Physically Based Animation and Rendering of Lightning

Theodore Kim Ming C. Lin
Department of Computer Science
University of North Carolina at Chapel Hill
{kim, lin}@cs.unc.edu
http://gamma.cs.unc.edu/LIGHTNING

Abstract

We present a physically-based method for animating and rendering lightning and other electric arcs. For the simulation, we present the dielectric breakdown model*, an elegant formulation of electrical pattern formation. We then extend the model to animate a sustained, 'dancing' electrical arc, by using a simplified Helmholtz equation for propagating electromagnatic waves. For rendering, we use a convolution kernel to produce results competitive with Monte Carlo ray tracing. Lastly, we present user parameters for manipulation of the simulation patterns.*

1. Introduction

The forked tendrils of electrical discharge have a long history as a dramatic tool in the visual effects industry. From the genesis of the monster in the 1931 movie *Frankenstein*, to the lightning from the Emperor's fingers in *Return of the Jedi*, to the demolition of the Coliseum by lightning in last year's *The Core*, lightning is an ubiquitous effect in science fiction and fantasy films.

Despite the popularity of this effect, there has been relatively little research into physically-based modeling of this phenomenon. The existing research is largely empirical, essentially generating a random tree-like structure that qualitatively resembles lightning. The previous work is also limited to brief flashes of lightning, and provides no method for animating a dancing, sustained stream of electricity. However, modeling the fractal geometry of electrical discharge and similar patterns has attracted much attention in physics. To our best knowledge, our algorithm is the first rigorous, physically-based modeling of lightning in computer graphics. We also believe our approach is accurate enough that its applications extend beyond visual effects to more physically demanding applications, such as commercial flight simulation.

Main Contributions: In this paper, we present a physically-based algorithm to simulate lightning, and propose a novel extension for animation of continuous electrical streams. The simulation results are then rendered using an efficient convolution technique. The resulting image quality rivals that of Monte Carlo ray tracing. Lastly, we present user parameters for intuitive manipulation of the simulation. Our approach offers the following:

- A physically-inspired approach based on the *dielectric breakdown model* for electrical discharge;
- A novel animation technique for sustained electrical streams that solves a simplified Helmholtz equation for propagating electromagnetic waves;
- A fast, accurate rendering method that uses a convolution kernel to describe light scattering in participating media;
- A parameterization that enables simple artistic control of the simulation.

Organization: The rest of the paper is organized as follows. A brief survey of related work is presented in Sec. 2. In Sec. 3, we briefly summarize the physics of lightning formation. We present the original dielectric breakdown model as well as our proposed extension in Sec. 4. A efficient rendering method is present in Sec. 5. User parameters are presented in Sec. 6, followed by implementation details and discussion in Sec. 7. Finally, conclusions and possible directions for future work are given in Sec. 8.

2. Previous Work

Reed and Wyvill present a lightning model based on the empirical observation that most lightning branches deviate by an average of 16 degrees from parent branches [14]. A set of randomly rotated line segments are then generated with their angles normally distributed around 16 degrees. In subsequent work, modifications are made to this random line segment model. Glassner [6] performs a second pass on the segments to add “tortuosity”, and Kruszewski [9] replaces the normal distribution with a more easily controlled randomized binary tree.

Notably, Sosorbaram et al. [16] use the dielectric breakdown model (DBM) to guide the growth of a random line segment tree with a local approximation of the potential field. But, their approach does not appear to implement full DBM, as it does not solve the full Laplace equation.

Electric discharges are neither solid, liquid, or gas, but instead are the fourth phase of matter, *plasma*. It is a light source with no resolvable surface, so traditional rendering techniques are not directly applicable. To address this problem, Reed and Wyvill [14] describe a ray tracing extension for both a lightning bolt and its surrounding glow. Alternatively, [16] proposes rendering 3D textures. Dobashi, Yamamoto, and Nishita [4] provide the most rigorous treatment of the problem by first presenting the associated volume rendering integral, and then presenting an efficient, approximate solution.

In electrical engineering, there are three popular models of electric discharge: gas dynamics [5], electromagnetics [1], and distributed circuits [2]. However, none of these are directly applicable to visual simulation, as they respectively approximate the electricity as a cylinder of plasma, a thin antenna, and two plates in a circuit.

3. The Physics of Electric Discharge

We classify the physics literature into two categories. The first deals with the physical, experimentally observed properties of lightning and related electrical patterns. A good survey of this approach is given by Rakov and Uman [13]. The second is a more qualitative approach that characterizes the geometric, fractal properties of electric discharge. A good survey of this approach is given by Vicsek [17].

3.1. Physical Properties

Electrical discharge occurs when a large charge difference exists between two objects. For lightning, the case is usually that the bottom of a cloud has a strong negative charge and the ground possesses a relatively positive charge. Electrons possess negative charge, the charge difference is then equalized when electrons are transferred from the cloud to the ground in the form of lightning. This case is referred to as ‘downward negative lightning’. While other types can exist, downward negative lightning accounts for 90 percent of all cloud-to-ground lightning. For illustrative purposes, we will show here how to simulate this most common type of lightning. But, it should be noted that we can handle the other types of lightning by trivially manipulating the charge configuration.

