

CGを利用した落雷のビジュアルシミュレーションについての研究

2001MT064 水越 健悟

2002MT005 伴 文明

2002MT085 谷山 隼

指導教員 金 知俊

1 はじめに

近年,CGによる映像が多く見られる様になってきた.CGを使う用途として,架空映像の作成や実写映像への特殊効果があり,代表例としてアニメーション,映画,景観シミュレーション,CADなど多くの例がある.そのため多くの分野でCGが用いられ,その技術の進歩がますます重要なテーマとなっている.特に炎,水,煙などの自然物・自然現象のCGによる再現は様々な分野において必要とされるため,これらの研究は多く行われている.これらの自然現象はCGによる再現の重要性が高いこと,また,その中の1つである雷は非常に短い時間で発生するため認識が難しく,これを可視化することの有用性は高いのではないかと考え,雷のビジュアルシミュレーションを行うこととした.

なお,水越は主にモデル作成を,伴,谷山は主にプログラム作成を担当した.

2 落雷のメカニズム

2.1 雲放電と対地放電

落雷とは上昇気流の発生に伴い,雲の中で発生した氷晶やあられの粒が摩擦して帯電し,一定の量に達した時に放電する現象であり,この放電は雲放電と対地放電(落雷)の二つに分けることができる.私たちが普段目にするのは対地放電であり,以下に典型的な雲から大地への落雷の過程を示す.

2.2 落雷の過程

1. 初期放電 (preliminary discharge)

初期放電とは後述する前駆放電が発生する前に雲内において空気の絶縁破壊が起こる放電であり,この初期放電によって雲内の負電荷領域に電荷の移動を可能にした放電路が形成される(図1.a).

2. 先駆放電 (stepped leader)

初期放電により負電荷が地表に誘導された正電荷に移動しようとし,雲から電流を供給された放電路が空気の絶縁を破壊しながらリーダとなって進展する.これを先駆放電(ステップリーダ)といい,この先駆放電は非常に速い速度で停止,下降を繰り返しながら地表に向かって進展していく(図1.b).

3. 帰還雷撃 (return stroke)

ステップリーダが地表へ到達すると,地表から雷鳴と雷光を伴った放電が同じ経路を光速の

1/3程度の速さで逆方向に進展し,放電路に蓄えられた電荷を中和する現象が起きる.これを帰還雷撃(リターンストローク)といい,また,リーダの下降とリターンストロークの上昇による組み合わせを雷撃という(図1.c).

4. ダートリーダ (dart leader)

先駆放電と帰還雷撃による第一雷撃が終了した時点で雲内に発生したすべての負電荷が中和されていない場合,数十msという非常に短い時間を経て,帰還雷撃が辿った放電路を再び下降していく.これをダートリーダといい,このダートリーダは先駆放電の様なステップを踏まず連続的に進んでいく(図1.d).

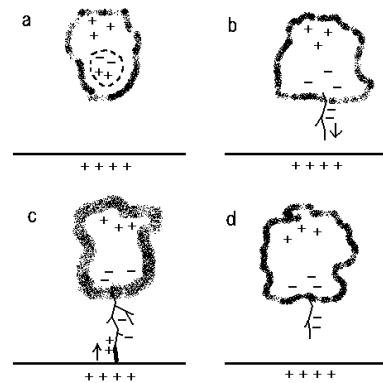


図1 落雷のメカニズム

3 関連研究

落雷シミュレーションの研究は多く行われており,文献[1]は雷の物理的特性に基づいて成長モデルを作成し,実際の落雷の動きの再現に重点を置きシミュレートを行っている.また,文献[2]は正規分布とランダム変数を利用して形状を作成し,レイトレーシングの拡張により落雷とその周りの発光を表現している.文献[3]はエクセルの表計算を数値計算として利用して結果をグラフ化し,グラフ機能を視覚表現として利用している.文献[4]では与えられた2点を基に作られた円柱をつなげることで形状を作成し,レンダリングにおいてはレイトレーシングとレイキャスティングを用いている.文献[5]は雷と雲の関係を考慮しており,与えられた雲の形状から雲内部の帯電をシミュレートし雲内部と地上部に点電荷を定義し,レンダリングを3Dテクスチャとして実現している.

