COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 33, 280–292 (1986)

Light Diffusion through Clouds and Haze

NELSON L. MAX*

Lawrence Livermore National Laboratory, Livermore, California

Received June 18, 1984; accepted July 9, 1985

This paper describes an algorithm for approximating the single scattering model for light diffusion through clouds or haze of constant optical density. The clouds are contained between two single valued surfaces. The algorithm assumes that the sun is directly overhead, but gives convincing pictures for other sun directions. Its efficiency comes from approximating the exponential decay of transmitted light by a quadratic polynomial, and from vectorization on the Cray 1. © 1986 Academic Press, Inc.

1. INTRODUCTION

Blinn [1] has described the physical basis for a single scattering model of light diffusion through clouds or smoke, made up of many small particles. Under the assumptions of this model, the light from the source may be scattered by one of the particles, and is also attenuated by other particles between the source and the scatterer, and between the scatterer and the eye, but secondary scattering from one particle to another and then to the eye is ignored. Blinn [1] showed some pictures of clouds over a simulated planet, and suggested the study of their geometric forms as an area for further research. Voss [2] has recently produced excellent cloud simulations using a fractal optical density defined as a function of three variables. In this paper, I consider the mathematically simpler situation of a constant optical density in the region between the graphs of two related functions of two variables, which define the clouds' extent above and below a mean cloud plane.

Dungan [3] has made pictures of semi-transparent clouds over terrain, integrating the cloud density along each ray. Fishman and Schacter [4] show an example of opaque ellipsoidal clouds, generated by a height field algorithm. In both these cases, the arrays of clouds look excessively regular.

Mendelbrot [5] has used fractals to generate much more random clusters of stars. Snitily [6] has also generated clouds using a fractal height field algorithm, and introducing transparency with a hit count for multiply covered pixels and a blurring post-process.

Norton, Rockwood, and Skolnoski [7] used a tabulated periodic cloud texture function to generate real-time cloud simulations for pilot training, while cleverly suppressing those periodic terms which might cause an aliasing moire, with a technique they call "clamping."

*Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-ENG-48. The U.S. Government's right to retain a nonexclusive royalty-free license in and to the copyright covering this paper, for governmental purposes, is acknowledged. The views and opinions of the author expressed herein do not necessarily state or reflect those of the United States Government thereof, and shall not be used for advertising or product endorsement purposes.

0734-189X/86 $3.00

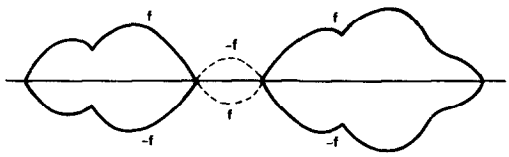

FIG. 1. The function $f(x, y)$ and its mirror image $-f(x, y)$ bound pointed edge clouds, symmetrical with respect to the cloud plane.

2. CLOUD SHAPES USING HEIGHT FIELDS

At a small scale clouds have a random fractal wispiness, and also at larger scales, a fractal distribution of sky coverage (Lovejoy [8]). But at a medium scale, one is impressed by a wavelike periodicity in cloud shapes, presumably caused by periodic waves in the atmospheric disturbances which formed them. Here I model the large scale distribution by a polynomial, and the medium and small scales by a series of superimposed long-crested sine waves, with different wave vectors, amplitudes, and phases. In the limit when enough terms are included, these series recapture the fractals which result from inverse fourier transforms. Here we are content with approximations using from 5 to 15 terms.

Height fields (see [4 or 9]) defined by mathematical functions yield efficient hidden surface algorithms, since the function evaluations are vectorizeable. A combination of polynomials, square roots, absolute values, and trigonometric terms can be used to compute a function $f(x, y)$ defining the clouds' height above a mean cloud plane, and also the depth of the clouds below this same plane. Where f is negative, the clouds are absent. Such a scheme would give clouds with mirror symmetry and pointed "equator" edges at the reference plane $z = H$, as in the cross section in Fig. 1. Therefore, $f(x, y)$ was modified to give two new functions $g(x, y)$ and $h(x, y)$, which have infinite derivative where $f(x, y) = 0$, so that they meet smoothly at the equator, as in Fig. 2. The function $g(x, y)$ defines the height of the clouds above the reference plane and the function $h(x, y)$ defines the depth of the clouds below the plane. In order to flatten the bottoms of the clouds, $h(x, y)$ is organized so that it approaches a finite limit as $f(x, y)$ increases.