Lightning is actually composed of several bolts, or ‘strokes’ in rapid succession. The first stroke is referred to as the *stepped leader*. The subsequent strokes, called *dart leaders*, tend to follow the general path of the previous leaders, and do not exhibit as much branching as the stepped leader. We note that the random line segment approach of previous work in computer graphics does not provide a clear method of simulating dart leaders. But, such a method is crucial for simulating sustained electric arcs, which are essentially stepped leaders followed by a large number of dart leaders.

Lightning is initiated in clouds by an event known as the *initial breakdown*. During the initial breakdown, the conductivity in a small column of air jumps several orders of magnitude, effectively transforming the column from an insulator (or *dielectric*) to a conductor. Charge then flows into the newly conductive air. Another breakdown then occurs somewhere along the perimeter of the newly charged air. This chain of events repeats, forming a thin, tortuous path through the air, until the charge reaches the ground.

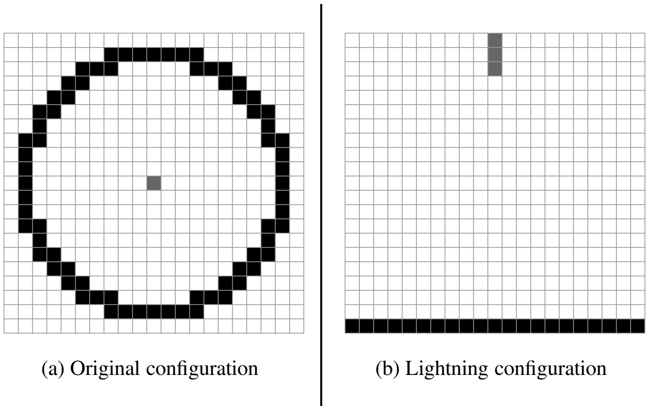
3.2. Geometric Properties

The physical processes that give rise to the breakdown are still not well understood. However, a great deal of progress has been made in characterizing the geometric shape that the breakdown ultimately produces. Electric discharge has been observed to have a fractal dimension of approximately 1.7 [11]. Many disparate natural phenomena share this same fractal dimension, including ice crystals, lichen, and fracture patterns. Collectively, all the patterns that share these fractal properties are known as *Laplacian growth* phenomena.

There are three techniques for simulating Laplacian growth: Diffusion Limited Aggregation [18], the Dielectric Breakdown Model [11], and Hastings-Levitov conformal mapping [8]. All three produce qualitatively similar results. We elect to use the Dielectric Breakdown Model here because it gives the closest correspondence to the physical system being simulated and allows the addition of natural, physically intuitive user controls.

4. The Dielectric Breakdown Model

The Dielectric Breakdown Model, or DBM, was first described by Niemeyer, Pietronero, and Wiesmann [11], and is also sometimes referred to as the η model. We first present the model described in the original paper, and then propose a modification to simulate dart leaders and sustained electric arcs.



(a) Original configuration

(b) Lightning configuration

Figure 1. Different charge configurations for simulation. Grey: $\phi = 0$; Black: $\phi = 1$

4.1. The Laplacian Growth Model

The original charge configuration from [11] is shown in Figure 1(a). Over a 2D grid, the quantity ϕ, the electrical potential at each point, is tracked. First, a negative charge is placed at the center by setting $\phi = 0$ at the center grid cell. Then, a circle of positive charge is constructed around the center charge by setting a surrounding circle to $\phi = 1$. The potential at the remaining grid cells are then set by solving the Laplace equation (Eqn. 1) over the grid, with the center charge and the surrounding circle treated as boundary conditions. The grid boundaries are also set to $\phi = 0$.

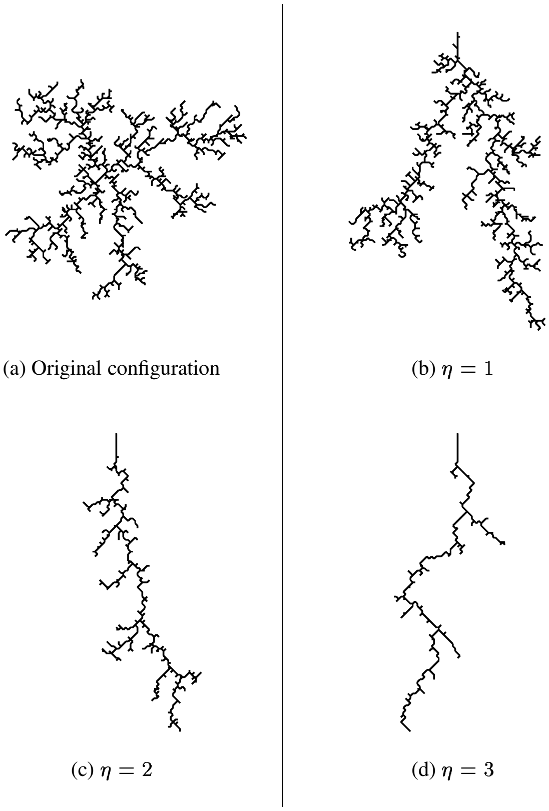
$$\nabla^2 \phi = 0. \tag{1}$$

The Laplace equation produces a linear system that must then be solved. For information on solving the Laplace equation and the related Poisson equation, the reader is referred to [3]. In our implementation, we solved the system using conjugate gradient with a diagonal preconditioner [15]. Once the Laplace equation has been solved, we construct a list of all the grid cells that are adjacent to a negative charge ($\phi = 0$). One of these grid cells is then randomly chosen as a growth site (i.e. the site of the next breakdown). The chosen cell is set to $\phi = 0$ and is treated as part of the boundary condition in subsequent iterations. The probability of a grid cell being chosen is weighted according to its potential. The weight function is given in Eqn 2.

$$p_i = \frac{(\phi_i)^\eta}{\sum_{j=1}^{n} (\phi_j)^\eta} \tag{2}$$

where i is a cell in the list of adjacent cells, and n is the total number of cells in the list. The η term is a user parameter that will be discussed in section 6.

