VISUAL SIMULATION OF CLOUDS

Geoffrey Y. Gardner

Grumman Corporation
Corporate Research Center
Bethpage, New York 11714

ABSTRACT

Clouds present serious problems to standard computer image generation techniques because clouds do not have well-defined surfaces and boundaries. In addition, clouds contain varying degrees of translucence, and their amorphous structure can change with time. Although several approaches to cloud simulation have produced impressive results, they have relied on complex mathematical models which produce high computation costs for a single image.

This paper describes a new approach to cloud simulation using simple planar and curved surfaces whose surface shading and translucence are modulated by a mathematical texturing function. This approach represents the appearance of clouds with enough realism for a wide range of visual simulation but does so at a reasonable computational cost, allowing the generation of sequences of images on small minicomputers. In addition, the cloud model can be constructed in a straightforward manner.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—display algorithms; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—curve, surface, solid and object representations; geometric algorithms, languages and systems; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—color, shading, shadowing, and texture; visible line/surface algorithm; J.7 [Computers in Other Systems]: Military, real time.

1. INTRODUCTION

The history of computer graphics has been highlighted by advances which have overcome specific stumbling blocks to realistic scene simulation. Terrain, trees, shadows, reflections, and motion blur are examples of stumbling blocks

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.

that have been surmounted by recent innovations in computer simulation. One remaining stumbling block is the common cloud. Clouds are a critical element in air-to-air combat. Yet, for over two decades, computer image generation (CIG) has been used to provide visual displays for flight simulation without the ability to simulate clouds with any degree of realism. Clouds are important in the simulation of intelligent weapon systems which seek and identify aerial targets in cluttered backgrounds. Realistic cloud simulation would also be an effective tool in the field of meteorology. Since clouds are such familiar objects in everyday life, it is desirable to simulate them effectively for other applications, such as entertainment, advertising, and art. Indeed, it is reasonable to say that we want to simulate these beautiful natural features simply because they are there.

The amorphous nature of clouds prevents simulation by standard CIG techniques employing solid surfaces with well-defined boundaries. Several researchers have produced impressive results by breaking away from the standard approach. Blinn [1] developed a functional model for light reflections from cloud surfaces. Max [9] used height fields and ray tracing to model clouds. Voss [10] applied fractal techniques to produce a very realistic looking cloud. Kajiya and Herzen [7] used ray tracing of volume densities to model cloud growth. In general, these efforts attempted to model cloud physics accurately and therefore involved fairly rigorous mathematics. The approach described in this paper is to represent the aesthetic quality of clouds as efficiently as possible. The goal is to allow generation of sequences of realistic looking images with moderate computational cost. This research is the most recent work in a study at the Grumman Corporate Research Center to develop cost-effective scene simulation technology [2-6].

2. A NEW CLOUD MODEL

In order to simulate arbitrary cloud scenes, we must be able to model different types of clouds viewed from a variety of distances and angles. Fortunately, the standard classification of cloud types is based on their appearance. The three fundamental classes are: cirrus ("curl of hair"), stratus ("layer"), and cumulus ("heap") [8]. Cirrus clouds are wispy clouds at high altitudes (5-13 km). Stratus clouds are layer clouds with no

distinct detail, lying at low altitudes (0-2 km). Cumulus clouds are heap clouds, also lying at low altitudes (0-2 km). Combinations of the basic class names are used to describe clouds with combined characteristics and altitudes of the basic types. "Cirrostratus" refers to a high layer of cirrus clouds. "Stratocumulus," "altocumulus," and "cirrocumulus" refer to low, intermediate, and high broken cloud layers, respectively. "Altostratus" refers to a thick layer cloud at intermediate altitudes (2-8 km). "Nimbostratus" refers to a dark layer producing rain or snow, and "cumulonimbus" refers to a large cumulus cloud producing a shower.

In developing a comprehensive model for visual simulation of clouds, it is important to note that cloud formations can follow either horizontal development (cloud layers) or vertical development (cumuliform clouds). To allow for the representation of a wide variety of cloud types and formations and at the same time keep scene modeling simple and image generation economical, we have adopted the strategy of the impressionist painters: to represent the essence of a scene as efficiently as possible.