The formulas for g and h in terms of f are as follows:

$$t = (\text{abs}(f) + c)^2$$

$$g = \text{sign}\,(\text{sqrt}(t - c^2), f)$$

$$h = -g/\text{sqrt}\,(t + c^2).$$

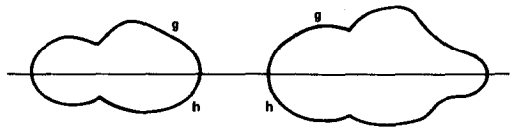

FIG. 2. The functions $g(x, y)$ above the cloud plane, and $h(x, y)$ below, define rounded edged clouds with flattened bottoms.

Here c is a positive constant, independent of x and y and sign (a, b) is the sign transfer function sign $(a, b) = |a|b/|b|$. Effectively,

$$g = \sqrt{f^2 + 2cf}$$

so that

$$\frac{dg}{df} = \frac{f + c}{\sqrt{f^2 + 2cf}}$$

which approaches infinity as f approaches 0. Similarly, dh/df approaches infinity as f approaches 0. Thus the cloud crosses the reference plane with a vertical tangent, as in Fig. 2. The use of the absolute value of f and the sign transfer function assures that g will be defined and negative whenever f becomes negative. This facilitates the height field computations, which interpolate between function values calculated at predetermined samples in a vertical scan plane. As discussed in [9], we establish a vector $\{y_i\}$ of increasing sample y values of the form $y_i = H/(\sigma(maxi + 1 - i))$, so that the distribution of picture plane heights $\beta = (H + g(\alpha y_i, y_i))/y_i$ is approximately σ per pixel, for nearby clouds and as well as distant clouds.

For positive f, h has the form $-\sqrt{(f^2 + 2cf)/(f^2 + 2cf + 2c^2)}$, which approaches -1 as f approaches infinity, flattening the bottom of the clouds. The constant c affects the rate at which g diverges from f and h diverges from -1, and thus affects the curvature at the equator.

3. SINGLE SCATTERING CALCULATIONS

In this section, we analyze the single scattering model introduced in Section 1, for clouds defined by two height functions as in Section 2, in the special case that the sun is directly overhead and the density is constant.

Assume, as in Fig. 3, that the viewer is at the origin, with the picture plane perpendicular to the y axis, and we wish to compute the cloud intensities along the vertical scan line $x_s = \alpha$. The ray through the pixel $(x_s, y_s) = (\alpha, \beta)$ will have direction vector $V = (\alpha, 1, \beta)$ of length $\gamma = \sqrt{1 + \alpha^2 + \beta^2}$. The general point P on this ray has coordinates $(\alpha y, y, \beta y)$.

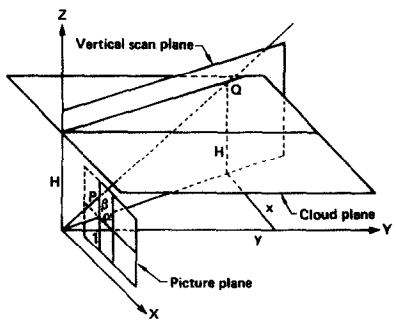

FIG. 3. The ray from the origin through the point $P = (\alpha, 1, \beta)$ on the screen meets the cloud plane in the point $Q = (x, y, H)$.

