Conf-9405113--1

UCRL-JC-115863
PREPRINT

Computer Animation of Clouds

Nelson Max

This paper was prepared for submittal to
Computer Animation '94
Geneva, Switzerland
May 25-27, 1994

January 28, 1994

Lawrence Livermore National Laboratory

Computer Animation of Clouds

Nelson Max
Lawrence Livermore National Laboratory
P.O. Box 808 / L-301
Livermore, California 94551
(max2@llnl.gov)

Abstract

Computer animation of outdoor scenes is enhanced by realistic clouds. I will discuss several different modelling and rendering schemes for clouds, and show how they evolved in my animation work. These include transparency-textured clouds on a 2-D plane, smooth shaded or textured 3-D cloud surfaces, and 3-D volume rendering. For the volume rendering, I will present various illumination schemes, including the density emitter, single scattering, and multiple scattering models.

2-D Clouds

When clouds lie in a thin layer high above the viewer, they do no change their apparent shape as they or the viewer move. Therefore they can be modelled on a horizontal 2-D texture plane. The texture is a real valued function $f(x,y)$ varying between 0 and 1. It is used to determine both the transparency and intensity of the clouds, according to the compositing equation

$$\text{new\_color} = f(x,y)\cdot\text{cloud\_color} + (1. - f(x,y))\cdot\text{old\_color}. \quad (1)$$

Here (x,y) is the point where the viewing ray hits the texture plane.

Figures 1 and 2 show such 2-D clouds above the ocean with two different color schemes. They are taken from the last scene of the film "Carla's Island", prepared for Siggraph '81. This scene used a 144 frame cycle of ray-traced wave motion, with the colors and the sun or moon position changing slowly during the cycle's 20 repetitions. Each pixel was assigned one of 256 possible 8-bit color table addresses, according to the ray's final intersection with an island, a cloud, or the sky, after possible reflection in the water. The color table was changed gradually during the animation, and the sun or moon was added behind the clouds. The details of the color table animation, the fast vectorized ray-tracing algorithm, and the wave motion are given in Max [1]. Here I will describe the details of the sky and cloud shading.

The sky color was divided into 12 bands, according to the vertical component of the viewing ray, so that the sky color could be made paler or warmer near the horizon. This vertical component was jittered by adding a random number before being truncated into one of the 12 bands, in order to break up the sudden color changes between the bands. The cloud opacity $f(x,y)$ was also jittered before being truncated into one of three values: completely transparent, half transparent, or completely opaque. There were 12 entries in the color table for half-transparent cloud pixels, containing the equation (1) mixture of sky color and cloud color. There was a single cloud color, which varied with the time of day.

When the sun goes behind a cloud, the semi-transparent edges shine with forward scattered sunlight, giving a "silver lining". I was able to use the 12 sky/cloud mixture color entries to give an approximation to this effect. When the sun moved near the level in the sky corresponding to one of the12 bands, the color mixture was gradually brightened, and then faded as the sun passed. Figure 2 shows this effect with the moon behind the clouds.

After the sunset, the cloud color changes to a red glow. Pixels whose viewing ray intersects the cloud after one or more reflections in the water get the table address corresponding to this intersection, so the glowing clouds are reflected in the water.

The 144 frames were stored on three magnetic tapes, and automatically cycled during the five days necessary to record the two minute sequence onto 35mm film. The minicomputer controlling the film recorder gradually changed the color table, and added in the sun or moon behind the clouds. The ray tracing itself took only 20 minutes on the Cray-1 supercomputer.

A year later, in 1982, I prepared a real-time demo of this animation on the new Apollo DN660 workstation. I stored a 15 frame cycle of the lower section of the frame, containing the moving waves, in the off-screen image memory. I then used the workstation CPU to change the color tables, and fast raster operations to put the sun or moon behind the clouds, and cycle the waves into the displayed image. The next year, for the 1983 Electra Exhibition in Paris, I made the animation interactive, so that the user could change the time of day, the phase of the moon, or the basic colors being interpolated. (See Max[2] for details.) This interactive installation was also exhibited at the Siggraph '85 art show.

Cloud textures

The cloud texture function in these images was the sum of a quadratic polynomial and a trigonometric series:

$$F(x,y) = d - a(x - x_0)^2 - b(y - y_0)^2 + \sum_{i=1}^{n} \sin(p_i x + q_i y + r_i). \qquad (2)$$

If you look at a bank of clouds, you often see periodic rows of density, presumably caused by some quasiperiodic physical phenomenon like convection cells. The trigonometric series approximates this periodicity. Many sine terms with different amplitudes, frequencies, and directions produce random interference, so that every region is slightly different. The polynomial term serves to concentrate the clouds in a particular region of the sky.

