Visual Simulation of Lightning

Todd Reed and Brian Wyvill
The University of Calgary[1]

ABSTRACT

A method for rendering lightning using conventional raytracing techniques is discussed. The approach taken is directed at producing aesthetic images for animation, rather than providing a realistic physically based model for rendering. A particle system is used to generate the path of the lightning channel, and subsequently to animate the lightning. A technique, using implicit surfaces, is introduced for illuminating objects struck by lightning.
CR Categories: I.3.3 [**Computer Graphics**]: Picture/Image Generation; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism, *Animation*.
Keywords: Computer Graphics, Lightning, Raytracing, Implicit Surfaces.

1 INTRODUCTION

The synthetic reproduction of natural scenes has been a primary goal of the computer graphics community. Much attention has been given to the problem of rendering plant life, mountainous terrain, clouds, fire, etc. Surprisingly, lightning, one of nature's most spectacular phenomena, has been largely ignored. This paper presents a simple method for modeling and rendering lightning and objects struck by lightning, using raytracing techniques. Lightning is represented as a collection of connected, finite length rays in 3D space. A particle system is used to generate the segments for a complete lightning stroke, and subsequently for animating a lightning strike. For some time we have been working on skeletal-implicit surface modeling techniques and have built a number of models [7]. Such models are represented as an iso-surface in a scalar field. A method is presented for using the scalar field to provide a value for the brightness of the glow around an object that has been struck by lightning.

The best understood source of lightning is cumulonimbus, ordinary thunderstorm clouds. The majority of cloud related lightning discharges are intracloud, and are not visible from earth. Visible types of discharges include cloud-to-cloud, cloud-to-air, ground-to-cloud, and cloud-to-ground[2]. Research on lightning has focused on the familiar cloud-to-ground lightning for the practical reason of minimizing its damaging effect on our environment. This interaction with our environment is also what makes cloud-to-ground lightning so visually stimulating. Photographs of lightning strikes that encounter "ground zero" are among the best[3] (see Figure 9), and provided the inspiration for rendering lightning.

Research in physically based lightning models has been reported (see [2], for example). However, the mathematical complexity and the numerous parameters required to describe the complete lightning environment make these models impractical for the computer artist.

The primary goal of this research is to produce visually realistic images of lightning with minimal complexity and computational cost. Hence, the technique developed in this paper does not attempt to model cloud physics, atmospheric conditions, or the physics of lightning discharges. Photographs of lightning from [4, 5] provided a yardstick for measuring the accuracy of the images produced.

2 A TYPICAL LIGHTNING STRIKE

Even though our model is not physically based, a short discussion of lightning physics is essential to introduce terminology and concepts.

A typical (negative) cloud-to-ground lightning strike is initiated within a cloud that is positively charged in its upper region, and negatively charged in its lower region; sparse positive charges also occupy the lowest region of the cloud (see Figure 1). The *stepped leader*, a negative stream of electrons, propogates toward ground zero in discrete steps. The stepped leader itself is initiated by an unknown process termed the *preliminary breakdown* (a). The descending stepped leader forms a jagged and branching channel, still invisible, that attracts an *upward positive leader* from the ground, thus starting the *attachment process* (b). The meeting of the downward and upward leaders triggers the first *return stroke*, which is responsible for illuminating the channel, and creating thunder (c). The return stroke begins at the point of contact between the downward and upward leader, usually close to the ground. If sufficient charge remains in the cloud, *dart leaders* may travel through the residual channel and create subsequent return strokes (d). A lightning strike will typically have less than five return strokes, but can have many more. Returns strokes subsequent to the first usually do not illuminate branches off the main channel. The entire discharge, termed a *flash*, lasts about half a second.

[1] Department of Computer Science, The University of Calgary, 2500 University Drive NW, Calgary, Alberta, Canada T2N 1N4, E-mail: reedt@cpsc.ucalgary.ca, blob@cpsc.ucalgary.ca

[2] If you happen to be a shuttle astronaut, include cloud-to-space lightning in this list of visible discharges. On April 28, 1990, a cloud-to-space discharge was recorded for the first time by a payload-bay camera aboard the space shuttle. See [6] for details.