We compose our cloud model using three basic building blocks:

1. A sky plane
2. Ellipsoids
3. A mathematical texturing function.

We define the sky plane to be parallel to the ground plane at a specified altitude and use it to model a two-dimensional cloud layer viewed from a distance. We define the ground plane in scene coordinates (X,Y,Z) to be the X-Y plane. Therefore, the sky plane can be written as:

$$P(X,Y,Z) = Z - A = 0, \quad (1)$$

where A is the altitude of the plane.

We use ellipsoids to model gross three-dimensional cloud structure. A typical ellipsoid can be expressed as:

$$Q(X,Y,Z) = Q_1X^2 + Q_2Y^2 + Q_3Z^2 + Q_4XY + Q_5YZ + Q_6XZ \\ + Q_7X + Q_8Y + Q_9Z + Q_0 = 0. \quad (2)$$

We use the mathematical texturing function to model cloud detail by modulating the shading intensity and translucence of the sky plane and the ellipsoids. We have found it convenient in modeling and economical in image generation to define the function to represent the spectral content of the texture pattern. We do this by using a "poor man's Fourier series" composed of short sums of sine waves according to the following formula:

$$T(X,Y,Z) = k \sum_{i=1}^n [C_i \sin(FX_i X + PX_i) \\ + T_0] \sum_{j=1}^n [C_j \sin(FY_j Y + PY_j) + T_0]. \quad (3)$$

Natural-looking texture patterns can be produced with four to seven sines in each series if the frequencies and coefficients are chosen by the relationships:

$$FX_{i+1} = 2 FX_i \quad (4)$$

$$FY_{i+1} = 2 FY_i$$

$$C_{i+1} = .707 C_i \quad (5)$$

Note that this gives a 1/f spectral shaping similar to that of fractal surfaces.

PX_i and PY_i are phase shifts to add randomness and are determined by the following relationships for planar texturing:

$$PX_i = \pi/2 \sin(.5FY_i Y) = \pi/2 \sin(FY_{i-1} Y) \text{ for } i > 1 \quad (6)$$

$$PY_i = \pi/2 \sin(.5FX_i X) = \pi/2 \sin(FX_{i-1} X) \text{ for } i > 1$$

The phase shifts produce a controlled pseudo-random effect by shifting the X sine components as a function of Y and the Y components as a function of X.

To provide three-dimensional variations for texturing ellipsoids, the phase shifts are augmented by an added sine variation with Z:

$$PX'_i = PX_i + \pi \sin(FX_i Z/2) \\ PY'_i = PY_i + \pi \sin(FX_i Z/2) \quad (7)$$

T_0 is a parameter controlling contrast of the texture pattern, and k is computed to produce a maximum value of 1 for $T(X,Y,Z)$.

Antialiasing of the texture pattern is facilitated by the organization of the texturing function because, as each higher frequency in the series is computed, it can be tested against the spatial sampling frequency on the scene surface, and the series can be truncated before aliasing frequencies are added in.

The texturing function modulates the shading of a planar or ellipsoidal surface by the following equation:

$$I = (1-a) \{ (1-t) [(1-s)I_d + sI_s] + tI_t \} + a \quad (8)$$

where a is defined fraction of surface reflection due to ambient or scattered light,
t is a defined fraction of texture shading
s is a defined fraction of specular reflection
 I_d is the shading intensity due to diffuse or Lambertian reflection,
 I_s is the shading intensity due to specular reflection,
 I_t is the shading intensity contributed by the texturing function
($I_t = T(X,Y,Z)$),
I is the combined surface shading intensity.

The values of I_d and I_s are computed in the standard way using the vector relations between surface normal, light vector, and eye ray [4].

We modulate the translucence of the sky plane by defining a threshold value for the texturing function below which we assign full translucence to the plane. The modulated translucence is given by the equation:

$$TR = 1 - (I_t - T_1)/D \quad (9)$$

$$0 < TR < 1$$

where T_1 is the threshold value
 D is a range of texturing function values across which translucence varies from 0 to 1
 TR is the modulated translucence.