The actual opacity function f for use in equation (1) is defined by

$$f(x,y) = \max(1, \min(0, F(x,y)))$$

so when F is negative, f becomes 0 and the clouds become completely transparent. This happens far from the center (x_0, y_0) of the decreasing polynomial term.

It is possible to get a more natural fractal appearance by taking more and more terms in the trigonometric series. If all the terms ia a 2-D discrete Fourier transform are included, the resulting fractal texture can be computed by Fast Fourier Transform (FFT) techniques, as suggested by Voss [3].

Gardner [4] computed a "poor man's fractal" texture as a product of two series of perturbed sine waves:

$$F(x,y) = \left\{ c \sum_{i=1}^{n} 2^{-i/2} \sin(2^i x + \frac{\pi}{2} \sin(2^{i-1} y)) + d \right\} \times \left\{ c \sum_{i=1}^{n} 2^{-i/2} \sin(2^i y + \frac{\pi}{2} \sin(2^{i-1} x)) + d \right\}. \qquad (3)$$

The amplitudes are chosen to give the "$1/f$" power spectrum characteristic of fractals, and the phase perturbations like $(\pi/2)\sin(2^{i-1}y)$ are added to break up the straight wave crests. The 2-D cloud texture in figure 3 used such a function.

This figure is one frame in an animated film called "Sun and Shade", showing the illumination changes as the sun goes behind a cloud. The trees were produced by Jules Bloomenthal [5]. The glow in the haze beneath the trees was computed by the method of Max [6]. A 48 frame cycle of the leaves fluttering in the breeze was produced, with separate images for the haze glow, and for the sharp point-source shadows on the ground. This motion cycle was repeated as the illumination changed.

For each final frame, the sharp shadow image was convolved with the image of the sky and clouds, including the bright sun as covered by the clouds. This convolution was performed by FFT techniques, and resulted in a new shadow image showing the penumbra from the sun and the effects of sky illumination. Details of this convolution scheme are given in Max[7]. The moving clouds were then added behind the trees, and the haze glow was added proportional to the varying intensity of the direct sunlight.

2-D Cloud Surfaces

Clouds can also be rendered as smoothly shaded surfaces, either ray traced analytic surfaces, or polygonal approximations.

I defined analytic cloud surfaces using two single valued functions: $g(x,y)$ representing the height of the upper surface above a central cloud plane, and $h(x,y)$ representing the height of the lower surface. This permits efficient "height field" ray tracing, as described in Max[1].

I started with the function $F(x,y)$ in equation (2), and let

$$g = (F^2 + 2cF)^{1/2}$$

and

$$h = -g / (F^2 + 2cF + 2c^2)^{1/2},$$

with c an adjustable constant. When $F(x,y)$ is negative, the clouds are absent. As shown in Max[8], this makes the upper and lower surfaces meet smoothly with a vertical tangent plane at the cloud edges where $F(x,y) = g(x,y) = h(x,y) = 0$. Also, h is always > -1, and approaches -1 as F approaches infinity. This has the effect of flattening the edges of the clouds realistically, even when they are puffy on top.

In the second scene of "Carla's Island", the clouds grew until they covered the sky, as a storm began. This was done by gradually changing the coefficients in the expansion of $F(x,y)$, in particular, by increasing the constant term d. Figure 4 is a frame from this scene, produced by ray tracing the two height field surfaces.

Ideally, cloud shapes should come from a weather simulation, instead of an ad-hoc mathematical formula. Kajiya and Von Herzen [9] produced volume rendered clouds from a simple weather simulation. At Lawrence Livermore National Laboratory, we have been visualizing cloudiness data from a global climate simulation. (See Max, Crawfis, and Williams [10].)

The cloud data represents per-cent cloudiness in one of 19 variable thickness layers above a 1.125° latitude-longitude square. Figure 5 shows the contours of 15% cloudiness, as smoothly shaded polygonal surfaces. To generate the polygons, we use grid cells whose vertices corresponded to the centers of the cloudiness data cells. We divided each grid cell crossed by the contour surface into five tetrahedra. Inside each tetrahedron, we assume the cloudiness function is linear, so that the contour surface is a planar quadrilateral or triangle. To make figure 5, we applied a smoothing filter to the simulation output, before generating the contour polygons, to get rounder clouds.