[3] See [4] for an excellent collection of lightning photographs.

SIGGRAPH '94, July 24-29, Orlando, Florida

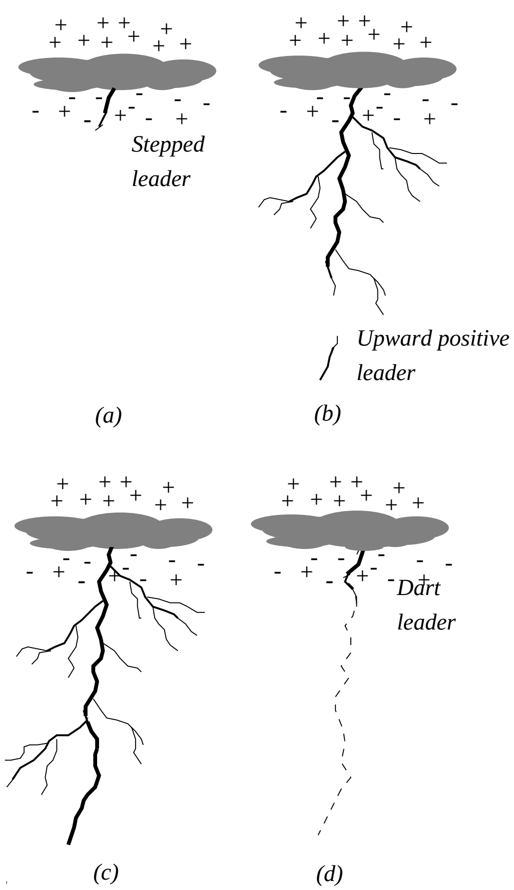

Figure 1: A lightning flash.

The above description is admittedly scant. Readers interested in the physical details of the lightning phenomena are referred to [5].

3 LIGHTNING MODEL

The skeletal structure of lightning is generated by a particle system which simulates the stepped leader progression toward ground zero. Starting with a seed segment (in the clouds), new segments are spawned and randomly rotated about the seed segment. Always rotating children segments with respect to the initial seed segment guarantees the channel assumes a linear shape [4]. The segments are concatenated to form a complete channel. Enough segments are generated so that the main channel hits ground zero. During the leader progression, branches are recursively generated. Branches are allocated a number of segments, chosen from a uniform distribution, and grow until its segments are depleted, or ground zero is encountered.

Uman reports that the directional changes between successive channel segments are randomly distributed, independent of segment length, with a mean absolute value of about 16 degrees [5]. This is simulated by rotating non-seed segments $\pm\alpha$ degrees with respect to the seed segment, where α is chosen from a normal distribution with mean 16 degrees and variance of about 0.1 [5]. In three dimensions, two rotations in orthogonal planes are required. Segment lengths are chosen from a uniform distribution; segment lengths reportedly vary from less than 1 metre to over 1 kilometre. For the purpose of rendering, small segments provide more visually realistic results.

Branching is controlled by a probability function. From the main channel, branching is usually more frequent near the ground. In practice, the erratic behavior of the pseudo-random number generator used[6] made it difficult to consistently control branching. The lightning strokes generated were very sensitive to the seed selected for the number generator. Some seeds completely eliminated branching, while others produced excessive branching (see Figure 2).

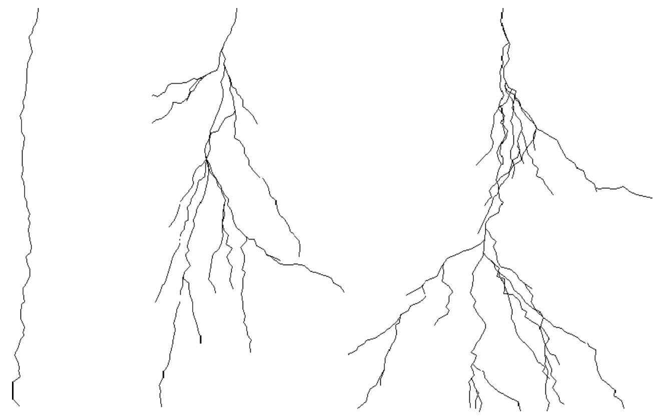

Figure 2: Various lightning skeletons obtained with different seeds for the pseudo-random number generator.