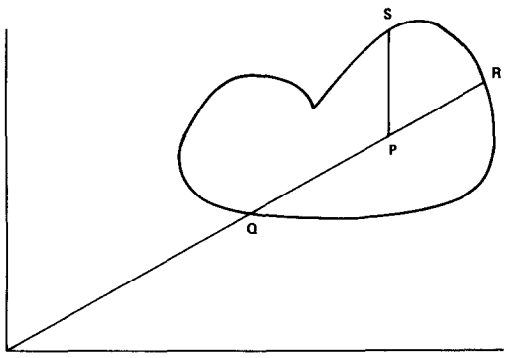

FIG. 4. A viewing ray from the eye piercing the cloud surface at Q and R. The light from the sun traverses the cloud along the segment SP before reaching the viewing ray.

We first consider the case where this ray meets the cloud in a single segment, between $Q = (\alpha y_1, y_1, \beta y_1)$ and $R = (\alpha y_2, y_2, \beta y_2)$. Figure 4 shows the projection in the yz plane of the slice plane $x = \alpha y$.

Let $P = (\alpha y, y, \beta y)$ be a representative scattering point along the ray QR. The vertical ray from P to the sun meets the top surface of the cloud at the point $S = (\alpha y, y, g(\alpha y, y) + H)$. We will write $g(\alpha y, y)$ simply as $g(y)$ below.

The sunlight hitting the cloud at S is attenuated by the scattering along ray SP. Reinterpreting the calculations in Blinn [1], we model the absorption by an optical density ρ per unit length. The total density along SP is then $\rho\,\overline{SP}$, or $\rho\,(g(y) + H - \beta y)$, and the fraction of sunlight reaching P is $\exp(-\rho(g(y) + H - \beta y))$. (Note that our optical densities are interpreted using natural logarithms, rather than the usual logarithms to the base 10.)

Let ds be an element of length along the ray QR. Then the fraction of energy scattered by points P on this element toward the eye is

$$\rho\omega\varphi(a)\,ds,$$

where ω is an albedo, and $\varphi(a)$ is a phase factor depending on the angle a between the incidence direction SP and the viewing direction PQ (see Blinn [1]).

This scattered light is then further attenuated along the ray PQ by a factor

$$\exp(-\rho\overline{PQ}) = \exp(-\rho\gamma(y - y_1)).$$

Therefore the total contribution of light from the line element ds is

$$I_0\exp(-\rho(g(y) + H - \beta y))\rho\omega\varphi(a)\exp(-\rho\gamma(y - y_1))\,ds$$

where I_0 is the intensity of the sunlight incident on the cloud top. To get the total cloud glow $I(y_1, y_2)$ along the ray QR, we must integrate as P varies from Q to R.

Replacing ds by $\gamma\,dy$, we get

$$
\begin{aligned}
&I(y_1, y_2)\\
&\quad = \int_{y_1}^{y_2} I_0 \exp(-\rho(g(y) + H - \beta y)))\rho\omega\varphi(a)\exp(-\rho\gamma(y - y_1))\gamma\,dy\\
&\quad = I_0\rho\omega\varphi(a)\gamma\int_{y_1}^{y_2}\exp(-\rho(g(y) + H - \beta(y - y_1) - \beta y_1 + \gamma(y - y_1)))\,dy\\
&\quad = I_0\rho\omega\varphi(a)\gamma\exp(-\rho(H - \beta y_1))\int_{y_1}^{y_2}\exp(-\rho(\gamma - \beta)(y - y_1) - \rho g(y))\,dy\\
&\quad = K\int_{y_1}^{y_2}\exp(-\delta(y - y_1))\exp(-\rho g(y))\,dy
\end{aligned}
$$

where $K = I_0\rho\omega\varphi(a)\gamma\exp(-\rho(H - \beta y_1))$ and $\delta = \rho(\gamma - \beta)$.

4. APPROXIMATIONS

Now we approximate the exponential function by a polynomial. I have used the approximation

$$
\begin{aligned}
\exp(x) \simeq \operatorname{sexp}(x) &= 0.25\,(\max(x + 2, 0))^2\\
&= 1 + x + 0.25\,x^2 \qquad \text{if } x \geq -2\\
&= 0 \qquad\qquad\qquad\quad \text{if } x \leq -2.
\end{aligned}
$$

This has the correct derivative at $x = 0$, meets the x axis smoothly at $x = -2$, and is never negative.