Clouds with textured surfaces

Gardner [3] was able to generate realistic clouds with ellipsoidal surfaces, using a 3-D version of the perturbed trigonometric series $F(x,y)$ in equation (3) as a transparency texture. We textured our cloudiness contour surfaces with a similar sum of perturbed sin waves. We used a 3-D texture function because it is difficult to assign 2-D texture map parameters to the polygon vertices of a contour surface. As in the 2-D case, F represents an opacity, and the clouds are completely transparent when F is negative. Figure 6 shows the same clouds as in figure 5, with the front-facing polygons rendered with this transparency texture.

These polygons must be composited by equation (1) in back-to-front order, so they need to be sorted. The volume rendering algorithm in the next section also requires the volume cells to be sorted for compositing in back-to-front order. A sorting scheme is given in Max[11] which is adapted to the complex geometry of this simulation, which used altitude levels curving to follow the terrain heights.

In the animation, we needed to make the 3-D texture function move with the clouds as they are advected with the wind. Therefore we advected the coordinates of the texture function by the wind velocity field output from the simulation. Details are given in Max and Crawfis [12].

Volume Rendering Models

Actual clouds result from a volume density of water droplets, which varies from point to point. To get physically accurate images of clouds, it is necessary to use volume rendering techniques. These vary in the sophistication with which they model the light scattering inside the cloud.

Sunlight reaches a water droplet only after passing between or being scattered from other droplets, so a given droplet is in partial shadow. The simple density-emitter model of Sabella [13] neglects this shadowing, and assumes that each droplet emits a glow of its own, independent of its position in the cloud. However, the model does account for the attenuation between the emitting droplet and the viewpoint, due to absorption and scattering.

To calculate this attenuation, assume that the droplets are all spheres of the same radius r and volume $4/3\ \pi r^3$. Let the density $\rho(X)$ represent the volume fraction taken up by water droplets at position X, and let U be a unit vector along the viewing ray from the viewpoint at (0,0,0).

Consider a thin cylinder of the cloud, normal to the viewing direction U, of thickness ds, cross section area A, and volume $V = A\ ds$. This cylinder has a volume ρV of water, contained in $N = \rho V / (4/3\ \pi r^3)$ droplets. If ds is small enough so that their overlap can be neglected, these droplets have projected surface area $N\pi r^2$. Thus the fraction of the cylinder cross section area A that they occlude is $N\pi r^2/A = \rho\ A\ ds\ \pi r^2/(4/3\ \pi r^3 A) = 3\rho ds/(4r) = \tau\ \rho\ ds$, where the quantity $\tau = 3/(4r)$ is the opacity per unit length per unit volume fraction.

Suppose a source, for example a glowing droplet, at position s_0U a distance s_0 along the viewing ray, emits an intensity I_0 towards the viewpoint. Let $I(s)$ be the intensity remaining at position sU, $0 \le s \le s_0$, after attenuation by the droplets between sU and s_0U. As shown above, the occlusion fraction along a ray segment ds is $\tau\ \rho\ ds$, so we have

$$I(s) = I(s + ds) - \tau\ \rho(sU)\ ds\ I(s + ds)$$

This can be rearranged as the differential equation

$$\frac{I(s+ds) - I(s)}{I(s+ds)} = \frac{dI}{I} = \tau\rho(sU)\,ds$$

which integrates to

$$ln(\ I(s)\ ) = \exp(\ \int_0^s \tau\rho(tU)dt\ ) + C$$

or

$$I(s) = K\exp(\ \int_0^s \tau\rho(tU)dt\ ).$$

The integration constant $K = \exp(C)$ can be found from the boundary condition

$$I(s_0) = I_0 = K\exp(\ \int_0^{s_0} \tau\rho(tU)dt\ )$$

with solution

$$K = I_0 \exp(\ -\int_0^{s_0} \tau\rho(tU)dt\ ).$$

Thus

$$I(s) = I_0 \exp(\ -\int_s^{s_0} \tau\rho(tU)dt\ ),$$

and at the viewpoint,

$$I(0) = I_0 \exp(\ -\int_0^{s_0} \tau\rho(tU)dt\ ). \qquad (4)$$

Am alternate derivation of this equation, using the Poisson distribution, is given by Blinn [14].