Subsequent iterations proceed by solving the Laplace equation again over the 2D domain, and again selecting a growth site according to Eqn 2. The iterations are repeated until the user obtains the desired results. The technique generalizes trivially to three dimensions by simply solving the 3D Laplace equation.



(a) Original configuration

(b) $\eta = 1$

(c) $\eta = 2$

(d) $\eta = 3$

Figure 2. Simulation results from different charge configurations. 2(a) is result of configuration from Figure 1(a). 2(b) - 2(d) are configurations from 1(b) with various η.

The classic configuration produces a radial discharge, as shown in Figure 2(a). In order to produce lightning-like patterns, we instead use the initial configuration shown in Figure 1(b). We start with a small amount of negative charge at the top of the 3D domain, representing an initial branch of lightning. The bottom edge of the domain represents the ground, and is thus set to positive charge. The remaining grid edges are again set to $\phi = 0$. The results of running the simulation on this initial configuration with different η are shown in Figures 2(b) - 2(d).

4.2. A Poisson Growth Model

Once we have formed an initial stepped leader, we would like to have a method for generating subsequent dart leaders that follow the same general path. Since the path changes slightly with each successive dart leader, a large number of dart leaders will produce the 'dancing' effect present in a sustained electric arc.

We hypothesize that the reason that a dart leader follows the same general path as a stepped leader is because there exists residual positive charge along the old leader channel that attracts the new dart leader. In order to simulate this behavior, we need a method of introducing residual charge into the simulation.

While DBM can simulate many different kinds of natural phenomena, we observe that for the case of electricity, the Laplace equation can be viewed as a special case of the Helmholtz equation for propagating electromagnetic waves (Eqn 3).

$$\left(\nabla^2 + \left(\frac{\omega}{c}\right)^2\right)\phi = -4\pi\rho \tag{3}$$

where ω is angular velocity, c is the speed of light, and ρ is charge density. The Helmholtz equation is derived directly from the Maxwell equations for electricity and magnetism, so it provides a clean connection between fractal growth and classical physics. The Laplace equation can be viewed as the case where the charge density is equal to zero and the relativistic $\left(\frac{\omega}{c}\right)^2$ term is ignored. As lightning bolts have a linear velocity that already approaches the speed of light, the angular component should be negligible. So, if we continue to ignore the relativistic term but re-introduce the charge density term, the electromagnetic Poisson equation is obtained:

$$\nabla^2\phi = -4\pi\rho. \tag{4}$$

If we now solve this equation in place of the Laplace equation, we can produce the desired dart leader behavior. The value of ρ is determined by a second grid of values in space that is initially set to zero. This essentially reduces Eqn. 4 to the Laplace equation for the initial iteration. After we generate our first bolt, we deposit charge along the leader channel by setting ρ in the cells along the channel to a positive value. When generating subsequent bolts, the new ρ values will automatically attract the new bolt to the old path. After each new bolt is generated, we clear the previous ρ field and repopulate it with charges along the new leader channel.

Fortunately, because the Poisson and Laplace equations are very similar, the only implementation overhead required for our modified model is a minor change to the residual calculation in the conjugate gradient solver. It is worth noting that a similar model has been proposed in the physics literature [12] which also accounts for inhomogeneous dielectric permittivities. Our model was developed independently. For efficient visual rendering, we choose to ignore inhomogeneity and treat air as a homogeneous media.

5. Rendering

For the rendering of electricity, we borrow the method of Narasimhan and Nayar [10]. In the paper, analytical models are obtained that reduce the rendering of certain types of participating media to a 2D convolution. The results are competitive with expensive Monte Carlo techniques such as photon mapping, but run in seconds instead of hours. We will first summarize the pertinent formulae from [10], then describe how we use it to generate a convolution kernel, and finally show how we render electricity.