At the end of this section, we will show how to guarantee that $x \geq -2$ whenever $\operatorname{sexp}(x)$ is used to compute $I(y_1, y_2)$. Applying this approximation to the exponentials in $I(y_1, y_2)$, we have

$$
\begin{aligned}
\exp(-\delta(y - y_1)) &\simeq \operatorname{sexp}(-\delta(y - y_1))\\
&= 1 - \delta(y - y_1) + 0.25\,\delta^2(y - y_1)^2\\
&= 1 + \delta y_1 + 0.25\,\delta^2 y_1^2 + \left(-\delta - 0.5\delta^2 y_1\right)y + 0.25\delta^2 y^2\\
&= a + by + cy^2
\end{aligned}
$$

where a, b, and c are expressions in δ and y_1. Similarly, we have

$$
\begin{aligned}
\exp(-\rho g(y)) &\simeq \operatorname{sexp}(-\delta g(y))\\
&= 1 - \delta g(y) + 0.25\,\gamma^2(g(y))^2\\
&= 1 + dg(y) + e(g(y))^2.
\end{aligned}
$$

Therefore

$$
\begin{aligned}
I(y_1, y_2) &\simeq K\int_{y_1}^{y_2} \operatorname{sexp}(-\delta(y - y_1))\operatorname{sexp}(-\rho g(y))\, dy \\
&= K\int_{y_1}^{y_2} (a + by + cy^2)\big(1 + dg(y) + e(g(y))^2\big)\, dy \\
&= K\Bigg[\int_{y_1}^{y_2} (a + by + cy^2)\, dy + ad\int_{y_1}^{y_2} g(y)\, dy + ae\int_{y_1}^{y_2} (g(y))^2\, dy \\
&\quad + bd\int_{y_1}^{y_2} yg(y)\, dy + be\int_{y_1}^{y_2} y(g(y))^2 dy + cd\int_{y_1}^{y_2} y^2 g(y)\, dy \\
&\quad + ce\int_{y_1}^{y_2} y^2 (g(y))^2\, dy\Bigg].
\end{aligned}
$$

The first of these terms can be integrated trivially. Each of the other terms is of the form

$$m_{ij} = \int_{y_1}^{y_2} y^i (g(y))^j\, dy.$$

These integrals are moments of the region under the curve $z = g(y)$, between y_1 and y_2. Thus m_{01} is the area of the region, m_{11}/m_{01} is the y coordinate of the center of gravity, m_{21} is the moment of inertia about the z axis, $0.5\, m_{02}/m_{01}$ is the z coordinate of the center of gravity, and so forth.

As discussed in Section 2, the values of $g(y_k)$ have been calculated in a vectorized loop, for a vector $\{y_k\}$ of unequally spaced y values y_k. Since we interpret negative values of $g(y)$ as transparent gaps where the density is 0, we replace $g(y)$ below by $\max(0, g(y))$ using a vectorized maximum function. The quantities $y_k^i \max(0, g(y))^j (y_k - y_{k-1})$ can then be calculated in vectorized loops, and the indefinite integral

$$\int_0^{y_k} y^i \max(g(y), 0)^j\, dy$$

can be approximated by

$$M_{ij}(k) = \sum_{l=1}^{k} y_l^i \max(g(y_l), 0)^j (y_l - y_{l-1}).$$

The definite integral $\int_{y_1}^{y_2} y^i (g(y))^j\, dy$ can then be approximated by subtracting $M_{ij}(k1)$ from $M_{ij}(k2)$, where y_{k1} and y_{k2} are the precomputed y values nearest to y_1 and y_2, and compensating for the differences $y_1 - y_{k1}$ and $y_2 - y_{k2}$. Conceptually, we tabulate the accumulated moment as an indefinite integral, and then determine the definite integral by subtraction. Each table entry is gotten by adding one new term to the previous entry, so the table is rapidly computed. The efficiency of the algorithm results from reusing the same indefinite integrals for each pixel in a vertical scan line.