Now, suppose, as Sabella did, that each droplet glows diffusely with an intensity κ. In the cylinder of thickness ds discussed above, the fraction of the cross section covered by projected droplets is $\tau\ \rho\ ds$, so the average intensity is $\kappa\ \tau\ \rho\ ds$. Substituting this for I_0 in equation (4), we get the contribution dI of this cylinder to the intensity at the viewpoint:

$$dI = \kappa \tau \rho(s_0 U)\, I_0 \exp\left(-\int_0^{s_0} \tau\rho(tU) dt\right) ds.$$

Finally, integrating over all such thin cylinders along the ray, we get Sabella's formula

$$I = \int_0^{\infty} \kappa \tau \rho(sU) \exp\left(-\int_0^{s} \tau\rho(tU) dt\right) ds. \qquad (5)$$

This can be simplified by noting that

$$\frac{d}{ds}\exp\left(-\int_0^s \tau\rho(tU)\, dt\right) = -\tau\rho(sU)\exp\left(-\int_0^s \tau\rho(tU)\, dt\right)$$

so that

$$I = -\kappa\int_0^{\infty} \frac{d}{ds}\exp\left(-\int_0^s \tau\rho(tU)\, dt\right) ds$$
$$= \kappa\left(1. - \exp\left(-\int_0^{\infty} \tau\rho(tU)\, dt\right)\right).$$

The total transparency along the ray is

$$T = \exp\left(-\int_0^{\infty} \tau\rho(tU)\, dt\right). \qquad (6)$$

Thus we get a version of the compositing equation (1):

composite_color = (1. - T) κ + T background_color (7)

Figure 7 shows the clouds from the climate simulation, rendered with these equations. The density ρ was taken from the per-cent cloudiness data. The volume cells were sorted from back to front, and composited using equation (7). Along each ray/cell intersection segment, the density was assumed to vary linearly between the endpoint values on the cell faces, which were found by Gouraud style bilinear interpolation from the cell vertex values. Further details of this projection technique are given in Max, Crawfis, and Hanrahan [15].

Directional Scattering

Let a, the albedo, be the fraction of light leaving a droplet that is scattered; the rest, 1 - a, is absorbed. For water droplets, the scattering is not isotropic, but depends on the angle between the incoming and scattering directions. Mie [16] has developed series expansions for the scattering from spherical droplets, which depend on the size of the droplet and the wavelength of the light. Keeping these other parameters fixed, the directional dependence is expressed by a phase function $p(\omega,\omega')$, where ω is the unit direction vector of the incoming ray, and ω', of the scattered ray.

If $p(\omega,\omega')$ depends only on the angle between the two vectors ω and ω', as is the case for Mie scattering, it can be expressed as a function of their dot product $x = \omega \cdot \omega'$. A common approximation to Mie scattering is the Henyey-Greenstein phase function [17]:

$$p(\omega, \omega') = \frac{1}{4\pi}\frac{1-g^2}{(1+g^2-2gx)^{3/2}},$$

where g is a parameter which is positive for forwards scattering, negative for backwards scattering, and zero for isotropic scattering.

The projected droplet area argument from the last section shows that along the length ds of the viewing ray, in direction ω, the scattered intensity from irradiance I_0 coming from direction ω', is $a\ \tau\ \rho\ I_0\ p(-\omega,-\omega')\ ds$. The minus signs are needed because the vector arguments in p refer to the direction of light propagation. (Blinn [14] uses a different convention.)

Single Scattering

An improvement on Sabella's density emitter model is the single scattering model, which accounts for attenuation of sunlight along an illumination ray from the sun to the scattering droplet, as well as from the droplet to the eye. However, it neglects multiple scattering effects.

Let V be the unit vector in the direction of the sun, and let J_0 be the sun's irradiance (its intensity times the solid angle it subtends). By the transparency integral (6), the irradiance reaching a point sU on the viewing ray is

$$I_0 = J_0\exp\left(-\int_0^{\infty} \tau\rho(sU+tV)\, dt\right),$$

so the scattered intensity dI along a ray length ds is

$$dI = a\tau\rho(sU)\, J_0 p(-U,-V)\exp\left(-\int_0^{\infty} (\tau\rho(sU+tV)\, dt)\right) ds.$$

Accounting for the attenuation up to the viewpoint and integrating along the viewing ray, as in the derivation of equation (5), the total single scattering intensity is

$$I = \int_0^{\infty} a\tau\rho(sU)\, J_0 p(-U,-V)$$
$$\exp\left(-\int_0^s \tau\rho(tU)\, dt\right)\exp\left(-\int_0^{\infty} \tau\rho(sU+tV)\, dt\right) ds. \qquad (8)$$