5.1. Atmospheric Point Spread Function

The convolution kernel produced by the method of [10] is called an Atmospheric Point Spread Function, hereon referred to as an APSF. The APSF is a series expansion of the Henyey-Greenstein phase function, a popular function for describing the scattering of light in participating media. The basis functions used are Legendre polynomials, whose series form are shown in Eqn. 5.

$$L_i(x) = \frac{((2i-1)\cdot x \cdot L_{i-1}(x) - ((i-1)\cdot L_{i-2}(x))}{i} \tag{5}$$

In order the evaluate the series, the following base cases are also necessary: $L_0(x) = 1$, $L_1(x) = x$. The full APSF, $I(T,\mu)$, is then given in Eqn. 6.

$$I(T,\mu) = \sum_{m=0}^{\infty} \left(g_m(T) + g_{m+1}(T)L_m(\mu)\right) \tag{6}$$

where

$$\begin{aligned} g_m(T) &= e^{-\beta_m T - \alpha_m T} && (7)\\ \alpha_m &= m+1 && (8)\\ \beta_m &= \frac{2m+1}{m}(1-q^{m-1}). && (9) \end{aligned}$$

Again a base case is necessary: $g_0(T) = 0$. The variable q is the scattering parameter from the Henyey-Greenstein phase function. Increasing q from 0 to 1 increases the density of the medium, and can be thought of as transitioning the weather from clear skies to rain. The *optical thickness*, T, is equal to $R\sigma$, where R is the radial distance from the viewer, and σ is the extinction coefficient of air. Finally μ is the cosine of the radial direction θ from the source.

5.2. Generating a Convolution Kernel

The APSF is a three dimensional function that describes how much light is reaching any point in space around a point light source. If we can determine how a single point

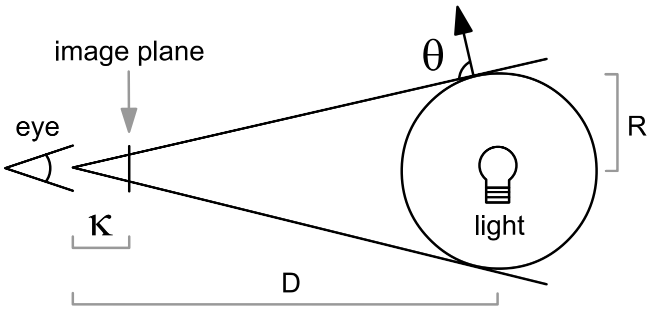


Figure 3. Pinhole camera geometry for generating APSF kernel

light spreads out on the image plane, we can then use this point spread function as a convolution kernel to render a light source of arbitrary shape.

Assume we want to generate an $n \times n$ resolution convolution kernel of physical size $M \times M$. We sample the APSF according to the geometry in Figure 3. In this figure, we assume a pinhole camera model where $\kappa = 0.025$ meters, about the width of an eyeball. We also assume the lightning stroke is two kilometers away: $D = 2000$. We treat R as a user parameter that allows control over the width of the 'glow' around the stroke.

In order to compute the value $I(T, \mu)$ at each point on the kernel, we need to determine a value μ at each sample. If we assume the point light source projects onto the center of the kernel, the μ value at kernel sample (x, y) follows by trigonometry (Eqns. 10 - 12).

$$\nu_{x,y} = \sqrt{\frac{(x-\frac{n}{2})\times M}{n} + \frac{(y-\frac{n}{2})\times M}{n}} \tag{10}$$

$$\omega_{x,y} = \frac{\kappa^2 D - \kappa\sqrt{-\nu_{x,y}^2 D^2 + \kappa^2 R^2 + \nu_{x,y}^2 R^2}}{\kappa^2 + \nu_{x,y}^2} \tag{11}$$

$$\mu_{x,y} = \pi - \tan^{-1}\frac{\kappa}{\nu_{x,y}} - \sin^{-1}\frac{D-\omega_{x,y}}{R} \tag{12}$$

If $\frac{\nu_{x,y}}{\kappa} > \frac{R}{D}$, then we are outside the desired width of the glow, and the kernel value should be set to zero. The APSF only drops off to zero at infinity, so in practice, the smallest non-zero value of the kernel must be subtracted from the other non-zero values of the kernel to prevent the silhouette of the kernel from appearing in the final image.

We are making a simplifying assumption here that all parts of the lightning bolt are exactly two kilometers away. While this is not strictly true, unless the bolt spans a very large physical domain, we believe it is a reasonable approximation. If the user would like to perform a fly-by of the bolt, the distance constantly changes, and several different depth-dependent kernels must be computed. However, even in this case, the time required to generate several different kernels is still orders of magnitude less than using a Monte Carlo renderer.

5.3. Rendering Electricity

Even for large electric discharges like lightning, the plasma channel is only several centimeters in diameter [13]. We hypothesize that humans perceive that the stroke is thicker because the the brighter portions of glow exceed the range perceptible by the human visual system, so they bleach together into what looks like a thicker bolt.

With this hypothesis in mind, we model the plasma channel as a series of thin line segments. We then apply the APSF kernel to a 2D rendering of these line segments to simulate the glow. If the brightness of the plasma channel is set correctly, the APSF should produce luminance values that exceed the range of the display device, creating the expected thick bolt. In this way, we can remain physically consistent while avoiding the need for a complex geometric representation of plasma.

We proceed in three stages. First, we construct a graph from the simulation. We then assign different luminance values to each graph edge, as some parts of a lightning stroke are brighter than others. Finally, we render the graph edges as line segments and apply the APSF.

5.3.1. Constructing the Graph We observe that the construction of the lightning stroke can also be seen as the construction of a directed tree. The root of the tree is represented by the initial discharge from the beginning of the simulation. When a grid cell is added to the lightning stroke, we create a corresponding graph node, and then search the cell neighbors for one that is already on the stroke. Such a neighbor must exist, as it is a necessary condition for the grid cell to have been selected as a growth site. This neighbor is then set as the parent node, and the newly added grid cell is recorded as the child. When a grid cell adjacent to the ground is added, we halt the simulation. In nature, growth would end at this point because the charge now has a direct conduit to the ground.