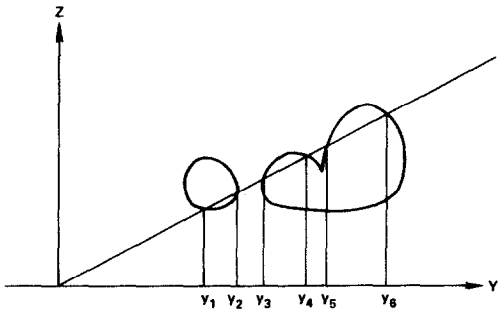

FIG. 5. Projection in the YZ plane of a section through a cloud, with a ray piercing the cloud surface in the six points of y_1 through y_6.

Now consider the general case, where the ray intersects the cloud in a number of disjoint segments, projecting on the y axis to several intervals, say $[y_1, y_2]$, $[y_3, y_4]$ and $[y_5, y_6]$, as shown in Fig. 5. Recall that β represents the slope dz/dy of the ray, as in Fig. 3. The y intervals can be found by using the singularity algorithm (see Newman and Sproull [10]) to scan convert in the (β, y) plane the polygon approximating the cloud outlines (see Sect. 2). Since β represents the slope dz/dy of the ray, the vertices of this polygon are $(\bar{g}_k, y_k)$ and $(\bar{h}_k, y_k)$, where $\bar{g}_k = (H + g(\alpha y_k, y_k))/y_k$ and $\bar{h}_k = (H + h(\alpha_k, y_k))/y_k$.

The lists of intervals are never stored, but processed on the fly. The left-hand interval endpoint $y_L(\beta)$ for each pixel is initialized at 0 to account for a possible viewpoint inside the cloud, a transmission factor $T(\beta)$ is initialized at 1, and an intensity $J(\beta)$ is initialized at 0.

The polygon edges for $\bar{g}_k$ and $\bar{h}_k$ are then processed in order of increasing k, and the total intensity $J(\beta)$ and transmission $T(\beta)$ are accumulated from front to back. Each "front facing" edge is used to update the left-hand endpoint $y_L(\beta)$ for the pixels affected, and each "back facing edge" is used to create a right-hand endpoint $y_R(\beta)$. As each right-hand endpoint is found, the integral $I(y_L(\beta), y_R(\beta))$ is approximated as above, the value $J(\beta)$ is replaced by $J(\beta) + T(\beta)I(y_L(\beta), y_R(\beta))$, and the value $T(\beta)$ is then replaced by $T(\beta)\exp(-\gamma\rho(y_R(\beta) - y_L(\beta)))$. At the end, $T(\beta)$ can be used to weight the contribution of the background color.

We now return to the inequalities necessary to stay in the range where $\text{sexp}(x)$ is quadratic. For the factor $\text{sexp}(-\delta(y - y_1))$, the requirement is that $-\delta(y - y_1) \geq -2$, or $y \leq y_1 + 2/\delta$. This can be guaranteed by replacing each $y_R(\beta)$ by $\min(y_R(\beta), y_L(\beta) + 2/\delta)$. This is equivalent to assuming that light scattered from further than $y_L(\beta) + 2/\delta$ is totally absorbed before it reaches the eye. In addition, $y_R(\beta)$ is bounded by the distance to the nearest opaque surface so that, for example, a cloud and a mountain will intersect properly. For the factor $\exp(-\rho g(y))$, the requirement is that $-\rho g(y) \geq -2$, or $g(y) \leq 2/\rho$. This can be guaranteed by correctly choosing g and ρ, or by replacing $g(y)$ by $\min(g(y), 2/\rho)$.

There are three separate places where the optical density is used: (a) to determine the attenuation of the sunlight along the ray SP of Fig. 4, (b) to attenuate the scattered light along the ray PQ, and (c) to attenuate the background or farther clouds along the ray QR. In the program, three corresponding constants ρ_a, ρ_b, and ρ_c are used, which can be set independently. This is less scientifically correct, but

gives greater flexibility in satisfying the inequalities of the previous paragraph, and in varying the appearance of the clouds.