We modulate the translucence of ellipsoids in a similar manner, but vary the threshold value to increase translucence at the boundary of the ellipsoid. To do this, we use the equation of the limb curve, the elliptical projection of the silhouette of the ellipsoid on the image plane (x, z). This equation is written in image coordinates as

$$f(x, z) = a_1 x^2 + a_2 z^2 + a_3 xz + a_4 x + a_5 z + a_6 = 0 \quad (10)$$

where z is the vertical image coordinate or scan line number
 x is the horizontal image coordinate.

For points on the interior of the limb curve, corresponding to points on the surface of the ellipsoid, $f(x, z)$ is non-zero, with a maximum value at the geometric center of the curve. If we divide $f(x, z)$ by the maximum value we get a normalized limb curve function, $g(x, z)$, which varies from 0 to 1 on the image of the ellipsoid. We can then modify equation (9) to raise the threshold as we approach the limb curve:

$$TR = 1 - (I_t - T_1 - (T_2 - T_1)(1 - g(x, z)))/D \quad (11)$$

$$0 < TR < 1$$

where T_1 defines the threshold at the center of the limb curve
 T_2 defines the threshold at the boundary of the limb curve.

2.1 Two-Dimensional Cloud Model

An efficient model of a cloud layer can be defined by a single textured plane in the sky (Eq (1)). We can place the plane at any altitude, A , and we can specify the solidity and density of the cloud layer by the texturing function translucence parameters T_1 and D (Eq (9)). We can define the spectral content of the cloud pattern by the texturing function parameters C_i , FX_i , FY_i , and T_0 (Eq (3)).

For example, we can model a high cirrus layer at an altitude of 10 km. We represent the characteristic long wispy tails of cirrus clouds by specifying FX_i to be double FY_i and using high values of T_1 and D to produce large regions of

complete translucence fading gradually into opaque clouds (Fig. 1).

We can model a stratus cloud layer at a typical altitude of 2 km, by raising the texture function offset, T_0 , to reduce both translucence and contrast in the intensity modulation. This produces a fuller cloud layer with less detail, characteristic of stratus clouds (Fig. 2).

Fig. 1 2-D Cirrus Cloud Layer

Fig. 2 2-D Stratus Cloud Layer

We can use the two-dimensional model to represent the bottom of a thick cumulus cloud formation. Figure 3 shows a view of an overcast sky modeled by a textured plane at 1 km. We make the clouds opaque by defining a very low value for T_1 .

The textured plane can be used to simulate cloud dynamics by varying selected texturing function parameters with time. We can simulate cloud development by decreasing high values of T_1 and D . We can also move the entire texture pattern across the plane to simulate drifting [6].

The use of the two-dimensional model is limited because it has no vertical depth. For this reason it must be viewed at a distance, serving as

Fig. 3 2-D Cumulus Cloud Layer (Overcast Sky)

a backdrop for three-dimensional scenes. Another limitation is that the same texture pattern is applied across the entire plane, precluding the representation of different cloud formations in different parts of the sky. To complement the two-dimensional model and to allow arbitrary eye movement among the clouds, we need to provide a three-dimensional model.

2.2 Three-Dimensional Cloud Model

We have chosen the ellipsoid as the basic building block with which to model the primary three-dimensional structure of clouds. A single ellipsoid can be used to define an entire volume with only nine parameters, three specifying size and shape, three specifying position, and three specifying angular orientation. As in the two-dimensional model, we use the texturing function to simulate the secondary topographical detail by modulating surface shading intensity and translucence (Eq (8) and (11)). The texture model can be defined in a straightforward manner by specifying the frequencies and amplitudes of the function sine waves to produce a desired spectral content and by choosing the translucence thresholds to produce a desired cloud density. This approach results in a very compact data base because the same texture pattern can be used for any number of ellipsoids.

The elements of the three-dimensional model are demonstrated in Fig. 4. Figure 4a shows a textured ellipsoid, with the texture function modulating only surface shading intensity. Figure 4b shows the same model with modulation of translucence using a constant threshold across the image (Eq (9)). Figure 4c shows the final three-dimensional modulation of translucence which increases translucence near the boundary of the image ellipse (Eq (11)).

