PROCEDURE MODELS FOR GENERATING THREE-DIMENSIONAL TERRAIN\*

Robert Marshall, Rodger Wilson and Wayne Carlson

Computer Graphics Research Group The Ohio State University

## ABSTRACT

A method for generating arbitrary terrain models, including trees, bushes, mountains, and buildings, is described. Procedure models are used to combine fundamental data elements in the creation of unified objects comprising the terrain model. As an example, a procedure model to generate arbitrary trees of various species is implementation. These are the generation of the low level data elements, specification of input parameter requirements, and a brief explanation of the algorithmic structure. Terrain images rendered by this process are included, as are diagrams and illustrations explaining the procedure model organization. Comparisons with previous work are made.

CR CATEGORIES: 3.12, 3.21, 3.56, 3.57, 3.65, 3.70, 8.1, 8.2

KEY WORDS: procedure models, terrain model, data generation, flight simulation, computer graphics, interactive systems, computer animation

#### INTRODUCTION

Computer animation has developed to the stage where systems can handle the 3-D display of objects or scenes involving several hundred thousand faces (1). There is a need for systems that can generate more complex data in a fashion and format that minimizes processing time while maintaining data integrity. Such is the case with terrain models. A typical 3-D image of a tree, for example, might consist of hundreds of thousands or even millions of faces, and a scene involving trees along with hills, mountains, and other terrain elements involves even more data.

It seems that what is needed is not only a system capable of generating and displaying

This work is supported in part by National Science Foundation Grant No. MCS76-18659 and by U. S. Navy Contract No. N61339-80-C-0008.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

•1980 ACM 0-89791-021-4/80/0700-0154 \$00.75

millions of faces, but one in which the user can interactively manipulate the data in a fashion not unlike a set designer, painter, or architect. The user needs to be able to look at a scene and decide to make a tree fuller by adding new branches, or move a corn field, or add a mountain. The massive quantity of data involved in creating a landscape make this type of interactive art direction nearly impossible without the aid of a language which permits the user to alter a few parameters without changing the entire object or scene. The procedure model fits this need. Although the use of procedure models in computer graphics is not new, it has not been applied to solving the specific problem of generating 3-D terrain models of this complexity.

By definition, a procedure is a precise stepby-step method for effecting a solution to a problem. In the case of terrain models, it is a method by which data is organized so that storage, retrieval, and display is simplified and made faster. With the help of a procedure model it is possible for the user to generate a tree without individually placing all of the thousands of leaves, or generate a mountain range without placing each individual mountain. By changing a few parameters the user is able to display an entirely new tree or mountain, and this greatly enhances the terrain images which can be displayed.

Digitizing real world objects is one way to accurately describe scenes or objects for use in a computer graphics environment. The problems which arise when using this method significantly inhibit the user. For example, digitization can involve much time and effort (5). Also, the data base must be carefully organized in order to facilitate the manipulation of objects in the scene. A second method of generation and display would be to interactively generate and store each individual terrain element for retrieval and display. However, the effort a user must go through to create and position each individual object is far too prohibitive to make it a reasonable alternative in this case. As in digitization the storage of such a large quantity of data is also a difficult problem.

Our research efforts in computer graphics have produced techniques that may have important implications for future computer image generation systems. We have developed procedure model algorithms for data generation that automate the creation of environmental features such as trees and textured objects. It is now possible to gencrate and display scenes with realistic looking features that demonstrate increased image complexity. Instead of images representing several thousand edges, it is now possible to display images involving millions of edges.

Other groups have undertaken the general problem of terrain display, using methods other than procedure models. Dungan, Stenser, and Sutty (3) have produced terrain pictures by using a texture tile technique. This two-dimensional technique is capable of assigning a specific texture approximation to a surface within an image. Although this capability is desirable in many instances, there are certain limitations with texture tiles that lead one to seek other techniques for terrain generation. The fact that only relatively few levels of detail are available limits the number of applications suitable for this routine. In many cases a three-dimensional image is required (single trees or a small group of trees) to render a terrain image so that it is possible to not only fly over the terrain, but also to move through it.

Another method of simulating terrain uses source data from the Defense Mapping Agency (DMA). This again produces a two-dimensional image, but with the added enhancement of cloud or fog simulation (4). Using the DMAAC terrain file as the data source, an array of height values is accumulated and used as the basis for calculating a 2-D representation of terrain. Reflecting the organization of the data base, the result resembles a topographic map. Once again, however, the image produced by this method is only suitable for very specific instances. The very nature of this method renders it appropriate for use in exercises in which the observer flies over the terrain. Even though a ground level image can be displayed, it is not possible to move through the terrain. Both the texture tiles and DMA techniques are better suited for high level flyovers of terrain rather than ground terrain flythrough or the display of single terrain images.