4 RENDERING

Although conventional raytracing techniques are used for rendering, the line-segment representation of lightning prevents us from treating lightning strokes as geometric objects. The usual shading methods employed by raytracers are inappropriate because lightning does not have a definable surface.

The proposed shading method modifies the conventional shading algorithm by adding a color contribution from the lightning. For each ray, r, the following shading calculation is made:

$$I_{total_\lambda} = \sum_i I_{i_\lambda} \quad I_{i_\lambda} = m_{i_\lambda} \exp\left(- \left(\frac{d_i}{w_{i_\lambda}} \right)^{n_{i_\lambda}} \right) \quad (1)$$

where

- λ is the wavelength (*red*, *green* or *blue*).
- I_{i_λ} is the light contribution from segment S_i for wavelength λ.
- $m_{i_\lambda} \epsilon [0, 1]$ is the maximum value of I_{i_λ}
- d_i is the shortest distance between the ray r and the segment S_i.
- $w_{i_\lambda} > 0$ is half the width of the lightning channel segment S_i. If $d_i > w_{i_\lambda}$, then I_{i_λ} is essentially zero.
- $n_{i_\lambda} > 1$ controls the contrast of the lightning channel with respect to the background. Small values of n create fuzzy lightning channels, and large values ($n > 8$) create sharp lightning channels.

[4] Lightning usually obeys the philosophy that the shortest path is the best path, so most channels are linear. Nonetheless, it is not difficult to find photographs showing channels of irregular shape.

[5] The variance was experimentally determined.

[6] The random number generator and distributions used are from GNU's libg++.

Most photographs of lightning show a strong glow surrounding the main channel of the lightning stroke. Reproducing this effect is achieved by a secondary illumination function which is added to equation 1:

$$G_{total_\lambda} = \sum_i G_{i_\lambda} \tag{2}$$

$$G_{i_\lambda} = g_\lambda l_i \exp\left(-\left(\frac{d_i}{W}\right)^2\right) \tag{3}$$

where

- G_{i_λ} is the glow light contribution for wavelength λ.
- g_λ is the maximum value of G_{i_λ}.
- $l_i \epsilon [0,1]$ is a "life" factor that describes the brightness of segment S_i.
- W is half the width of the glow (when $l = 1$).
- d_i is the shortest distance between the ray r and the segment S_i.

The left-most image of Figure 5 was generated with equation 1; the second and third images show the effect of adding a blue and red glow with equation 2 to the lightning channel.

The calculation of d_i [3] for each segment S_i is a computational bottleneck. By generating the lightning in a plane parallel to the view plane, the rendering process can be optimized by simplifying the calculation of d_i. If L is the plane containing the lightning stroke to render, and p is the point where the ray r intersects L, then let d_i be the shortest distance between the segment S_i and the point p. Using this technique, rendering time can be reduced by a factor of 2-5. While a boon to the time conscious artist, this method has the disadvantage of requiring the plane L to be oriented perpendicular to the line of sight. For the purposes of animation, the more general approach is adopted, thus allowing free motion of the camera.

As the particle system proceeds, appropriate values for m, w, n, and l are assigned to each segment. Seed values for m, w, and n are input parameters, and l is automatically initialized to 1.

For the main channel, the width attribute, w, is fixed for all segments, thus maintaining a uniform thickness for the entire channel. To simulate the gradual narrowing of branches, $w = 0.95 w_{predecessor}$ for branch segments. For a primary branch seed segment[7], $w = c w_{parent}$, where $c \epsilon [a, b]$ is chosen from a uniform distribution, and $0 < a < b < 1$. Branches off the main channel are consistently less than half as thick as the main channel, so $b < 0.5$ is appropriate. For branch seeds not attached to the main channel, $w = w_{parent}$.