Observing real clouds, one can sometimes notice individual clouds that are surprisingly close to ellipsoidal in shape. In general, however, there is a tremendous variety of shapes that cannot be modeled faithfully by isolated ellipsoids. To account for this variety, our three-dimensional model includes the capability to link ellipsoids to create more complex shapes. In

(a) Textured Ellipsoid

(b) Textured Ellipsoid with 2-D Translucence Threshold

(c) Textured Ellipsoid with 3-D Translucence Threshold

Fig. 4 3-D Cumulus Cloud

our original approach to joining quadric surfaces, we chose to insert bounding planes at the junctures of the quadrics. Our strategy was to avoid the computation of fourth-order boundary curves produced by intersecting quadric surfaces [5]. This approach works well for solid objects, but tends to produce unnatural planar boundaries between highly translucent objects. To avoid these anomalies, we have included in our three-dimensional cloud model the capability to handle intersecting quadric surfaces without bounding

planes. To do this we have extended our image generation algorithms to compute the visible points on the curve of intersection between two quadric surfaces. To minimize the added computation, we perform the computation on a scan line basis and start with an approximate depth sort used in the earlier algorithm. Using the computed quadric intersection points, we determine scan line regions in which surface priority is constant. We then reorder the surfaces from front to back within each scan line region. This provides an efficient means of rendering intersecting translucent surfaces. Figure 5 shows how this approach represents a typical cloud with a small number of ellipsoids.

Fig. 5 Complex Cloud Modeled by Linked Ellipsoids

2.2.1 Horizontal Cloud Formations

To model a wide range of horizontal cloud formations, we have developed scene modeling software that generates clusters of ellipsoids. For each cluster we define a template ellipsoid by specifying size and orientation parameters. We also define a horizontal elliptical boundary for the cluster, typical X and Y spacings, and the altitude of the cloud layer. The program replicates the template ellipsoid within the defined boundary at positions separated by the defined spacings. Size, position, and orientation parameters for each ellipsoid are altered by a prescribed amount of random variation. All ellipsoids in the cluster are assigned a common color and set of texture function parameters. We can model a specific class of cloud by choosing the shape of the ellipsoids in the cluster, the altitude of the cluster, and the texture function frequency and transluence parameters. For example, we use round ellipsoids and a high-contrast texture pattern for cumulus clouds, and we use shallow ellipsoids and a low-contrast texture pattern for stratus clouds (Fig. 6).

To facilitate modeling of complex cloud scenes, we extended the cluster software to generate macroclusters, or clusters of clusters. Each macrocluster is defined in the same way as the cluster, by a large horizontal region and grid spacing, a template ellipsoid, an altitude, a color, and a texture pattern. The macrocluster program generates the input data for the cluster program, adding random variations to each cluster. In this way, we can quickly generate a model for complex cloud formations covering large areas of the sky (Fig. 7).

(a) Cumulus Cloud Cluster

(b) Stratus Cloud Cluster

Fig. 6 Cloud Clusters

Fig. 7 Multiple Cumulus Cloud Clusters

2.2.2 Vertical Cloud Formations

We can model vertical development of clouds in several ways. We represent minor vertical development within cloud layers simply by increasing the size of the ellipsoids. By varying the ellipsoid size in a controlled manner, we can model natural variations in both vertical and horizontal cloud development (Fig. 8).

Fig. 8 Vertical Development Modeled by Ellipsoid Scaling

Major vertical development due to updrafts can be simulated by pushing the ellipsoids upward. To do this, we use an option included in our original cluster software. This option, designed to position trees on hills, allows us to define a set of "terrain" objects upon which subsequently generated objects will be placed. As each object is created, its (X,Y) position is tested against the (X,Y) region occupied by each previously defined terrain object. If a new object overlays a terrain object, its position is adjusted in altitude by adding the height of the terrain object at the value of (X,Y). We can therefore raise ellipsoids in a cluster arbitrarily by defining a dummy terrain object below them. This technique can be used to warp a cluster or to model a solid vertical formation using two clusters, one for the base and one for the cap (Fig. 9). This device has several advantages. First, it mimics nature because we simulate the phenomenon of vertical movement. Second, it models solid three-dimensional clusters with only exterior visible ellipsoids. Finally, it is computationally economical because quadric surfaces simplify the terrain object testing.