4 落雷画像の生成

本研究では、落雷のメカニズム 1,2 の現象に着目し、それをできるだけ複雑さと計算の負荷をかけずに再現することを目的としている。また、それに加えて、避雷針などの誘電体が存在する場合の落雷の動的シミュレーションを考慮し、避雷針に雷が誘導される様子を再現する。

落雷の軌道には、雲から地面への落雷、雲から雲、地面から雲など様々な種類の落雷が存在するが、本研究では雲から地面への落雷を対象とする。落雷は発生から消滅までが一瞬で、人間の目には一本の稲妻にしか見えない。しかし、実際は 2 章で述べたような、絶縁破壊、帰還雷撃などをごくわずかな時間の間に繰り返しながら進行している。この様子を記録することは困難で、未だ解明されていない点が多い。そこで、本研究では落雷画像から経路の統計を取り、統計結果に基づいた落雷画像の生成を目標とした。それゆえ、本研究で提案する手法は、落雷の厳密な物理的根拠に基づいたシミュレーションを行うためのものではないが、実際の落雷画像から求めた統計データから経路を決定することで、視覚的にリアルな落雷の再現を目指している。本研究では、実際の現象や画像から見出した以下の特徴を基本条件とする。

- 上から下へ向かう
- 一直線ではなく、ある程度蛇行しながら進む
- 複数本の枝分かれが発生する可能性がある
- 放電により、幹、枝共に寿命がある

また、落雷を再現する方法として、C 言語用のグラフィックスライブラリの EGGX[8] を用いる。複数の連続した線分を描画し、それを繋いで落雷らしく表現する手法をとる。

5 統計による落雷経路の解析

本研究では、落雷の経路を表現することを目標としており、落雷経路の特徴を統計によって調べた。統計資料は基本条件の「上から下へ向かう落雷」に限定した。落雷の経路が折れ曲がるまでを 1 ステップとし、1 ステップの長さステップ間の折れ曲がりの角度の統計を取った。ここで、1 ステップの長さは、画像によってサイズがそれぞれ異なるため、各ステップの長さを全長比に直し集計した。また、折れ曲がり角の統計は、前ステップに対して左回りを正の方向とし、 10° 以上の折れ曲がりの角度の統計を取った。

統計結果より、長さは全長比 4 に集中し、折れ曲がり角度は 60° に集中して分布していることがわかる。角度の分布図はほぼ左右対称であり、左と右の折れ曲がり方向にはあまり差がないことがわかる。また、統計データから前回のステップとの折れ曲がり方向の関係については約 8 割が左から右、右から左の逆方向の折れ曲がりであった。本研究では、角度と長さの統計結果から得た落雷経路の特徴を生成式に取り入れる。落雷経路の生成手法を次章に示す。

6 生成手法

6.1 落雷経路の生成と乱数作成

落雷経路を線分の集まりとして考え、前の線分の折り曲げを繰り返し落雷経路を生成する。以下の式によって線分を決定する [3]。

6.1.1 生成式

$$x_n = x_{n-1} + L \sin \Sigma \theta$$

$$y_n = y_{n-1} - |L \cos \Sigma \theta|$$

x_{n-1}, y_{n-1} は前回の線分の終点であり、次の線分を生成するときの始点となる。また、 x_n, y_n は生成式によって得られる点で、線分の終点となる。 L, θ の値を変えることで毎回異なった線分を生成することができる。また、基本条件の「上から下へ向かう落雷」にするため y の値には計算結果の絶対値を用いて、常に下向きに動くようにする。始点と生成式によって得られた終点を結ぶ線分を基本線とし、幅を加えることで落雷を表現する。なお、本研究では物理的根拠に基づく落雷画像の生成ではないので、 L, θ には乱数を用いる。生成式で使う乱数については、第 5 章の落雷経路の統計結果を反映させた乱数を作成し、式に取り入れる。

6.1.2 統計に基づく乱数の作成

本研究では、落雷経路の決定に乱数を使う。第 5 章で調べた落雷経路の特徴の統計結果から、特徴に基づく乱数を生成し、画像生成に使用する。分布図には多少谷や山があるが、サンプル数が少ないために偏りがあったものと仮定し、この分布を滑らかな曲線に直す。角度分布は 0° を中心にほぼ左右対称に分布しているので、負の角度を正の角度に変え、絶対値のデータを使う。統計結果を図に直し、修正をして得た分布図を検討した結果、角度分布 (図 2)、長さ分布 (図 3) とともにカイ二乗分布に近似した分布が得られた。よって、本研究では角度分布乱数には自由度 13、長さ分布乱数には自由度 7 のカイ二乗分布乱数を用いる。角度分布には絶対値データを使うことにしたが、折れ曲がり方向を決定する符号については統計より約 8 割が前回と反対に折れ曲がるということがわかっている。よって、符号の決定には一様分布乱数を用いてデータに基づいた符号の決定をする。