5. SCATTERING IN HAZE

If there is any haze in the air below a layer of clouds, the pattern of light and shade caused by the clouds will be visible in the haze as columns of rays, apparently converging at the sun. This effect is particularly impressive when there are isolated gaps in an otherwise dense cloud cover. A computational scheme similar to the one above can be used to simulate this effect.

Suppose the haze has a density τ. In Fig. 6, we assume for simplicity that the function $h(x, y)$ is zero, so that the clouds lie entirely above the plane $z = H$. Consider the ray EP from the eye E at the origin, in the direction $(\alpha, 1, \beta)$, meeting the cloud level at $P = (\alpha y_0, y_0, H)$. Let $Q = (\alpha y, y, \beta y)$ be a point on the ray, and let $R = (\alpha y, y, H)$ be the point at cloud level directly above Q.

As above, the amount of light passing through the cloud to point R is $I_0\exp(-\rho g(y))$. The light scattered by a line element $(\alpha, 1, \beta)\,dy$ at Q is $\tau\omega\varphi(a)\gamma\,dy$. If desired, the function $\varphi(a)$ and the density τ can be wavelength dependent, to scatter more blue than red as does dust or haze in the air. The additional path length is $\overline{RQ} + \overline{QE} = H - \beta y + \gamma y = H + (\gamma - \beta)y$, and absorbtion by the haze along this path multiplies the intensity by a factor $\exp(-\tau(H + (\gamma - \beta)y))$.

Therefore, the total intensity scattered by the haze is

$$I_H = \int_0^{y_0} I_0\exp(-\rho g(y))\,\tau\omega\varphi(a)\gamma\exp(-\tau(H + (\gamma - \beta)y))\,dy$$

$$= I_0\exp(-\tau H)\,\tau\omega\varphi(a)\gamma\int_0^{y_0}\exp(-\rho g(y) - \varepsilon y)\,dy.$$

This integral can also be approximated by the methods discussed above. Since $\varepsilon = \tau(\gamma - \beta)$ is small, the upper limit y_0 can be much larger without introducing errors from the polynomial approximation.

The haze attenuates the color at P by a factor $\text{sexp}(-\tau\overline{EP}) = \text{sexp}(-\tau\gamma y_0)$. The color I at pixel (α, β) is then

$$I = \text{sexp}(-\tau\gamma y_0)\ (\text{color at } P) + I_H.$$

Thus the effects of haze can be calculated efficiently using the same tabulated functions $M_{ij}(k)$.

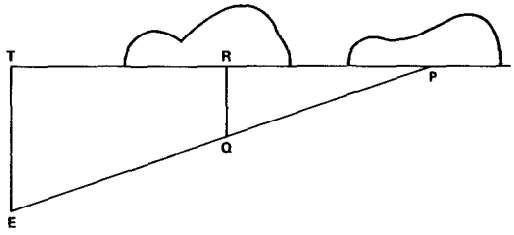

FIG. 6. Ray from the eye at E to a cloud at P, with vertical sun rays TE and RQ.

The restriction that the sun be directly overhead is more limiting here, because the columns of light are usually perceived as radiating out from the sun, rather than as parallel. One could remove this restriction by computing the haze on a polar coordinate raster about the sun, using the methods above, which assume that each scan line represents a plane through the eye and the sun. The methods of Catmull and Smith [11] could then be used to resample the haze into the coordinate system of the final raster image with which it is to be combined.

6. COMPOSITE PICTURES

The results of the cloud algorithm are shown in Figs. 7 through 11, taken from the animated film [12] prepared for presentation with this paper. Each has resolution 510 by 384 pixels, and took approximately 22 s of Cray-1 time.

The terrain data for the San Francisco Bay was taken from a Defense Mapping Agency altitude data base, and vertically exaggerated by a factor of 2. A ray tracing algorithm written by Upson and Weidhaas at Lawrence Livermore National Laboratory produced the image of the terrain along a vertical scan line. The waves were added by the algorithm of Max [9], and were clamped by the algorithm of Norton *et al.* [7] to suppress any moire patterns.