The "life" factor, l, which controls the glow around a segment, is modified in a similar fashion, except for all branch seeds, $l = c^2 l_{parent}$, where c is defined as above.

All segments share the same values for m and n.

5 MAKING LIGHTNING A LIGHT SOURCE

Employing the method described above produces realistic lightning images which provide a good background for raytracing scenes. However, if other objects surround the lightning channel, then the lightning must behave as a light source. An expensive, but adequate solution is to add line-segment light sources to each segment. The light source for segment S_i is scaled by l_i, so that the main channel contributes more light than the branches. When calculating the light contribution from segment S_i to point p on some object the light intensity is attenuated by the scalar $1/d_i^2$, where d_i is the shortest distance between S_i and p; this attenuation factor prevents the lightning from illuminating distant objects.

[7] A branch off the main channel.

6 ANIMATION

A single lightning flash, lasting about 0.5 seconds, is composed of at least one return stroke, and possibly subsequent return strokes. Each return stroke is identified by a surge of light illuminating the channel. The first return stroke illuminates the main channel and the branches, while subsequent return strokes typically illuminate only the main channel. Numerous return strokes account for the flickering of lightning. This can be simulated by appropriately modulating the attributes l_i and m_i over time.

For animation, additional attributes are assigned to each segment:

- the age of the segment (one unit of time = one frame)
- the branch level (0 for the main channel, 1 for branches off the main channel, etc.)
- the position of the segment within a branch (0 for the first segment, 1 for the next segment, etc.)

The particle system used for generating the segments is extended to account for time. The animation of a lightning strike is done in two distinct phases:

Progression In this phase, the lightning is initiated from a seed segment, and propogates toward ground zero, as described above. Traveling up to 160,000 kilometres per hour, the appearance of a lightning stroke is instantaneous to the human eye. However, for an animation, this gives unsatisfactory results. Recall that a lightning channel is not illuminated until the first return stroke is initiated by the contact of the downward and upward leaders. Rather than replicate this process, we have adopted to illuminate the channel as it progresses toward the ground. This technique, although not physically accurate, is effective for animation because it agrees with people's intuition of how lightning should appear.

Regression The flickering of lightning is a result of multiple return strokes down the main channel (but not the branches). To simulate this effect, a positive, decaying, sinusoidal function is used to calculate the intensity of main-channel segments for each frame. The intensity of branch segments decreases monotonically.

To enhance the visual impression of a lightning stroke, we have doubled the time duration of a typical flash to one second, or 30 frames. Figure 6 shows selected frames from a lightning animation. The first five images correspond to the *progression* stage, and the last image starts the *regression* phase.

7 GLOWING OBJECTS

The visual effects of lightning striking an object depend on a number of factors, such as the shape of the object, the material from which it is made, etc. Since our goal is to provide a reasonable visual impression of lightning, particularly for animation, making an object glow when it is struck is sufficient for our purposes. Since we already have a skeletal implicit surface system, a value for the glow can be obtained directly from the field in which such models are defined. Readers unfamiliar with implicit surface modeling are directed to [1, 7].

Given a scalar field defined by the implicit function F, an implicit surface is defined by the set of points $\{p : F(p) = k\}$, where $k\epsilon(0, 1)$ is a chosen constant. For a non-trivial model, F is the aggregate of several fields, each associated with a primitive skeletal element. The train and coach in Figure 6, for example, are composed of several ellipsoids, each with its own field, F_i. Hence, the surface of the train is defined by the implicit function

$$F(p) = \sum_{i=1}^{N} c_i F_i(p) \tag{4}$$

where N is the number of ellipsoids, and c_i is a scalar value used to adjust the contribution of F_i.

In our system, k is conventionally 0.5. Thus, if $F(p) < 0.5$, p is outside the surface, and if $F(p) > 0.5$, p is inside the surface. The objective is to make the field between the surfaces defined by $\{p : F(p) = 0.5\}$ and $\{p : F(p) = 0.0\}$ glow; the latter surface, the zero-contour, defines the outer extent of the glow.

