content can be maximized while staying within the image generator's display capacity.

Scene Element Data

The image generator's major task is to create the proper perspective image of the visual environment for generally arbitrary and unconstrained viewing positions. To do this, the elements of the visual data base are described by a collection of points. A point is specified in three-dimensional Cartesian coordinates requiring an X, Y, and Z component relative to some global origin. For example, the X, Y, and Z components might specify how far the point is to the east, north and above the origin, respectively. The various scene elements require different numbers of points. A light requires only one point. A polygon (sometimes referred to as a face) uses a series of at least three points to define a bounded planar surface. (Other scene elements such as analytic surfaces and textured volumes not yet in wide-spread use require different types of geometric data which are similar in nature.) Accompanying the positional data is data which describes the other visual attributes of the polygon. These include color, shading information, opacity and texture with all of its associated attributes. Note that all of the data mentioned here directly affects the scene perceived by the viewer.

There is data associated with polygons and other scene elements which does not directly affect the final rendering. This data is used to provide efficient means to cull or remove from the processing list those elements which do not affect the current scene, thereby increasing the efficiency of the system. For example, polygons have a surface normal which indicates the visible side of the polygon so that polygons facing away from the viewer can be culled. Data indicating the size of the polygon is used to remove it when its perspective size gets too small to be visually significant. All types of scene elements might have bounding volumes which can be easily compared against the boundaries of the channel to effect a field-of-view test.

Management Data

To further increase the efficiency of the processing in the image generator and to more easily manage the scene content, it is necessary to dedicate a significant portion of the data base to data structures which relate the various scene elements topologically. A system of topological relationships that has been shown to be quite efficient is the hierarchical structure described in Reference 1. The basis of this hierarchical structure is a data element called a cell. A cell represents a geographic section of the data base by referencing other cells, each of which represents a subsection of the data base. At the lowest level in this tree of cells is a cell that represents the entire data base, while at the highest level, the cells represent single data base items by referencing objects instead of more cells. An object is a collection of polygons which make up all or a portion of an item. In addition, each cell contains information that allow the referenced cells to be efficiently culled due to field of view, allows some cells to be selected over others based on level of detail, and provides a means to quickly sort the remaining cells into a visual priority order (roughly closest objects first) as required by the display processor. All of this management data can easily add 33% to the size of a visual data base.

To understand the complexity of a data base, consider the following example. A reasonably sized data base might need to cover 400 by 400 nautical miles. (This is a small data base in that a fighter jet could traverse it in less than an hour.) Such a data base might require items (trees, houses, road segments, etc.) to have an average density of 400 per square mile or an item placed every 300 feet. The number of items in the data base is 64 million. This item density is quite sparse compared to the real world but practical in terms of a computer image generator. If an average of 20 polygons per item is assumed, the data base would contain 1.3 billion polygons. Another significant portion of the data base is the terrain. The terrain is usually composed of triangular polygons that span between regularly spaced elevation grid posts. The horizontal grid post spacing in a typical data base is approximately 300 feet, which results in over 130 million triangular polygons. This brings the total number of polygons in the data base to 1.4 billion.

DATA BASE MANAGEMENT

The management data described above has a hierarchical structure to improve management efficiency. The reason for this is easy to show in terms of the example data base. Assume for the moment that only two levels of detail exists for the items, all and none, and that the image generator is capable of displaying 12,000 polygons or about 600 items. The items must be removed from the scene at about one mile from the viewpoint to insure that an overload will not occur. If no data base hierarchy exists, every polygon in the data base would have to be tested at least once per frame to see if it were within visible range. This would result in 131 billion tests to be performed in a frame time (about 33 ms.) which is well beyond the performance of a practical image generator. An obvious improvement is to test each item instead of each polygon which reduces the testing load to 65.5 million but this is still much too large a number.