Fig. 9 Vertical Development Modeled by Terrain Following

2.3 Combined Cloud Model

To represent a variety of clouds in the sky at the same time, we can combine the two-dimensional and three-dimensional models. For example, we can use the two-dimensional model for a high layer of

cirrus or alto-stratus clouds. In addition, we can include a pseudo-two-dimensional model composed of a few thin ellipsoids to represent isolated cloud layer regions (Fig 10). We can also use the ground plane as a two-dimensional cloud model representing a thick cloud layer viewed from above. Ellipsoids bounded by the ground plane can be added to provide three-dimensional clouds rising from the layer. This technique produces a complex scene very economically [3].

3. CONCLUSIONS

The cloud model described here, composed of a sky plane, ellipsoids, and a mathematical texturing function, allows economical visual simulation of a wide variety of complex cloud formations. The model is easy to construct, and image generation can be performed by small computers in a reasonable amount of time. Figures 10 and 11 were generated in 25 and 20 minutes, respectively, at 480 x 640 x 24-bit resolution using a Data General Eclipse S/250 16 bit minicomputer. This economy was achieved by approaching the problem of visual simulation from an aesthetic standpoint rather than attempting to model rigorously the mathematics of atmospheric physics. The aim here has been to

Fig. 10 Cloudscape Using Combined 2-D and 3-D Cloud Model

Fig. 11 Landscape Using Combined 2-D and 3-D Cloud Model

produce a visual impression of clouds as efficiently as possible. As a result, the model as it now stands might fall short of providing adequate simulation in applications where greater physical fidelity is required. However, we make no claim that we have developed the model to its full potential. On the contrary, we have taken only a first step in modeling the tremendous variety, complexity, and subtlety of cloud phenomenology. Nonetheless, we feel that the current model can be useful in many different applications including flight simulation, target recognition, meteorology, entertainment, and art. Because the model provides such a wide range of control over scene geometry, shading, and translucence, it holds the potential for simulating infrared as well as visual images.

4. ACKNOWLEDGEMENT

One reviewer made several helpful suggestions which I hope I have implemented satisfactorily to clarify this paper.

5. REFERENCES

1. Blinn, J.F. Light reflection functions for simulation of clouds and dusty surfaces. *Computer Graphics* 16, 3 (Jul 1982), 21-29.
2. Gardner, G.Y. Computer-generated texturing to model real-world features. *Proc 1st Interservice/Industry Training Equipment Conf, Orlando, FL (Nov 1979)* 239-245.
3. Gardner, G.Y. Non-edge CIG. *SIGGRAPH Video Review*, Issue 11 (Oct 1983).
4. Gardner, G. Y. and Rulon, R.S. Producing high scene content with perspective validity. *Proc 1984 IMAGE III Conf, Phoenix, AZ (May 1984)*, 79-94.
5. Gardner, G.Y. Simulation of natural scenes using textured quadric surfaces. *Computer Graphics* 18, 3 (Jul 1984), 11-20.
6. Gardner, G.Y. Beethoven's Sixth in CIG. *SIGGRAPH Video Review*, Issue 17 (Aug 1984).
7. Kajiya, J.T. and Von Herzen, B.P. Ray tracing volume densities, *Computer Graphics* 18, 3 (Jul 1984), 165-173.
8. Mason, B.J. The Physics of Clouds. Oxford University Press, London (1957).
9. Max, N. The simulation of natural phenomena panel. *Computer Graphics* 17, 3 (Jul 1983), 137-139.
10. Voss, R. Fourier synthesis of gaussian fractals: 1/f noises, landscapes, and flakes. *State of the Art in Image Synthesis. Tutorial No. 10, SIGGRAPH '83 Conf, Detroit, MI (Jul 1983)*.