For densities $\rho(x,y,z)$ defined on a cubical voxel grid, Kajiya and Von Herzen [9] propose a two pass "slab" method for evaluating this integral. In the first pass, light is propagated from the sun into each horizontal layer or "slab" of the grid in turn, taking into account the attenuation of the slab before continuing to the next one below. This pass results in an auxiliary array $i(x,y,z)$ of illumination for each voxel. Then in the second pass, the integral

$$\int_0^{\infty} a\tau\rho(sU)\, p(-U,-V)\, i(sU)\exp\left(-\int_0^s \tau\rho(tU)\, dt\right) ds$$

is evaluated along the viewing rays.

I have found an efficient way of approximating the integral of equation (8), for the case of clouds bounded by two height fields $g(x,y)$ and $h(x,y)$, with the density ρ constant inside the clouds and zero outside, and with the sun directly overhead. I approximated the function $\exp(x)$ by the function $\mathrm{sexp}(x) = \max(x+2,0)^2/4$, which is equal to 0 if $x \geq 2$, and equal to $1-x+x^2/4$ if $x \leq 2$. By replacing $\exp(-x)$ by $\mathrm{sexp}(-x)$ twice in equation (8), I can be expressed as a linear combination of precomputed moments of the function $g(x,y)$ in vertical slice planes, whose coefficients depend on the direction of the viewing ray. The scattering from the haze under the clouds, as illuminated by the light coming through the clouds, can be found from the same precomputed moments. See Max [8] for details of this technique.

Figure 8 shows a single frame from the film "Ethereal flight, produced by this technique. Note the vertical light beams in the haze, illuminated by the light coming through the hole in the clouds. This hole in the solid cloud cover was produced by giving the quadratic polynomial terms in equation (2) positive instead of negative coefficients, so that the cloud thickness increased away from the center (x_0, y_0) of the hole. The phases r_i of the sine terms were changed linearly as time progressed. They all changed by an exact multiple of 2π after a cycle of several hundred frames, so that the animation could be cycled repeatedly.

Multiple Scattering

The ultimate in computer simulation of cloud illumination is to account for the multiple light scattering from one droplet to another. To do this, it is necessary to solve for the intensity $I(X,\omega)$ at each point X in the cloud and for each unit direction vector ω.

Let U be a viewing direction from point X, with $\omega = -U$ the corresponding light propagation vector towards X. Let $X + s_0U = X - s_0\omega$ be the point where the viewing ray finally leaves the cloud, and let $I_0(X - s_0\omega, \omega)$ be the known illumination entering the cloud in direction ω at this point, from the sun or the background scene.

The intensity scattering into direction ω from a ray segment of length ds at $X - s\omega$ is

$$ds \int_{4\pi} a\,\tau\,\rho(X - s\omega)\,p(\omega',\omega)\,I(X - s\omega, \omega)\,d\omega'$$

where 4π represents the sphere of incoming directions ω'.

Taking into account the attenuation of the background intensity $I_0(X - s_0\omega)$ by the transparency in equation (6), and the attenuation of the scattered light by the droplets between X and $X - s\omega$, as in the derivation of equation (5),

$$I(X,\omega) = I_0(X - s_0\omega, \omega)\exp\left(-\int_0^{s_0}\tau\,\rho(X - t\omega)\,dt\right) + \int_0^{s_0}\left[a\,\tau\,\rho(X - s\omega)\exp\left(-\int_0^{s}\tau\,\rho(X - t\omega)\,dt\right)\int_{4\pi} p(\omega',\omega)\,I(X - s\omega, \omega)\,d\omega'\right]ds. \quad (9)$$

Note that the unknown function $I(X,\omega)$ appears on both sides of this integral equation.

If the cloud is divided up into a finite number of volume cells, and the unit sphere is divided into a finite collection of direction bins, there are a finite number of unknowns. I have developed an efficient approximate solution method, by propagating the light from cell to cell, as in progressive radiosity.

For each volume cell X_i and each direction bin ω_j I save an estimate of $I(X_i, \omega_j)$ and also the "unshot" radiosity $U(X_i, \omega_j)$. These quantities are initialized as in the first pass of the slab method of Kajiya and Von Herzen [9]. Then the albedo a and phase function $p(\omega_j, \omega_k)$ are used to scatter the unshot radiosity in successive passes in each of the six axis directions through the slabs making up the cloud volume. Details are given in Max [18], and a similar algorithm is described by Patmore [19]. Figure 9 shows a cloud rendered by this technique.