The same test on a hierarchically structured data base is much less computationally intensive. Now assume that the data base hierarchy consisted of a four-to-one subdivision which means that at the bottom of the data base tree is a cell which references four cells, each of which contains four cells etc., until the highest level cells contain up to four objects. The number of levels of subdivision for the example data base would be 13. Starting at the top-level cell, each cell it references is tested to see if that cell is within a mile of the viewpoint. Typically only one of the top level cells will survive thus eliminating three-forths of the data base from further testing. The same will be true of the next several levels of subdivision until more than one cell is within range. If the analysis were carried out completely, it would show that less than 1,000 tests are required to find the right 600 items.

Besides managing the scene content, the data base manager must perform memory management tasks as well. A typical data base is large enough that it would be impractical to keep the entire data base in the image generator's fast memory. Consequently, image generators are configured with a system disk as shown in Figure 1. which contains the entire data base. A high-speed interface connects

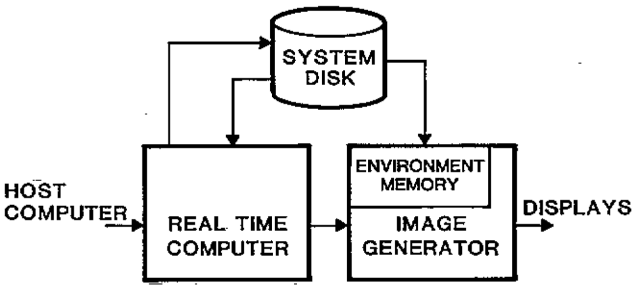

Figure 1

the disk to the environment memory inside the image generator. The environment memory only needs to contain that portion of the data base that immediately surrounds the eye. In a process called paging, portions of the data base that the vehicle is moving away from are discarded, while the portions that it is moving towards are read from the disk.

The criterion that the data base manager uses for determining what should be in memory has a strong influence on the system's performance. The simplest criterion is to pull an item into memory when it is within a certain range of the eyepoint. The maximum range at which an item has to be paged in is the visibility range. A more efficient criterion is to consider the range at which each level of detail of the item is needed. In this case, only the current and lower levels of detail of each item are kept in the environment memory.

As a result of the level-of-detail criteria, the density of scene elements held in the environment memory diminishes as a function of range from the eyepoint. Ideally, the density should fall off as $1/r^2$ to keep the total number of displayed scene elements constant as a function of range as shown in Figure 2a. The total number scene elements held in the environment memory is the volume under the surface created by rotating this curve about the eyepoint.

A highly flexible memory management scheme is necessary to handle the variety of simulation tasks that exist. For instance, in a nap-of-earth or ground vehicle simulation where all of the visual cues are close to the pilot, the visibility range could be kept quite short, possibly two miles. This allows the density of items to be quite high because the image generator does not have to hold or display distant polygons. A more detailed description of such a system can be found in reference 2. Such a system has a scene density profile shown in Figure 2b.

At the other end of the spectrum are simulations requiring magnified images such as infrared cockpit displays or targeting device simulators. Because distant objects still subtend a large portion of the screen, the high level-of-detail models of those items must be kept in memory. This does not increase the number of scene primitives to be displayed because the field of view is small, but does drastically increase the number of scene elements to be stored in the environment memory. The result is the flatter density profile shown in Figure 2c.