### PROCEDURE MODELS

Procedure models are a method of creating 2-D and 3-D object representations. The method is used to create instances of objects from the class of objects the procedure model can generate. Procedure models compute the exact specification of an object guided by the parameters that serve as input to the model. Procedure models are also capable of sending high-level information about the object generated (e.g., the region the object occupies or a measure of the complexity of the object) to other procedures such as a data base management system. Procedure models can produce enormous savings in the data hase description of terrain, but at the expense of computation time. Instead of storing the representation of a forest, for example, a procedure model description can be used to generatc the complete representation as needed.

Procedure models can be incorporated in a comprehensive object generator language, where the language consists of several different procedure models which are called with their respective parameters. Procedure models may or may not be structured around mathematical models. Mathematical models are usually approximations of physical properties, while procedure models only need to approximate the image of an object within a certain distance range from the observer. For example, a mathematical model of a tree trunk could represent every cell in a tree, while a procedure model need only to create the image of the visible bark.

#### PREVIOUS WORK

Newell (2) used procedure models to generate data that is displayed with a priority list algorithm. His basic elements are polygons and patches. The high-level information returned by his procedure model is the enclosing convex polyhedron for the object generated. This information is then used to make the priority algorithm more efficient. His procedure models can include information on attributes of the generated objects that would be costly to calculate from the physical data structure, like volumes and surface areas. They can combine different representation schemes and different object primitives, like polygons, patches, and lines. View dependence can be built into the procedure model to control the level of detail. Also, any necessary clipping can be done by the model.

In 1974, MAGI (6) produced a color picture of a deciduous tree, made up of several thousand leaves, and displayed at a 500x500 resolution. They used a combinatorial geometry technique that is similar to our procedure models. Their trees are made up of five parts: stems, primary branches, secondary branches, twigs, and leaves. The placement and repetition of the various elements are guided by statistical properties of different tree types, provided by tables generated from botanical studies of trees. The method is optimized for their ray-tracing algorithm, with the basic elements (twigs and leaves) represented as 2-D or 3-D grids with optical properties stored in tables. Each secondary branch on their tree is determined by two primary parameter arrays: length of each branch and the angle of the branch in relation to the primary branches.

A procedure model was used at CGRG (1) to create images of a smoke cloud. A 3-D mathematical approximation of a particulate cloud is generated, and then a type of ray-tracing (with the observer at infinity) is used to create a 2-D array of the intensities in the image. The parameters used include particle concentration, wind velocity, rate of emission, and height of source.

## SYSTEM SOFTWARE FOR PROCEDURE MODEL GENERATION

The Computer Graphics Research Group has designed and implemented a computer animation system called ANTTS (Hackathorn) which employs many interactive techniques and presents a unified approach to the graphical display of complex threedimensional data. It is currently in operation on the group's VAX 11/780, using a frame buffer designed and constructed locally by Marc Howard. The system facilitates the generation, manipulation, and display of highly detailed data with the aid of interactive devices and a video interface connected to a high resolution RGB monitor. The system enables the animator to create a variety of objects (including texture) and to specify the necessary transformations for an animation sequence. A run-length processing technique, combined with a brute force Z-buffer algorithm, has been designed and implemented tht can handle the intersection of several million faces, lines, and points. We are involved in an effort to integrate the procedure model technique for data generation into this software system for the following reasons:

- 1. the system has the ability to handle complex data
- the system can interface easily with various techniques used for data generation, including procedure models
- 3. the system has already been implemented and has been used to produce images of the desired complexity.

A PROCEDURE MODEL FOR TREES

The procedure model for a tree uses many parameters, some of which define the type of tree and others of which control its generation (Fig. 1). Parameters which define the tree type are the leaf type, the branch structure, the color, and the limits on the size and shape of the overall tree. The other group of parameters which determine the specific tree generated include the number of leaves in a tree, the actual size of the branches, the density of the leaves, and the height of the tree. The specific parameters are not independent. For example, the density of the leaves depends on the average size of the branches and the number of leaves. One need only specify two of the three parameters and the other will be implicit. If density and branch size parameters are specified, the number of leaves is already determined. Also, an interval for the specific parameters can be specified, with the value on the interval determined by a pseudo random number generator. This reduces the burden on the user of specifying cach parameter value explicitly.

Randomness is also used in combination with the specified parameters to produce unique trees of a given species. The approximate location of each branch and leaf is determined by the model using the input parameters. The final position and orientation of each individual leaf is selected by a random perturbation of the calculated values. The underlying structure, determined by the procedure model, insures that a realistic image is produced, while the randomness gives the unique appearance of the individual trees.