The cloud was a "noisy" union of ellipsoids. The density $\rho(x,y,z)$ was the maximum of several polynomials of the form

$$d - \frac{(x-x_0)^2}{a^2} - \frac{(y-y_0)^2}{b^2} - \frac{(z-z_0)^2}{c^2}$$

plus a version of Ken Perlin's $1/f$ noise function [20].

Once the intensities $I(X_i, \omega_j)$ have converged, which takes several hours, the cloud can be rendered from an arbitrary viewpoint in a few minutes, by calculating the integrals in equation (9), as in the second pass of the slab method. I have made an animated film cycle, showing the cloud in figure 9 from several directions. For example, figure 10 shows the sun almost behind the cloud, lighting up its edges. The anisotropic nature of the phase function, with a pronounced peak in the forward scattering direction, makes the "silver lining" around the edge shine brightest near the sun.

Acknowledgements

Several of the algorithms surveyed here were joint work with Roger Crawfis. John Blunden, Eugene Cronshagen, Dean Williams, Chris Anderson, Donald Harrison, Jan Nunes, Michael Allison, Brian Cabral, Jules Bloomenthal and Ross Gaunt helped produce the videos. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract number W-7405-ENG-48, with specific support from an internal LDRD grant and also from the High Performance Computing and Communications Program directed to Visualization For Global Climate Modeling.

References

1. Max, Nelson "Vectorized Procedural Models for Natural Terrain: Waves and Islands in the Sunset," Computer Graphics Vol. 15 No. 3 (August 1981) pp. 317 - 324.
2. Max, Nelson "Carla's Island Revisited," The Visual Computer Vol. 2 No. 3 (1986) pp. 171 - 173.
3. Voss, Richard "Fourier Synthesis of Gaussian Fractals: 1/f Noises, Landscapes, and Flakes," in Tutorial on State of the Art Image Synthesis, ACM Siggraph Course Notes (1983).
4. Gardner, Geoffrey "Simulation of Natural Scenes Using Textured Quadric Surfaces," Computer Graphics Vol. 18 No. 3 (August 1984) pp. 11 - 20
5. Bloomenthal, Jules "Modeling the Mighty Maple," Computer Graphics Vol. 19 No. 3 (July 1985) pp. 305 - 311.
6. Max, Nelson "Atmospheric Illumination and Shadows" Computer Graphics Vol. 20 No. 4 (August 1986) pp. 117 - 124.

7. Max, Nelson "Unified Sun and Sky Illumination for Shadows under Trees," CVGIP: Graphical Models and Image Processing Vol. 53 No. 3 (May 1991) pp. 223 - 230.
8. Max, Nelson "Light Diffusion through Clouds and Haze," Computer Vision, Graphics, and Image Processing Vol. 33 (1986) pp. 280 - 292.
9. Kajiya, James and Brian Von Herzen "Ray Tracing Volume Densities," Computer Graphics Vol. 21 No. 4 (July 1987) pp. 171 - 179.
10. Max, Nelson, Roger Crawfis, and Dean Williams "Visualization for Climate Modeling," IEEE Computer Graphics and Applications Vol. 13 No. 4 (July 1983) pp. 34 - 40.
11. Max, Nelson "Sorting for Polyhedron Compositing," in "Focus on Scientific Visualization," H. Hagen, H. Muller, and G. Nielson, editors, Springer Verlag, Berlin (1993) pp. 259 - 268
12. Max, Nelson and Roger Crawfis "Visualizing Wind Velocities by Advecting Cloud Textures," Proceedings of Visualization '92, A. Kaufman and G. Nielson, editors, IEEE Computer Society, Los Alamitos, CA (1992) pp. 179 - 184.
13. Sabella, Paolo "A Rendering Algorithm for Visualizing Scalar Fields," Computer Graphics Vol. 22 No. 4 (August 1984) pp. 51 - 58.
14. Blinn, James "Light Reflection Functions for Simulation of Clouds and Dusty Surfaces," Computer Graphics Vol. 16 No. 3 (July 1982) pp. 21 - 29.
15. Max, Nelson, Roger Crawfis, and Pat Hanrahan "Area and Volume Coherence for Visualization of 3-D Scalar Fields," Computer Graphics Vol. 24 No. 5 (November 1990) pp. 27 - 33.
16. Mie, Gustav "Optics of Turbid Media," Ann. Physik Vol. 25 No. 3 (1908) pp. 377 - 445.
17. Henyey, G. L. and J. L. Greenstein "Diffuse Radiation in the Galaxy," Astrophysical Journal Vol. 88 (1940) pp. 70 - 73.
18. Max, Nelson "Efficient Light Propagation for Multiple Anisotropic Volume Scattering," preprint UCRL-JC-115654, Lawrence Livermore N ional Laboratory, CA (1994).
19. Patmore "Simulated ultiple Scattering for Cloud Rendering" in "Graphics, D sign, and Visualization: Proceedings of the International Conference on Computer Graphics - ICCG93", S. P. Mudur and S. N. Pattaniak, eds., Elsevier Science Publishers (1993) pp. 29-40.
20. Perlin, Ken "An Image Synthesizer," Computer Graphics Vol. 19 No. 3 (July 1985) pp. 287 - 296.