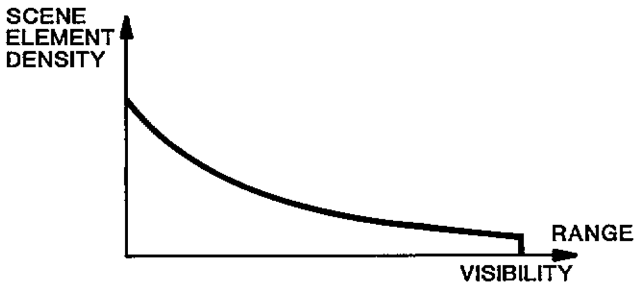

Figure 2a

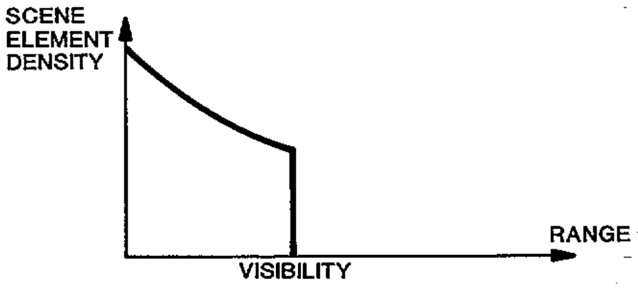

Figure 2b

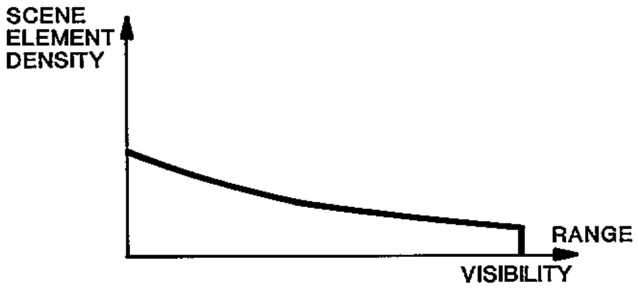

Figure 2c

DATA BASE COMPRESSION

In the example data base cited above, it was determined that 1.4 billion polygons were needed to fill the requirements. If no attempt is made to compress this data, the number of polygons makes this data base impractical. A single polygon with texture and culling information takes on the order of 100 bytes of information to describe it. This would require about 25 to 50 magnetic disk drives to store just the polygon data. There are many schemes for compressing a data base and for the most part different portions of the data base demand different methods of compression. The following is a description of several types of data base compression.

Implied Data Compression is a method where part of the information is derived from the arrangement of data in the data base. An example would be the case where a complex surface such as terrain needs to be represented. The image generator needs to represent the terrain as polygons, each vertex being a three-component vector. But if the terrain can be described as a regular array of Z values, similar to the way terrain data is described in DMA (Defense Mapping Agency) data bases, then only the Z information needs to be stored on the disk. The X and Y components of the vertices can be derived from the location of the Z values in the array. In addition, because all of the terrain polygons at a given level of detail are about the same size and aligned with the X,Y data base axes, much of the information necessary for texture, culling and sorting can be derived as the image is being drawn instead of storing it on the disk. As a result the 131 million polygons needed for the terrain can be represented on the disk with 65.5 million Z values. If a Z value is described with four bytes, (compared to a polygon description of 100 bytes), the compression ratio achieved for terrain is 50 to 1.

Resolution compression reduces the amount of data by reducing the numerical resolution of each of the data elements describing an object. An image generator must typically place objects with a resolution of a fraction of an inch but some parts of the data base may not require this kind of resolution. Again terrain is an example of a good use of resolution compression. Quite often the elevation data need only be accurate to within a foot of the real elevation. If this technique is applied to the Z values described above, the number of bytes can be reduced from four to two, improving the compression ratio by a factor of two.

A method of data base compression commonly used now is instancing. Instancing reduces the data storage requirement by keeping only one description of an item and allowing that item to be replicated as many times as necessary. In its simplest form, instancing generates exact copies of the item at each position specified by the modeler. More sophisticated instancing would allow other attributes of the object to be specified such as color or size. Instancing is a very powerful technique that can be used for something as simple as runway stripes to modeling strategies involving the entire data base. One such strategy is described in reference 2.

One way to apply instancing to the example data base is to create a catalog of items. One-thousand items would provide a fairly rich variety and take about two megabytes to describe. Each instance of an item requires information about which item is being instanced, the position of the item and any other attributes that might be modifiable. This could be done in as little as 20 bytes, resulting in a compression ratio of about 100 to one (plus the overhead of the item catalog).

Procedural modeling is a technique where items are built or placed by an algorithmic description rather than an exact description. A simple example would be a case where a wheel needs to be generated out of polygons. A modeler would normally enter a description of how to build a wheel into the modeling system and, in an off-line process, the modeling system generates a complete set of polygon and point data for the wheel. Instead of storing the polygon data, a data base could hold the description of how to build the wheel which is more compact. A processor within the image generator expands the description into polygons for display. In addition, the on-line procedure could adjust the number of generated polygons according to the wheel's current level of detail.