To compute the cloud shadows, a ray was traced from the terrain or water towards the sun direction. At the point (x, y), where this ray pierces the mean cloud plane, the vertical cloud thickness $T = g(x, y) - h(x, y)$ was found, and the sunlight was attenuated by the factor $\text{sexp}(-\rho T)$. The result was used to multiply the diffuse reflection component of the terrain shading, and to modify the color of the wave shading component representing light scattered upward by particles in the water and then refracted towards the eye.

FIG. 7. Scattered clouds above San Francisco airport, viewed from below.

FIG. 8. Scattered clouds above downtown San Francisco before it was settled, viewed from above.

FIG. 9. Dense clouds above Angel Island, with Mount Tamalpias in the distance.

FIG. 10. Sunlight shining through hole in clouds above San Bruno hills, lighting up haze.

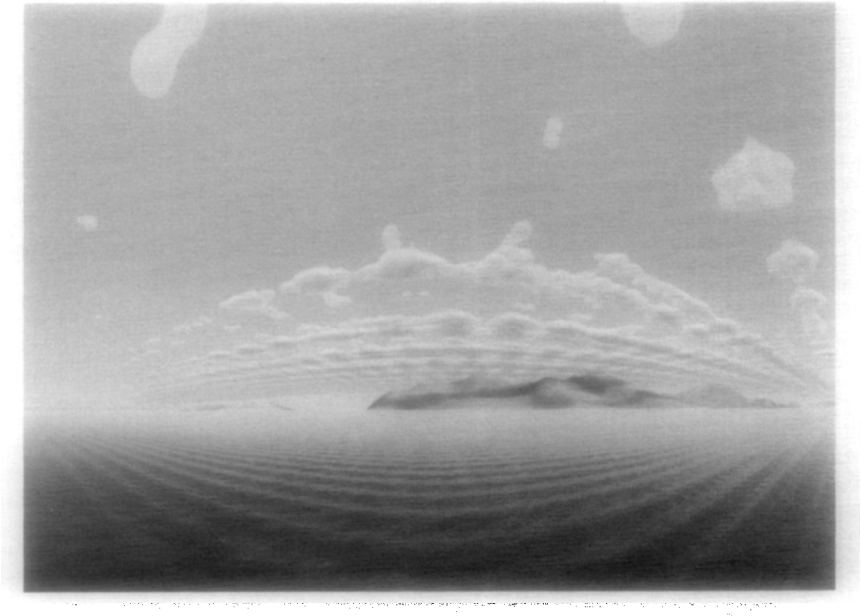

FIG. 11. A 180° view of Marin County, showing San Francisco and the Golden Gate on the left, and Mt. Tamalpias on the right.

The clouds were then added by the algorithm described in Sections 2 through 4. Figures 7 through 9 used 11 terms in the trigonometric series, Fig. 10 used 7 terms, and Fig. 11 used 10 terms. The three density constants described at the end of Section 4 were $\rho_a = 0.0093$, $\rho_b = 0.0163$, and $\rho_c = 0.0198$, with all distances in meters.

Figure 11 shows a 180° view of Marin, in cylindrical coordinates, suitable for projection in the IMAX or OMNIMAX format (see [13]). It is part of a 49 frame cycle prepared for the Siggraph '84 Omnimax film. The wave vectors for the trigonometric cloud terms were arranged so that each cloud moves one rank towards the viewer during each cycle. Each vertical scan line represents a vertical plane through the eye, so the height field algorithms still apply. It took $2\frac{1}{2}$ min of Cray-1 time to compute Fig. 11, at 1764 × 1280 resolution.

There is a noticeable defect in the single scattering model: clouds are actually brighter than the model predicts when they are thick enough for multiple scattering to be important, but agree with the model near their almost transparent edges where multiple scattering is unlikely.