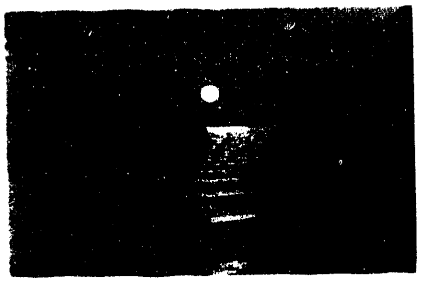
Figure 1

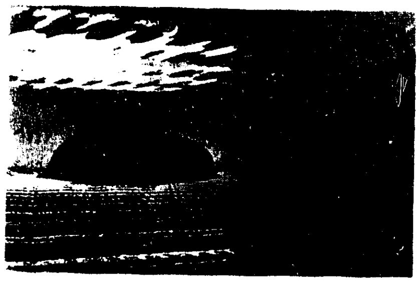
Figure 2

Figure 3

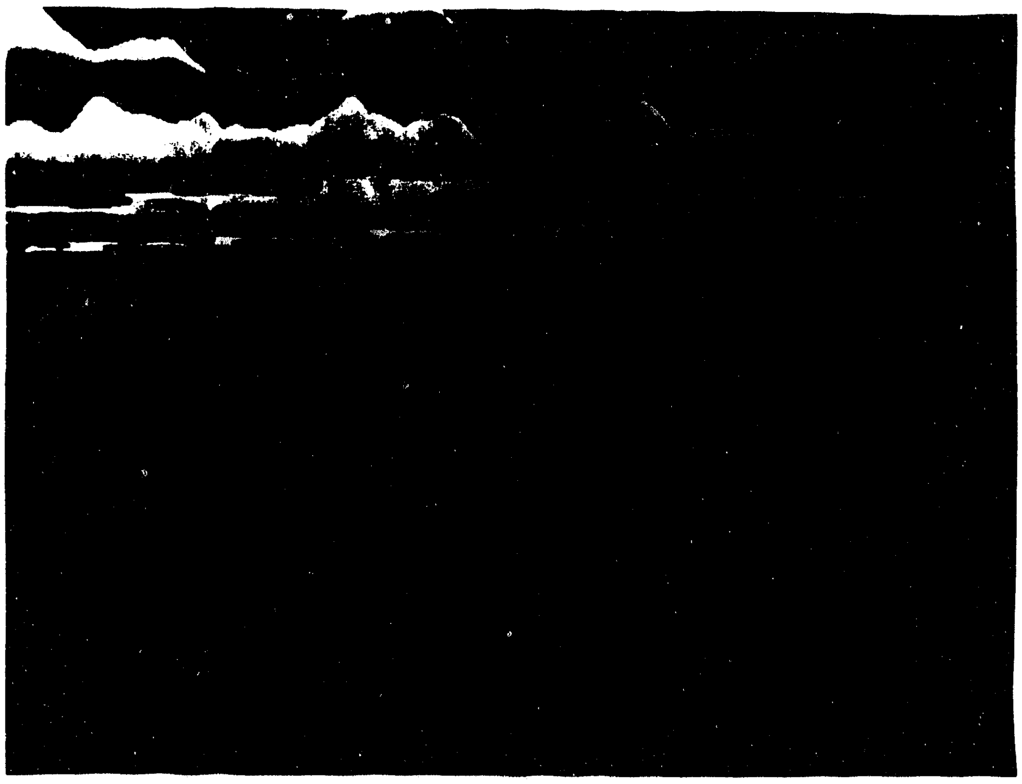
Figure 4

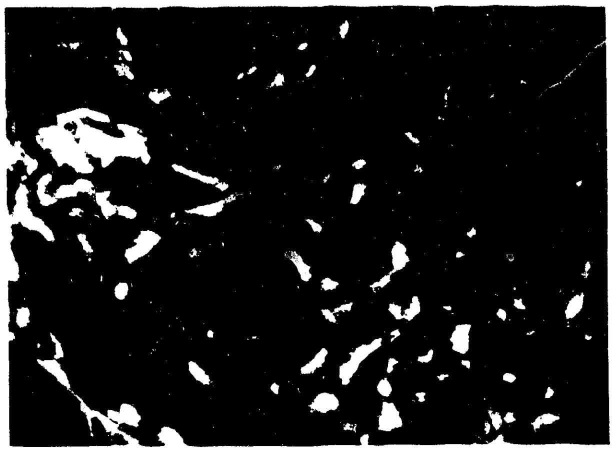