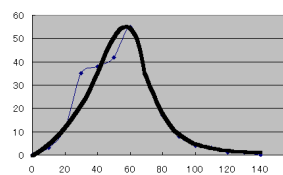


図 2 角度分布の近似

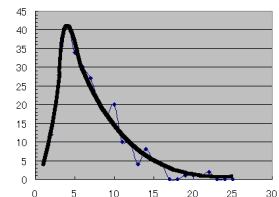


図 3 長さ分布の近似

6.2 落雷の幅の再現

落雷の幅を再現するための描画方法を手法1から3まで考えた。以下にそれぞれの結果と検証を示す。

■ 手法1 (図4)

先に述べた生成式により基本となる線分を決め、その右側に平行に a 本描く。

ある程度の落雷の幅は表現することができるが、太さが一定にならず、角度が水平に近づくにつれて細くなってしまふ。

■ 手法2 (図5)

基本となる線分から垂線を引き、三角形の相似により各ステップごとに a の値を求める。手法1と同じく、線分と平行に a 本描く。

幅はほぼ一定に描くことができるが、ステップごとの a の値が異なるため右端のラインが合っていない。

■ 手法3 (図6)

手法2で得た右端の線分を使い、その交点を求めることによってラインのずれを修正する。

手法2の問題点である右端のラインのずれが解消でき、左端と同じように右端も綺麗なラインを描くことができる。

この手法3を本研究では用いることにする。



図4 手法1による生成画像 図5 手法2による生成画像

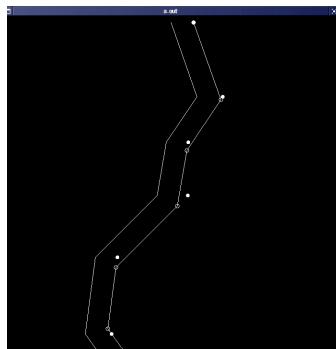


図6 手法3により得られる点

6.3 グラデーション

落雷の発光する様子をグラデーションを付けることによって再現する。その方法として、充分な幅を持った暗い色の四角形を描き、その上に少し幅が狭く、少し明るい色の四角形を重ねて描くという手順を繰り返す。

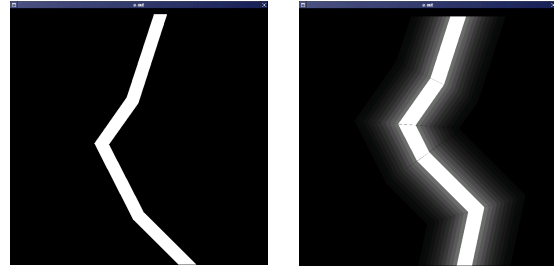


図7 グラデーションなし 図8 グラデーションあり

発光している様子のある程度再現できているが、ステップのつなぎ目にわずかな隙間ができてしまっている。これを解決するため、取得した点をいったん配列に格納し、多角形として一度に描く方法をとったところ、綺麗に表示することができた。

6.4 枝分かれの表現

落雷をより視覚的にリアルに見せるために枝分かれを表現する。実際の落雷写真から、枝分かれについて以下の二つの性質を設定した。

- 幹よりも水平に近い角度で進む
- 地面に到達せずに消滅する

枝分かれの発生位置は幹の点から一様乱数によりランダムに選ぶ。また、幹が折れ曲がった方向と逆方向にのみ発生を許可することで不自然な枝分かれの発生を防いでいる。

これらの手法により生成された落雷の全体画像を以下に示す (図9)。

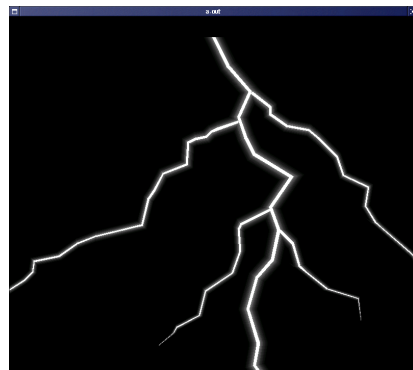


図9 全体の生成画像

6.5 落雷の軌道の変化

雷は高いところや先の尖ったものに落ちやすいという特徴がある。この特徴について、避雷針の有無による落雷の軌道の変化を以下のようにして表現した。

プログラム実行時に避雷針を配置し、避雷針 (x_u, y_u) を中心とした同心円の領域を設ける。領域は3段階までとし、コマンドラインにて領域を設定できるようにした。各領域に応じて異なる生成式を用いることで、落雷が避雷針に向かうように軌道を変化させる。各領域に用いる生成式を以下に示す。