5.3.2. Assigning Wattage With our tree, we can now assign a separate luminance value to each line segment. We divide the line segments into three classes: the main channel, secondary channels, and side channels. The majority of the charge flows through the main channel, so it should be brightest. By inspection of photographs, it is clear that there are dimmer but distinct secondary channels in most strokes, and branching off from the secondary channels, barely perceptible side channels. Locating the main channel is straightforward. The node corresponding to the grid cell that hit the ground, along with all its ancestors, constitute the main channel.

Locating the secondary and side channels is more involved. Every node adjacent to the main channel that is not on the main channel forms the root to a new tree. Within each such tree, the charge selects a preferred path that becomes the bright secondary channel. There is a poverty of theories on how this path is selected; perhaps the path that had the largest potential differences during the breakdown process is selected. For aesthetic effect, we set the path with the greatest number of nodes as the secondary channel. Off of this longest secondary channel, we also add other 'long' paths according to a user-defined cutoff. This technique maximizes the length of the dramatic, snaking tendrils that surround the central channel. All the remaining edges are now considered to be side channels.

We must now assign a wattage to each edge. While there exists some data on the wattage of the main channel (Between 1.3×10^6 Watts / m and 3.9×10^5 Watts / m according to [13]), we have been unable to find data on the wattage of secondary or side channels. We have attempted to estimate the wattages by deconvolving photographs of lightning, but this method requires a high dynamic range image of lightning that can resolve the bleached portion of the stroke, as well as the APSF values corresponding to the scene. We instead used heuristic values that brought us into close qualitative agreement with photographs.

We rendered the line segments and convolved them with the APSF settings given in Table 1. The resulting image was then composited into a raytraced rendering of the remaining scene objects. We do not set the main channel to the wattage given by [13], because in the absence of tone mapping, this step would bleach the entire scene. The application of tone mapping to lightning rendering is discussed in our future work.

Figure	n	M	q	m	T	R
4, 5, 7	256	1.0	0.99	200	1.001	200
6	64	1.0	0.9	200	1.1	100

Table 1. *APSF settings used:* m corresponds to the number of terms used in the Legendre series.

6. User Controls

Our modified DBM permits user control through four parameters: an η variable to control the 'branchiness' of the stream, a charge density field ρ to control the path of the stream, a boundary condition to repel the stream, and an overall charge configuration to control where the stroke begins and ends.

The effect of the η variable in Eqn. 2 can be seen in Figure 2(b) - 2(d). At $\eta = 1$, dense branching is observed. As η increases, the density of the branching decreases. Hastings observes that at $\eta = 4$, the stream transitions into a non-fractal, one-dimensional curve [7]. So, the domain of the η parameter is effectively in the range of $(1, 4)$. A physical interpretation of η is not entirely clear, it can perhaps be viewed as the amount of resistance that the air offers to the process of dielectric breakdown.

As ρ is a 2D field representing the image plane, the user can 'paint' into it any desired charge distribution. The lightning stroke will then be attracted to this painted path as described in Section 4.2.

In addition to attracting the electric arc, the user may want to repel the arc from certain regions. For instance, there may be an obstacle in the scene that the user does not want the arc to intersect. This effect can be achieved by setting the interior of the obstacle to $\phi = 0$. This sets the charge of the object to the same charge as the arc, causing the obstacle to repel the arc. However, we must then be careful in our implementation not to add grid cells adjacent to the obstacle to the list of candidate growth sites in Eqn. 2.

Finally, we have only shown two charge configurations: the circle in Figure 1(a), and the lightning configuration in Figure 1(b). However, arbitrary charge configurations also produce electric arcs. The arc can begin from any arbitrarily shaped negative region, and terminate at a positive object. In this way, it is possible to construct an arc between any two objects in an arbitrary scene.

7. Implementation and Results

We have implemented our algorithm in C++. We ran simulations for several scenes on a 2.66 GHz Xeon processor. Unless otherwise noted, all simulations were performed on a 512×512 grid with $\eta = 1$ and $\rho = \frac{1000}{4\pi}$ along the main channel. The renderings were performed in POV-Ray, and then convolved and composited using ImageMagick. Although we set $\eta = 1$, the final results tend to resemble those where $\eta = 2$ and 3. This is because the majority of the growth sites are treated as side channels, and are thus very dim. However, we found that in order to obtain long, dramatic, secondary channels, setting $\eta = 1$ was necessary.

Note that when implementing Eqn. 5, recursively evaluating the series is an exponential time operation. However, evaluating from the bottom up (i.e. in the order $L_0(x), L_1(x), L_2(x)$ ...) is a dynamic programming solution that can be done in linear time. Using this method is more efficient. Also, as the convolution kernel in subsection 5.2 is separable, it can be performed quickly with two $n \times 1$ filters instead of one $n \times n$ filter.