One of our implemented procedure models for



trees (of a general type) uses the following parameters:

- 1. number of leaves
- 2. length of each branch
- 3. leaf element description
- 4. color
- 5. position of tree
- 6. size of leaf element
- 7. distance between branches
- 8. distance between leaves
- 9. random number seed

Once these parameters have been specified, the degrees of variability of individual trees of this type is determined. On input of the parameters, the procedure model for a tree works as follows: the leaf elements are organized on the branch according to the number of leaves, the length of the branch, and the size of the leaf. These initial element positions are modified slightly with the use of a random number generator to provide a nonuniform orientation (with respect to the rotation of the element) and final position on the branch. After the number of branches specified has been created as above, they are positioned around the trunk element. (Plates 1 4 2) Their positions and rotations are also modified in order to complete the three-dimensional tree model. Finally the tree is positioned in the terrain model according to its input parameter, and colors are selected from the color palette to be assigned to each primitive element according to its type and orientation.

The goal of the prototype technique used by MAGI was the realistic modeling of vegetation and terrain based on a combinatorial geometry technique. Our goal is to use the concept of procedure models to give the same kinds of realism while keeping computation times low enough to allow for efficient design and display of terrain models. A deciduous tree of about the same complexity at 500x500 resolution took approximately 10 minutes to create and display on our PDP 11/45. (While direct comparison of computation time could be misleading, it is interesting to note that the same kind of tree required approximately 3 hours on MAGI's IBM 360/65.) While the basic underlying structure, i.e., building a complex model from simple elements, is the same, the technique described here has certain advantages over a technique such as that used by MAGI. The model is created to satisfy a visual requirement, so that little mathematical or biological accuracy is necessary. MAGI bases their construction on a statistical model obtained from data from biological surveys. Thus, all position, rotation, and magnification information must be included in statistical tables to be input into the system. This information is inherent in the procedure models used here. Because of the generality available with the use of procedure models, it is relatively easy to create a forest of similar or different vegetation types. Also, symmetric or asymmetric crown shapes are done with equal ease, merely by setting limits on the parameter input. All models created by various procedures are processed by the same visible surface routines, with shading, color, and light source subprograms resident. Each model doesn't have to have its own unique processing techniques associated with it.

The following section describes an extension to the camouflage net suggested by MAGI, but realized as another procedure model definition.

## OTHER TERRAIN

The procedure model technique, as exemplified by the above tree model, takes basic elements are a trunk, major branches, and leaf clusters (or leaves and twigs). (Plate 3) Optional elements such as fruits, flowers, and seeds can be incorporated into the tree structure. Each procedure model selects elements from a given class assigned to it depending on the type of spatial model desired. Thus, bushes, flowers, and other vegetation, as well as buildings and other man-made structures, can all be generated using different procedure models and element classes.

One of the major advantages of a procedure model technique is that the class of primitive elements assigned to it does not need to contain previously defined elements. This means that a procedure model can take as its primitive elements the output of one or more other procedure models. This leads to a hierarchical organization that allows for a reasonably simple construction of a complete terrain model. For example, Figure 2 shows the organization necessary to create a model of a sample terrain.

The lowest level element classes (tree elements, building elements, and mountain elements) are previously defined and stored on a mass storage device. The higher level element classes are comprised of output from procedure models. Thus, they need not require any space for storage, and additional flexibility is obtained by being able to define these element classes of the higher level procedure models dynamically.

In Figure 2 procedure models PM1, PM2 and PM3 take basic clements that are created and stored on the disk to create a complex model. The output of these procedure models are used as the primitive elements of procedure models PM4, PM5 and PM6. Finally, PM7 combines these outputs into the finished terrain model.

Figure 3 demonstrates an example of a program in a sample language that creates a forest line.

## USER INTERFACE

In the case of displaying a tree the individual elements which make up a tree are created in a data generation system developed by Parent (7). Using a Vector General display the user draws leaves of the specific tree to be displayed. The same routine is used to make the trunk and branchcs of the tree. When creating the trunk and branches, points indicating the form of the object are placed and connected on the VG screen. By rotation around the Y axis this shape is made to be 3-D. The number of slices around the axis is controlled with the use of a dial. This solid of revolution capability enables the user to specify the number of faces in the trunk. In order to display an accurate picture of a tree, the leaves or leaf clusters as well as the trunk and branches must be created to resemble the actual elements of the tree as closely as possible. For more



realistic images other elements such as seeds, buds, and flowers can be created in data generation.

```
FIGURE 3
```

```
FOR I = 1 to N
```

DO:

X = 100

```
LOOP: L = RANDOM (1000, 3000)
H = RANDOM (50, 100)
S = RANDOM (0, 1)
CALL TREES (Type = POPLAR, Leaves = L,
Height = H, Trunk = TRUNK1,
```

Position = (X,100), Random = S, Extent = ELIST)

X = X + 100

```
GO TO LOOP
```

END

The procedure model used can generate trees of various types. The variables L and H specify limits on the number of leaves and on the height of the tree, respectively. The variable S is a random number seed to the procedure model. The random number seed controls the exact placement of the leaves. The position parameter represents an X-Y mapping onto a previously defined grid. The trunk parameter refers to a trunk named TRUNK1, which was previously defined. The extent is a returned value which is a description of the approximate extent of the tree (a list of enclosing rectangles). Any unspecified parameters can be given default values by the model. The result of this example is a line of slightly different sized poplar trees.