また、よりリアルな落雷画像とするために背景画像を追加した。

■領域外の場合 領域外に落雷経路が作られている場合は、落雷は避雷針の影響を受けないものとして、生成式を参照する。

■外側の領域内の場合 この場合は、落雷は避雷針の影響をやや受けるものとする。このとき、落雷経路が避雷針の左側 $(x_n < x_u)$ であれば α を正の値に、右側 $(x_n > x_u)$ であれば α を負の値にし、ベクトルの終点を避雷針のある方向にずらす。

$$x_n = x_{n-1} + L \sin \Sigma \theta + \alpha$$

■中間の領域内の場合 軌道の変化には外側の領域内の場合と同様の方法を使う。避雷針からより大きな影響を受けるものとし、外側の場合よりも α の値を大きくする。

■内側の領域内の場合 この場合は、落雷は避雷針に落ちるものとし、ベクトルの終点を避雷針の位置にする。

$$x_n = x_u, \quad y_n = y_u$$

各領域の半径をそれぞれ 100,200,300 ドットとした例を図 10 に示す。

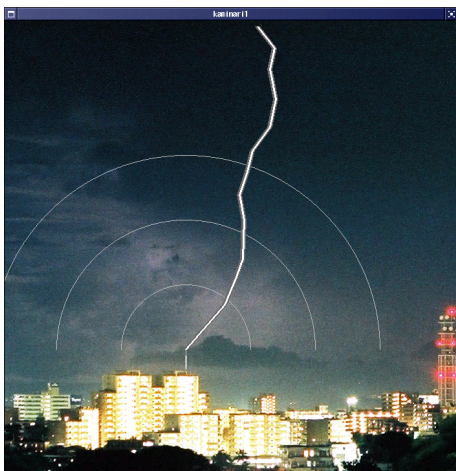


図 10 避雷針を考慮した落雷生成画像

7 おわりに

本研究では落雷の生成の様子や避雷針の有無の違いによる落雷の軌道の変化についてのシミュレーションを行うことを目的とした。手法として、まず落雷を線分の組み合わせとして考え、線分の長さや折れ曲がりによる角度を落雷画像の解析を行うことにより統計をとった。次に統計から抽出された特徴を反映した生成式を考え、その生成式を基にしてプログラムを作成した。この手法により生成された画像によりリアルさを持たせることができた。本研究では落雷経路を線分に分け、各線分の長さや角度から統計結果を作ったが、前後の線分の関係性までは十分に考慮していない。統計の要素を細かく分類することでよりリアルな落雷を作ることができると思われる。また、描画においてグラデーションをつけることによって発光している様子を再現したが、背景が黒以外の場合、落雷の周りが黒くなり不自然に見えてしまう。本研究ではグラフィックライブラリに EGGX を用いたため、発光をグラデーションで再現することを選んだが、OpenGL などの半透明処理の施せるプログラミングインターフェースを用いれば解決は可能である。また、避雷針による落雷軌道の変化については、本研究では領域を3段階に設定しているが、避雷針からの距離に反比例して軌道の変化を表現させることも可能である。上記の問題点を再度検討することが今後の課題である。

参考文献

- [1] Theodore Kim, Ming C. Lin: Physically Based Animation and Rendering of Lightning, proc. of Pacific Graphics 2004
- [2] Todd Reed and Brian Wyvill: Visual Simulation of Lightning, proc. of SIGGRAPH 1994
- [3] 白田昭司, 伊藤敏, 井上祥史: EXCEL で学ぶ理工系シミュレーション入門, CQ 出版社 (2003).
- [4] 阿部 大輔: 落雷のビジュアルシミュレーション, <http://hawk.ise.chuo-u.ac.jp/student/person/dabe/ronbun.html>
- [5] Batjargal SOSORBARAM, Tadahiro FUJIMOTO, Kazunobu MURAOKA, Norishige CHIBA: Visual Simulation of Lightning Taking into Account Cloud Growth, バーチャル・ロケーションを可能にする高度な CG 技術に関する研究開発 (2002)
- [6] 北川信一郎: 大気電気学, 東海大学出版会
- [7] 北川信一郎: 雷と雷雲の科学. 雷から身を守るには-, 森北出版
- [8] 山内千里: EGGX/ProCALL, http://phe.phyas.aichi-edu.ac.jp/cya-mauch/eggx_procall/
- [9] あおば屋 (統計参考画像元リンク集): <http://www.aobaya.jp/lightlink.html/photo/>