For each ray r spawned by the raytracer, the closest point on r to each primitive is found, giving N points, $p_1, p_2, \ldots, p_N$. For each point p_i, the implicit value $v_i = F_i(p_i)$ is calculated. The largest v_i, v_{max}, is used to calculate the brightness of the glow as follows:

$$G_\lambda = m_\lambda v' \tag{5}$$

where

$$v' = \begin{cases} 0 & \text{if } v_{max} < 0 \\ k & \text{if } v_{max} > k \\ v_{max} & \text{otherwise} \end{cases} \tag{6}$$

and $m_\lambda \epsilon [0, 1]$ are scalars controlling the intensity of independent wavelengths. If $m_\lambda = g_\lambda$ (from equation 3), then the glow around the implicit surface will coincide with the lightning channel glow, if this is desired.

Figure 7 illustrates the glowing effect on two spherical implicit surfaces blending. The animation frames in Figure 6 show lightning striking the train's coach, causing it to glow.

The advantage of using implicit surface models is that the glow can be calculated directly from the field as shown above. However, the technique can produce undesirable results in certain instances. Problems arise when the zero-contour surface, which defines the edge of the glow, does not faithfully follow the k-contour surface, which defines the model. This problem is especially evident when negative primitives, obtained by setting c_i in equation 4 to a negative value, are used.

8 CONCLUSION

We have presented a workable model of lightning, which produces aesthetic images sufficient for most animation requirements. Our lightning model takes account of the following features of real lightning:

- Accurate shape according to statistical data available.
- The glow around lightning which is often observed.
- Glow around objects that are struck (provided that they are modeled as implicit surface objects).
- Lightning as a light source.

The following points still require further attention:

- Implementing a physically based model and formalizing the branching algorithm used to generate lightning channels.
- Improving the appearance of objects struck by lightning; investigating further how glowing objects can be modeled with implicit surfaces.
- Improving rendering time.

9 ACKNOWLEDGEMENTS

We would like to thank the members of the *GraphicsJungle* research lab at the University of Calgary and the support of the computer science department. We would also like to thank Andrew Glassner of Xerox PARC for his encouragement and advice. Thanks go to all those who have contributed to Rayshade development, particularly Craig Kolb and Rod Bogart, at Yale and Princeton Universities respectively.

This work is partially supported by the Natural Sciences and Engineering Research Council of Canada.

A LIGHTNING SPECIFICATION

Figure 3 shows an example of the new commands added to the Rayshade input language to specify the lightning and glow around implicit surfaces.

The contents of the file `light.zap` are shown in Figure 4. The file `train.soft` is a binary file generated by the implicit surface editing system. This file contains the specifications of the skeletal elements which define the train model.

```
/* Lightning specification */
lightning light.zap

/* Implicit surface description
   for glow */
glow train.soft
```

Figure 3: An example of the new commands added to Rayshade for rendering lightning and glowing objects.

REFERENCES

[1] Jules Bloomenthal and Brian Wyvill. Interactive Techniques for Implicit Modeling. *Computer Graphics*, 24(2):109–116, 1990.

[2] L Dellera and E. Garbagnati. Lightning stroke simulation by means of the leader progression model. *IEEE Transactions on Power Delivery*, 5(4):2009–22, October 1990.

[3] Ronald Goldman. Intersection of Two Lines in Three-Space. *Graphics Gems*, page 304, 1992. Edited by David Krik.

[4] William Newcott. Lightning, nature's high-voltage spectacle. *National Geographic*, 184(1):83–103, July 1993. Photographs by Peter Menzel.

[5] Martin A. Uman. *The Lightning Discharge*. Academic Press, Inc., 1987.

[6] O.H. Vaughan, R. Blackesleee, et al. A cloud-to-space lightning as recorded by the Space Shnuttle payload-bay TV cameras. *Monthy Weather Review*, 120(7):1459–61, July 1992.