As can be seen from the examples above, each compression method works best with a particular kind of data in the data base. Implied data compression and resolution compression work best with the terrain skin, while instancing and procedural modeling work best with the data base items. These compression techniques will ultimately work best in various combinations. Applying three of these techniques to the example data base yielded an overall compression of about 100 to 1. Procedural modeling is a largely unexplored technique but this, in combination with the other techniques, particularly procedural instancing, promises similar improvements in compression performance.

THE SPECIFICATION

The challenge of good data base design is to create a model that best meets the requirements of the user while staying within the limits imposed by the image-generating system. Data base modeling and data base compression provide the designer with another set of tools to meet this challenge. Data base management gives the designer the ability to make the most effective use of the image generator's display capacity, while compression techniques allow for a more effective use of the available disk storage and environment memory.

Data base compression and, to a degree, data base management will impose constraints on the final model. For this reason, it is best that the user understand these constraints in relation to his training needs and convey those needs in the specification. The following makes a number of recommendations of how to reflect the issues of compression and management into the specification but the user must decide how important each of these specification items is to the training mission.

Size

The geographic extent of a data base obviously affects the amount of data needed to describe it. The portions of the data base that describe terrain and instances of items grow linearly with the area of the model. Another aspect of increased extent is that larger data bases will cover a wider variety of terrain. If a data base must include forested areas as well as desert areas for instance, then different types of items must be included in the data base.

Terrain

The most relevant specification of terrain is elevation accuracy. Specifying terrain elevation accuracy puts limits on how large the spacing is between elevation values and the accuracy of those elevations. This specification must be made relative to the accuracy of the source of elevation data (such as DMA) since it is impractical to expect the modeling team to survey the actual site to verify the elevations. If the image generator uses a terrain sample spacing similar to that of the source data, little information is lost due to the sample spacing. The only issue then is how accurately the elevation data is stored.

Level of detail must be applied to terrain to keep the number of on-screen polygons to a manageable size. From a visual standpoint, the above specification only has meaning within a small area around the current position of the vehicle. Lower levels of detail of the terrain will cause increased error in the absolute elevation accuracy. The farther the terrain is from the eye, the greater the error in elevation. For low level and nap-of-earth flight, it may be necessary to know how accurately distant terrain is displayed. This is particularly true of threat avoidance training where the pilot must keep the terrain between himself and a threat. A way to measure this is to specify the peak visual angular displacement of the terrain. This is the angle between the terrain as displayed at its instantaneous level of detail and where it would be displayed if forced to its highest level of detail. This angle is most easily measured at the horizon as shown in Figure 3. This number is valid between the eyepoint and the maximum visibility range. If the terrain is modeled to meet the ideal scene element density curve

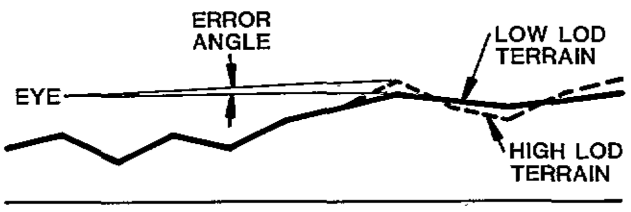

Figure 3

discussed earlier, this angle should be relatively constant out to the visibility range.

Items

When considering items that decorate the terrain in the context of data base compression, the issue of specificity becomes more important. Specificity refers to how accurately the modeled item represents the real item. For instance, it might be necessary to model a number of warehouses at various places in the data base. If it is not necessary to have the exact warehouse, it is preferable to specify instances of a generic warehouse because it requires less data. But if some of those warehouses must be recognizable as the actual warehouses, then a new model must be created which will increase the size of the data base.