In Figure 4, we demonstrate how the user can repel the bolt from arbitrary objects. The lightning must start from the top of the Cornell Box and find a path to the floor, while avoiding the two beams in the center. In Figure 5, we demonstrate how the user can attract the bolt to an arbitrary object. The magenta electrode in the center is set to a negative charge, and blue ball is set to a positive charge. As the blue ball moves, the electric arc follows. In Figure 7, we animate a dancing electric arc between two electrodes. In Figure 6, we validate our results by comparing our renderings with a photograph. The scene was simulated on a 256^3 grid.

8. Conclusion and Future Work

We have presented a physically based algorithm for the simulation, animation, and rendering of sustained electric arcs. We believe that our approach is the most rigorous, physically consistent method available up to date. However, there are several areas for refinement.

Primarily, the simulation can be very slow. For large 2D and 3D grids, the computation time can take hours. But, it is unclear if other Laplacian growth methods, such as DLA or Hastings-Levitov conformal mapping, can give superior performance while preserving the same level of control.

While our rendering method is physically consistent, it would be more realistic to use some sort of tone mapping operator to bring the luminance values back into the range of the display device. No operator was used here because we were unsure which would be appropriate. In the tone mapping literature, a ‘bright’ object is usually daylight or a lightbulb, so it is unclear if some of these methods would break down in the presence of luminance values many orders of magnitude brighter.

While the use of the convolution kernel generates impressive results, there are still some unresolved issues. It assumes the scattering medium is homogeneous, so it does not explicitly handle the effects of either internal obstacles or clouds. A scene requiring a volume caustic still needs a Monte Carlo renderer. The approach described in [4] appears to be the best solution for a scene containing clouds. While an analytical solution may also be possible for these cases, one has not yet been found.

Finally, we have only presented one type of Laplacian growth: electric arcs. Laplacian growth encompasses many disparate phenomena, including ice formation, material fracture, lichen growth, tree growth, liquid surface tension, vasculature patterns, river formation, and even urban sprawl. Modeling of Laplacian growth is well worth exploring for visual simulation of natural phenomena.

Acknowledgements

The authors would like to thank Srinivasa Narasimhan for his help with the APSF, and the anonymous reviewers for their help in improving this manuscript. The photo in Figure 6 is ©Warren Faidley/Weatherstock.com and is used with permission. This work was supported in part by Army Research Office, Intel Corporation, National Science Foundation, and Office of Naval Research.

References

[1] Y. Baba and M. Ishii. Numerical electromagnetic field analysis of lightning current in tall structures. *IEEE Trans. Pow. Rel.*, 16:324–328, 2001.

[2] C. Baum and L. Baker. Analytic return-stroke transmission-line model. *Electromagnetics*, 7:205–228, 1987.

[3] J. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.

[4] Y. Dobashi, T. Yamamoto, and T. Nishita. Efficient rendering of lightning tacking into account scattering effects due to clouds and atmospheric particles. *Proc. of Pacific Graphics*, 2001.

[5] E. Dubovoy, M. Mikhailov, A. Ogonkov, and V. Pryazhinsky. Measurement and numerical modeling of radio sounding reflection from a lightning channel. *J. Geophys. Res.*, 100:1497–1502, 1995.

[6] A. Glassner. The digital ceraunoscope: Synthetic thunder and lightning. Technical Report MSR-TR-99-17, Microsoft Research, 1999.

[7] M. Hastings. Fractal to nonfractal phase transition in the dielectric breakdown model. *Physical Review Letters*, 87(17), 2001.

[8] M. Hastings and L. Levitov. Laplacian growth as one-dimensional turbulence. *Physica D*, 116:244–252, 1998.

[9] P. Kruszewski. A probabilistic technique for the synthetic imagery of lightning. *Computers and Graphics*, 1999.

[10] S. Narasimhan and S. Nayar. Shedding light on the weather. *Proceedings of IEEE CVPR*, 2003.

[11] L. Niemeyer, L. Pietronero, and H. J. Wiesmann. Fractal dimension of dielectric breakdown. *Physical Review Letters*, 52:1033–1036, 1984.

[12] M. Noskov, V. Kukhta, and V. Lopatin. Simulation of the electrical discharge development in inhomogeneous insulators. *Journal of Physics D*, 28:1187–1194, 1995.

[13] V. Rakov and M. Uman. *Lightning: Physics and Effects*. Cambridge University Press, 2003.

[14] T. Reed and B. Wyvill. Visual simulation of lightning. *Proc. of SIGGRAPH*, 1994.

[15] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, Carnegie Mellon University, 1994.

[16] B. Sosorbaram, T. Fujimoto, K. Muraoka, and N. Chiba. Visual simulation of lightning taking into account cloud growth. *Computer Graphics International*, 2001.

[17] T. Vicsek. *Fractal Growth Phenomena*. World Scientific, 1992.

[18] T. Witten and L. Sander. Diffusion-limited aggregation, a kinetic critical phenomenon. *Physical Review Letters*, 47(19):pp. 1400–1403, 1981.