The creation of terrain models also involves building plains, hills, and mountains. One of the more difficult problems confronting the user was the generation and display of a large expanse of ground terrain stretching to the boundaries of the object space. A solution to this problem has been obtained by the introduction of a data generation routine that displays a quadrilateral grid which can be warped in X, Y, or Z to create mountain peaks, or to a lesser degree, rolling hills. The resolution of the grid can be specified by using dials which control the number of vertical and horizontal slices. The greater the number of slices. the more complex the image will be. Each intersection of a vertical and a horizontal line can be warped with a cursor. The degree of warp and area affected by the warp is again controlled with dials. The routine is thus capable of making a surface with peaks and crevices or rolling hills. Once the data is converted into display format a procedure model can be defined that will position plains, mountains, and hills to build a high complexity terrain model consisting of millions of edges. With the display system's virtual intersection capability mountains can be positioned in such a manner that mountain ranges can be formed, or one large mountain can be formed from several smaller ones. Likewise, the ability to copy and display objects allows the procedure models, through rotation and translation, to create a scene consisting of seemingly different objects from one or two

similar objects. Areas of texture, such as corn fields, can be specified by the user for use in building a scene for an animation sequence. By using a procedure model containing distribution information about fields, an element designed and drawn in data generation can be distributed on the screen to create a field of corn or a forest at a great distance from the observer.

CGRG has developed a system of designating colors consisting of a "palette" made up of different colors, each with many levels. Each element used to make up an image on the monitor is assigned a number designating a particular color in the palette, the different levels of a particular hue, or a combination of various hues and intensities. As an example of the latter, consider the case of a tree with autumn foliage. The colors may range from red to brown to yellow with values in-between. These levels in each color are affected by a light source command which enables the user to place the "sun" or light source at any coordinates for the desired effect. The closer the light source is to an object, the more acute the contrast. As the light source is moved, or as the object is moved in relation to the light source, the value changes according to the angle from the object to the light source. Each leaf, then, can have a different intensity value.

# CONCLUDING REMARKS

We feel that the implications for this kind of generation and display system are far reaching. One strength of the system is that it can be used for generating many different types of imagery. As the user desires, the pictures may be suitable for applications in architecture, urban planning, demographics, product design, visual communications, etc. Once the images are created the system allows for the animation of the objects in a straight forward, easy to understand format so that a relatively inexperienced user can produce sophisticated animation with a minimum of effort.

The value of this system is increased because of its accessibility to the inexperienced user. Likewise, the system itself can be enhanced because of the knowledge an animator as an artist can bring to it. The computer is only as good as the user, and while the ability to produce high quality complex images is impressive, the animation system still must be handled by a competent user who can observe and then simulate these observations on a TV monitor.

Results of the rescarch performed during the implementation of these techniques are going to be used by us in further investigations related to flight simulation. Efficient algorithms, for generating, displaying, and manipulating terrain data are essential for realistic imagery to be provided in real time. It might be that a marriage of techniques such as those described here and descriptive data sources, like the Defense Mapping Agency data base, will provide some initial steps toward the realization of this efficiency.

# REFERENCES

 Csuri, C., Hackathorn, R., Parent, R., Carlson, W., Howard, M. Towards an Interactive High Visual Complexity Animation System, Proceedings Siggraph '79, (Aug. 1979), 289-299.

- Newell, M.E., The Utilization of Procedure Models in Digital Image Synthesis. Computer Science, Univ. of Utah, UTEC-CSC-76-218, (1975).
- Dungan, W., Stenser, A., Sutty, G., Texture Tile Considerations for Raster Graphics. Siggraph '78 (Aug. 1978), 130-134.
- Dungan, W., A Terrain and Cloud Computer Image Generation Model. Proceedings Siggraph '79, (Aug. 1979), 143-150.
- Greenberg, D.P., An Interdisciplinary Laboratory for Graphics Research and Applications. Proceedings Siggraph '77, (July, 1977), 90-97.
- Brooks, J., Murarka, R.S., Onuoha, D., An Extension of the Combinatorial Geometry Technique for Modeling Vegetation and Terrain Features. NTIS Report AD-782-883, (Aug. 1974).
- Parent, R., Chandrasekaran, B., Moulding Computer Clay. From <u>Pattern Recognition and Artificial Intelligence</u>. C.H. Chen, Ed., Academic <u>Press</u>, (1977), 86-107.