Another aspect of specificity is the position of items in the data base. In the case of a forest, the exact position of each tree is probably not important. If the trees in a forest are automatically placed, thus reducing the size of the data base, the modeler loses control over the exact position. Again, if some specific trees need to be placed, more data is required. Specificity is a measure of the correlatability of a data base to the real world and a more specific data base results in a larger data base size.

A data base specification should include an average item density stated in items per unit area. This should include percentages of both specific and generic items. The percentage of each item type (house, tree, etc.) is important as well since each type of item requires different numbers of scene elements. For certain kinds of training tasks, a peak specific item density should be indicated as well as an average density. When training for a specific navigational task, it is sometimes useful to model a localized area or corridor with an increased density of specific items while decreasing it in others, thus keeping the total number of specific items constant. While this does not add any more data to the disk, the environment memory must hold more data when the eyepoint is in the area of high specificity.

Specifying the data base capacity in terms of items instead of scene elements is preferable in several ways. First of all, it is easier for non-modelers to understand what the resulting scene density will be like. It also leaves the modelers the flexibility to adjust item complexity to stay within the bounds of the display capacity. Item density specification also indirectly specifies data base compression in that higher item densities require more efficient compression techniques or more disk storage. The third effect comes about because the items must be displayed as well as stored. To stay within the image generators display capacity while avoiding visual anomalies, an efficient LOD strategy is necessary.

Scene Quality

A disadvantage of the above item density specification is that a data base can be contrived which only uses two polygons per item for the high level of detail. This will yield a very high item density but will not look realistic. Another way to fool this specification is to put the transitions to lower levels of detail too close to the eye. A specification could be put on the number of polygons in each item type but this will put unnecessary restrictions on the creative modeler who can make a more realistic looking item with fewer polygons. For that reason, the appearance of the items over the entire range of levels of detail must be subjectively evaluated when determining the acceptability of a system.

Another thing to consider is the variability of the generic items. A forested area will look unrealistic if all of the trees use the same generic tree type with no variation in size or color. The same could be said of housing developments or cities. Again, to put a strict specification on the variability of generic items may constrain clever modeling. Therefore, it is necessary to evaluate these areas of the data base for their visual quality.

Memory Management Capabilities

The data base management process takes a finite amount of time to determine which parts of the data base should or should not be in the environment memory. In addition, there is the time required to access and read the data from the disk. If the response time of the data base manager is too slow for the speed that the vehicle is traveling, items which are normally introduced into the scene beyond a perceptible range may suddenly appear when they are closer to the eyepoint because they were not paged into memory in time. The ability of the system to do data base management can be summed up in the question of whether the system keeps the right data in the environment memory at the maximum speed of the vehicle being simulated. For an aircraft, it is useful to specify the maximum speed at various heights above the terrain. If the aircraft is above a certain altitude, the highest level of detail of the items does not need to be in the environment memory because they will not be displayed. This results in reduced paging latency which allows higher rates of speed.

CONCLUSION

This paper has shown the need for data base management and compression for models that run on advanced computer image generators. These issues will become increasingly important with future systems. It discussed several forms of compression, namely implicit data compression, resolution compression, instancing and procedural modeling, to show the forms that various compression techniques might take and then gave some examples about how these might be used on the example data base. Finally it discussed the impact of data base management and compression on the model and on the specification of that model.

Ultimately it is up to the user to determine the degree to which the issues involving compression and management affect his particular requirement. Being aware of these issues and reflecting this awareness in the specification will aid in the choice of the right system to purchase and in the success of the training simulator.

REFERENCES

1. Cosman, M. and R. Schumacker, "System Strategies to Optimize CIG Image Content", Proceedings from the 2nd IITEC Conference, November 1980.

2. Wales, Cary E. and Michael Cosman, "DMA and CIG: A Shotgun Weeding", Proceedings of the 5th IITEC Conference, November 1983.

ABOUT THE AUTHORS

Mr. John Robinson is currently a Hardware System Designer at Evans and Sutherland Computer Corporation. He is involved in the definition of future image generator products. He has been involved with the hardware design of several computer image generators over the past eleven years. He received a BSEE degree from Manhattan College in 1975.

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