[7] Brian Wyvill, Craig McPheeters, and Geoff Wyvill. Animating Soft Objects. *The Visual Computer*, 2(4):235–242, February 1986.

```
LIGHTNING(
  -- seed point
  POINT{830.0,1150.0,3800.0},
  -- random number generator seed
  2,
  -- ground elevation,
  1055.0,
  -- initial value for w parameter
  VECTOR{0.4, 0.4, 0.64},
  -- value for m parameter
  VECTOR{0.7, 1.0, 0.7},
  -- value for n parameter
  VECTOR{10.0, 10.0, 4.0},
  -- colour of glow
  COLOUR{0.1, 0.05, 0.07},
  -- width of glow
  3.84,
  -- angle (in radians) and std. deviation
  -- for tortuosity (twist) of channel
  NORMAL{0.179, 0.1},
  -- length of each segment; uniform
  -- distribution
  UNIFORM{0.1, 0.25},
  -- number of branch segments; discrete
  -- uniform distribution
  DISC_UNIFORM{10,20},
  -- angle of branches wrt
  -- parent segment;
  UNIFORM{0.18, 0.75},
  -- width of branch relative
  -- to parent
  UNIFORM{0.45, 0.6},
  -- branch probability
  0.025,
)
```

Figure 4: Sample input for lightning specification.

Figure 5: Various lightning images; the left-most lightning image was generated without the additional glow.

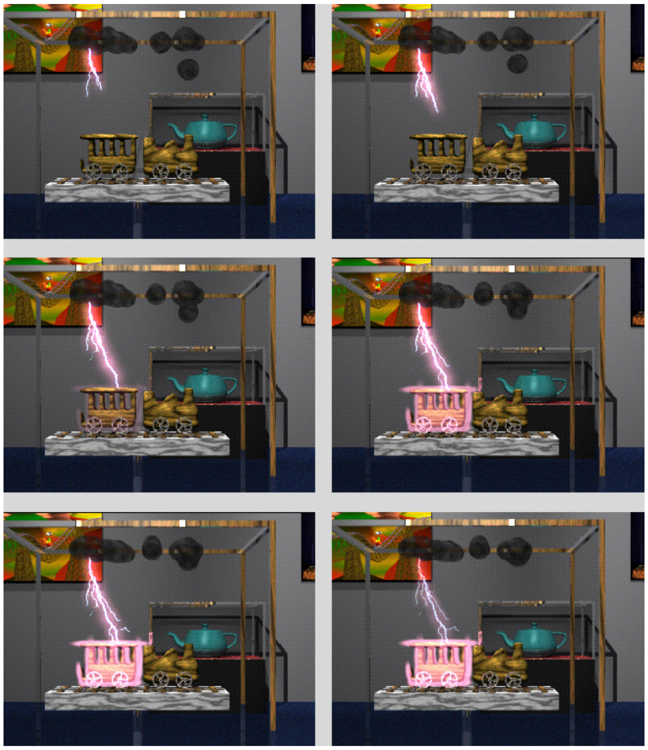

Figure 6: Selected frames from a lightning animation.

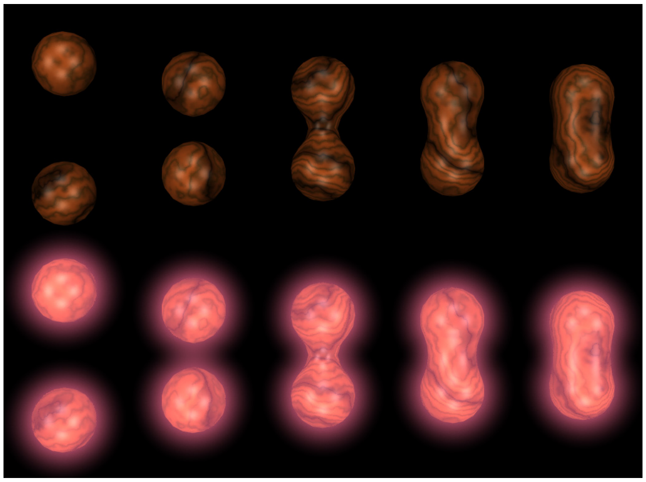

Figure 7: Glowing “soft” objects.

Figure 8: Lightning over a plane of water.

Photo not available.

Figure 9: A real photograph of lightning striking a tree; courtesy of Johnny Autery.