Figure 5

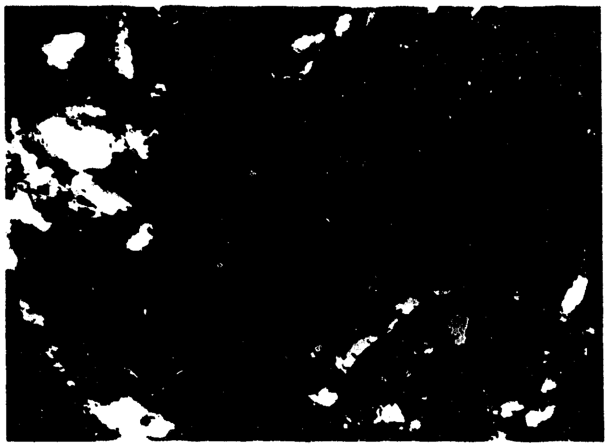
Figure 6

Figure 7

Figure 8

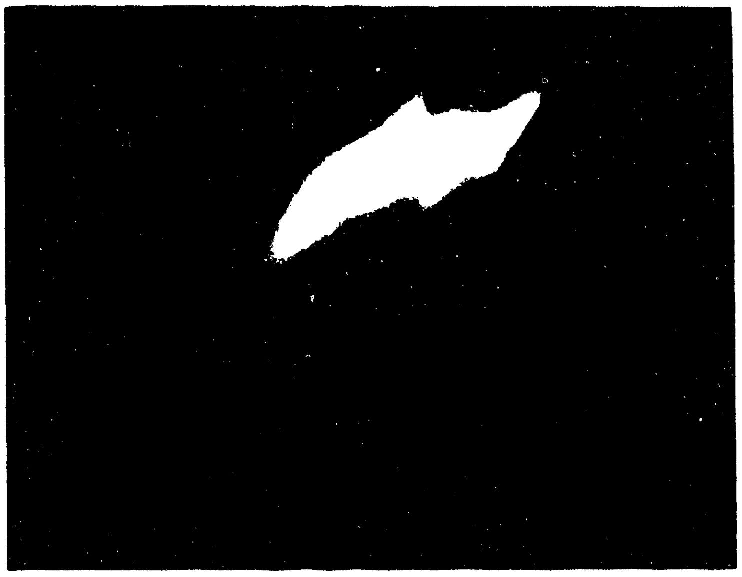

Figure 9

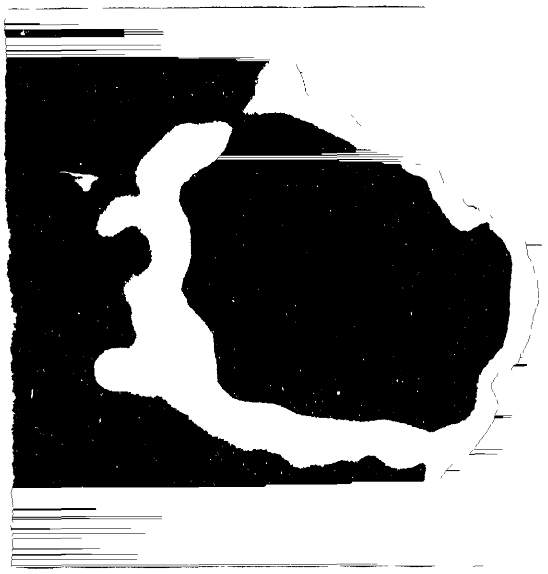

Figure 10

DATE
FILMED
5/12/94

END