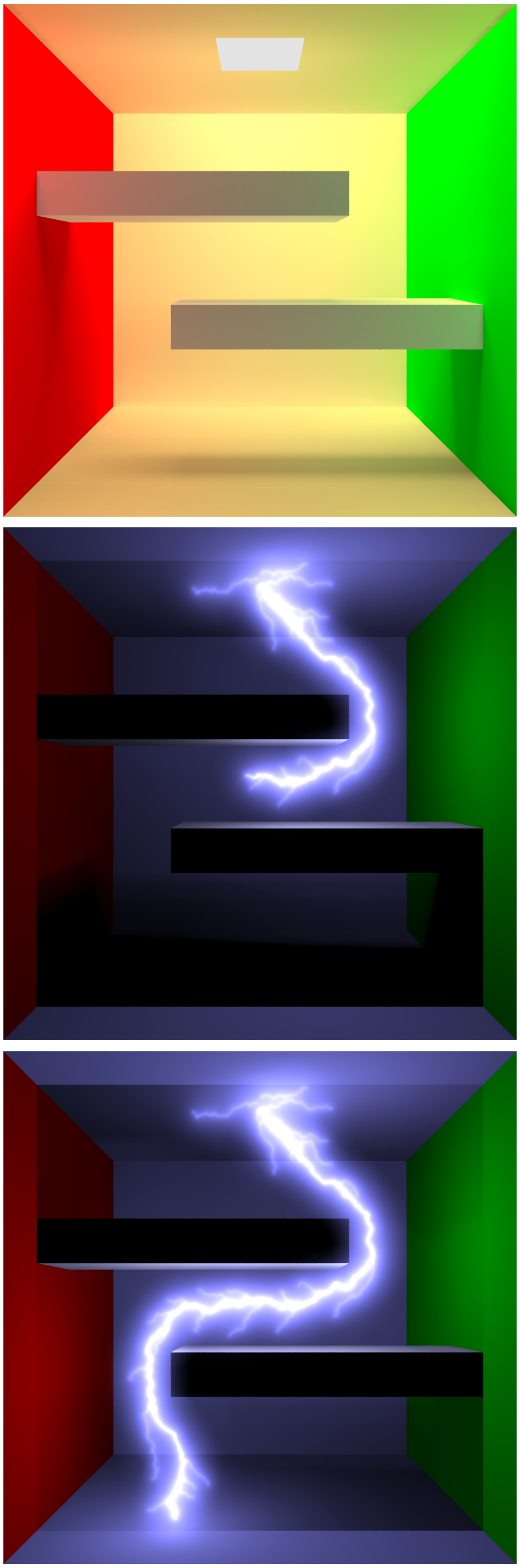


Figure 4. Lightning dodging obstacles in a Cornell Box. *Top to bottom:* The Cornell Box setup; Lightning dodging the first obstacle; Lightning dodging the second obstacle