Light coming directly from the sun is not completely lost after it has been scattered once; it can still diffuse down through the cloud by further scattering. However, the light from the background can no longer contribute to the coherent transmitted image after it has been scattered once.

If $\rho_c > \rho_b$, the attenuation of the background increases more rapidly than the cloud's internal glow near its edge. This darkens the cloud edges realistically, partially compensating for these defects in the model.

The haze color was mixed with the final picture using the factor $\text{sexp}(-\tau d)$, where d is the distance to the closest ray piercing point. In Figs. 7 through 9, $\tau = 0.00004$, causing complete obscuration at $d = 50{,}000$ meters.

In Fig. 10, T was 0.0002, and the haze glow of Section 5 was added. The cloud reflections were generated in the same way as the cloud shadows. Reflected rays were computed as in Max [9], and traced until they pierced the mean cloud plane. If the cloud thickness T at the piercing point was greater than zero, the factor $\text{sexp}(-\rho T)$ was used to mix a light color for the cloud edges with a dark color for the thicker regions. In Fig. 10, the reflections of the clouds and sky were further attenuated by the haze along the reflected ray from the waves to the mean cloud plane.

No anti-aliasing has been used. Nevertheless, the cloud profiles look quite smooth, because the variable transparency near their edges produces an averaging effect similar to that of an anti-aliasing algorithm.

The sun was in the direction $(-1, 0, 3)$ for the purposes of terrain and wave shading and cloud shadows, and was vertical for the purpose of cloud shading. This inconsistency is not disturbing in the pictures.

7. CONCLUSION

The single scattering model, with a quadratic approximation to the exponentials, can give realistic renderings of light diffusion through semi-transparent clouds. By reusing accumulated moments, the scattering computations can be made very efficient.

ACKNOWLEDGMENTS

I wish to thank Teresa Higuera for typing this paper, especially the equations, Craig Upson and Pat Weidhaas for their terrain algorithms, John Blunden for the

optical printing in the accompanying film, Dan Asimov for debugging help, Paul Renard for plotting help.

REFERENCES

1. J. Blinn, Light reflection functions for simulation of clouds and dusty surfaces, *Comput. Graphics* **16**, No. 3, 1982, 21–29.
2. R. Voss, "Fractal Forgery," presentation at Siggraph '83 tutorial number 10, State-of-the-Art in Image Synthesis, Detroit, Michigan, July 26, 1983.
3. W. Dungan, A terrain and cloud computer image generation model, *Comput. Graphics* **13**, No. 2, 1979, 143–150.
4. B. Fishman and B. Schachter, Computer display of height fields, *Comput. Graphics* **5**, 1980, 53–60.
5. B. Mandebrot, *Fractals, Form, Chance, and Dimension*, Freeman, San Francisco, 1980.
6. M. Snitily, personal communication, and thesis, University of Washington, 1980.
7. A. Norton, A. Rockwood, and P. Skolnoski, Clamping: A method of antialiasing textured objects by bandwidth limiting in object space, *Comput. Graphics* **16**, No. 3, 1982, 1–8; accompanying film of Evans and Sutherland novoview flight simulation, in *Siggraph '82*.
8. S. Lovejoy, Area perimeter relation for rain and cloud areas, *Science (Washington, D.C.)* **216**, April 9, 1982, 185–187.
9. N. Max, Vectorized procedural algorithms for natural terrain, *Comput. Graphics* **15**, No. 3, 1981, 317–324.
10. W. Newman and R. Sproull, *Principles of Interactive Computer Graphics*, 2nd ed., McGraw–Hill, New York, 1979.
11. E. Catmull and A. Smith, 3-D transformations of images in scanline order, *Comput. Graphics* **14**, No. 3, 1980, 279–285.
12. N. Max, C. Upson, and P. Weidhaas, *Ethereal Flight*, color sound 16 mm film, 1983 available from John Blunden, Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550.
13. N. Max, Computer graphics distortion for IMAX and OMNIMAX projection, in *Proceedings of Nicograph Conference*, December 1–3, 1983, Nihon Keizai Shimbun, Inc., Tokyo.