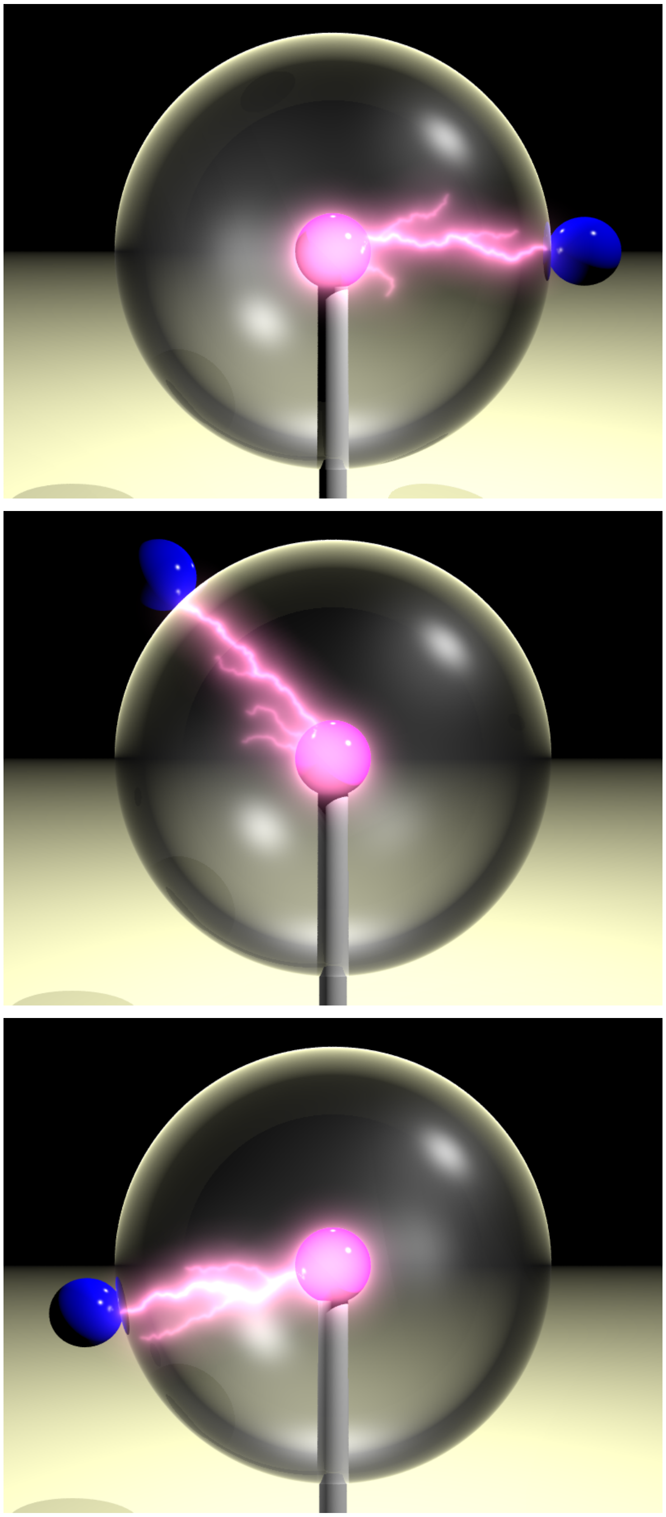
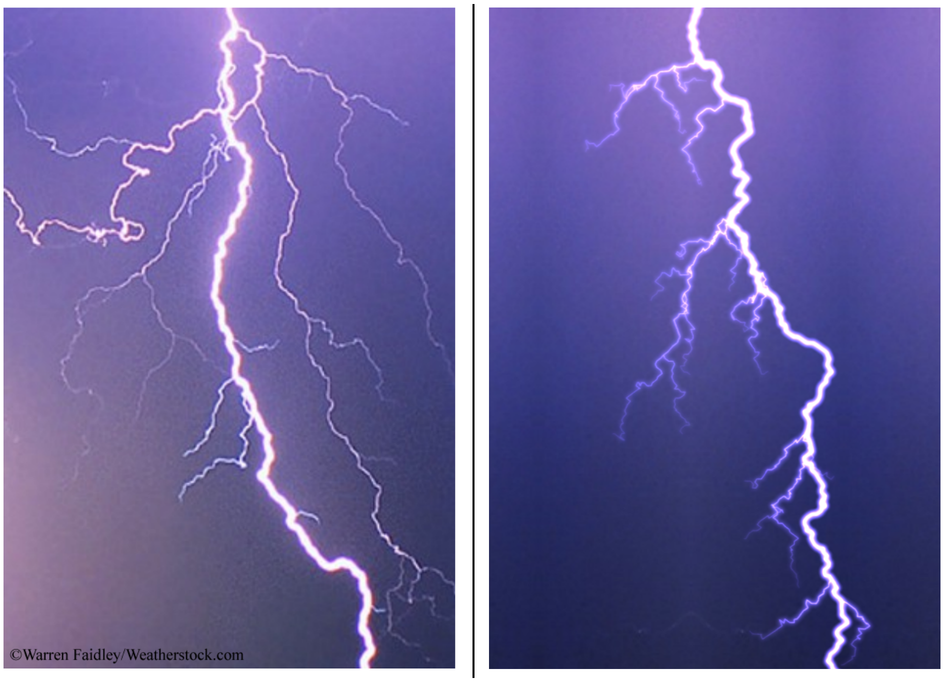


Figure 5. Lightning following a blue ball. The magenta electrode is set to negative charge, and the blue ball to positive charge. As the blue ball moves, the arc follows.



©Warren Faidley/Weatherstock.com

Figure 6. Validation *Left:* Photograph *Right:* Rendering

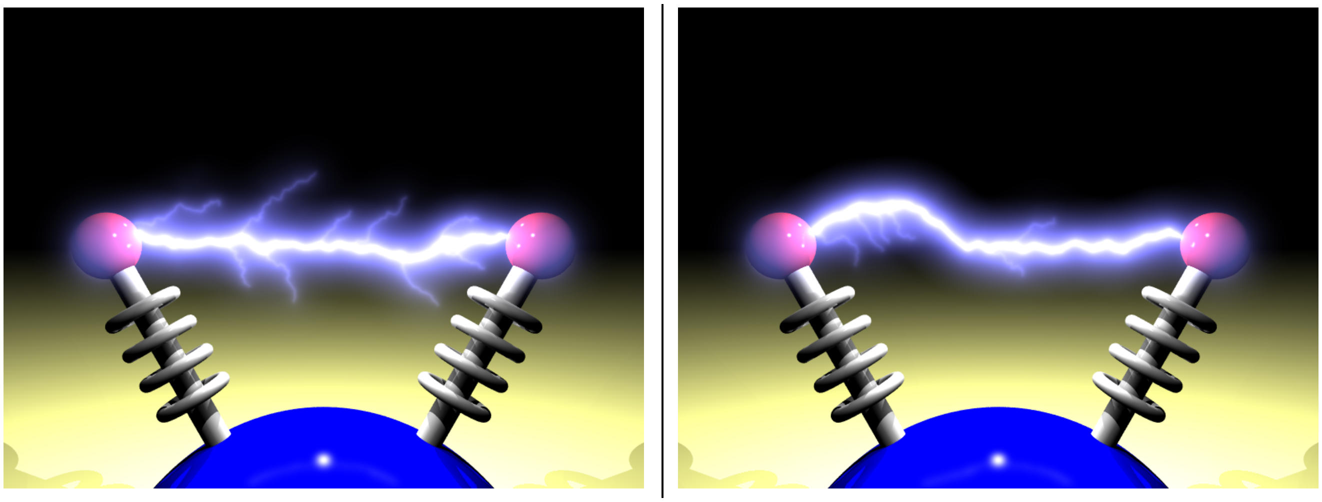


Figure 7. Electric arc leaping between two electrodes