

# 물리적 특성을 고려한 빠른 번개 렌더링

윤정수<sup>o</sup>

윤성의<sup>\*</sup>

한국과학기술원 전산학부

[jeongsu.yun@kaist.ac.kr](mailto:jeongsu.yun@kaist.ac.kr)<sup>o</sup>

[sungeui@gmail.com](mailto:sungeui@gmail.com)<sup>\*</sup>

## Physically Inspired Fast Lightning Rendering

Jeongsu Yun<sup>o</sup>

Sung-Eui Yoon<sup>\*</sup>

School of Computing, KAIST

### 요약

본 논문에서는 번개의 물리적 특성을 고려하여 기존의 랜덤 트리 기반의 번개 경로 생성 알고리즘보다 사실적이며, 물리 기반 시뮬레이션 알고리즘보다 빠른 번개 경로 생성 알고리즘을 제시한다. 절연 파괴 현상을 물리적으로 접근한 Dielectric Breakdown Model을 사용하고, 전위장(Electric potential field)의 특징을 빠르게 근사하여 번개의 경로를 생성하는 알고리즘을 제시한다. 또한, 장애물이 있는 복잡한 장면에서 번개가 장애물을 회피하는 가이드 경로 방법을 제시한다. 마지막으로, 번개 잔가지의 두께와 밝기의 물리적 특성을 고려하여 빠르고 사실적인 번개를 렌더링한다. 본 논문에서 제시한 번개 경로는 약 1.56의 프랙탈 차원(Fractal dimension)을 가져 자연 현상의 번개와 유사한 결과를 보여주고, 기존의 물리 기반 알고리즘에 비해 월등히 빠르게 경로를 생성한다. 다만, 실시간 게임에 적용 가능한 성능을 위한 추가적인 작업이 필요하고, 향후 GPU를 활용하여 번개 경로 생성 알고리즘을 개선하면 실시간 게임에 충분히 적용할 수 있을 것이라 판단한다.

### Abstract

In this paper, we propose an algorithm for generating lightning paths, which are more realistic than those of random tree based algorithm and faster than a physically based simulation algorithm. Our approach utilizes physically based Dielectric Breakdown Method (DBM) and approximates the electric potential field dramatically to generate the lightning path. We also show a guide path method for the lightning to avoid obstacles in a complex scene. Finally, our method renders fast and realistic lightning by considering physical characteristics for the thickness and brightness of the lightning stream. Our result of the lightning path shares similarity to natural phenomenon by having about 1.56 fractal dimensions, and we can generate the lightning path faster than a previous physically based algorithm. On the other hand, our method is difficult to apply on the real-time games yet, but our approach can be improved by performing the path generation algorithm with GPU in future.

키워드: 번개, 번개 경로, 번개 렌더링

**Keywords:** lightning, lightning path, lightning rendering

## 1. 서론

컴퓨터 그래픽스 기술과 하드웨어의 발전으로 예전보다 더욱 사실적인 장면 렌더링이 가능해졌고, 이러한 기술들은 영화나 게임 산업에 많은 영향을 미치고 있다. 영화나 고화질

게임에서는 사실적인 장면 렌더링이 관객이나 게임 유저의 몰입감을 높여주기 때문에 아주 중요한 요소이다. 가상의 캐릭터, 물체, 광원 효과뿐만 아니라 눈, 비, 번개와 같은 자연 현상 또한 몰입감을 높이기 위해 중요하다. 특히 번개의 경우 영화에서 공포심을 부각시키거나 분위기를 압도하기 위한 중요한 요소이며, 게임에서는 사실적인 날씨 표현뿐만 아니라 마법과 같은 효과를 잘 표현하기 위해서도 중요한 요소이다.

그러나, 사실적인 번개의 렌더링을 위한 물리 기반의 시뮬레이션 방법은 복잡한 알고리즘과 연산으로 많은 시간을 필요로 한다. 이 방법은 영화와 같이 제작 기간은 길지만, 결과물의 질을 중시하는 산업에서는 유용할 수 있지만, 게임과 같이 실시간 유저 상호작용이 필요한 응용 프로그램에는 아직 활용하기가 힘들다.

게임에서 실시간으로 번개 효과를 표현하기 위해서는 이미 사실적으로 렌더링 된 이미지들을 반복해서 사용하거나, 조금 덜 사실적이더라도 랜덤 트리 형태의 번개 모양을 빠르게 생성하는 방법을 사용하고 있다. 이러한 방법은 비슷한 패턴을 반복해서 보여주거나, 질이 낮은 결과를 보여주기 때문에 게임을 하는 유저의 몰입감을 저해할 수 있다.

본 논문에서는 번개의 물리적 특성을 고려하여 기존의 게임에서 사용되던 랜덤 트리 기반의 번개 생성 알고리즘보다 사실적인 번개 경로 생성 알고리즘을 제시한다. 또한, 랜덤 트리 방식의 알고리즘에서 고려하지 못했던 장애물이 있는 복잡한 장면에서 장애물을 회피할 수 있는 번개 경로 생성 방법을 제시한다. 비록 실시간 게임에 활용하기에는 아직 무리가 있지만, 기존의 물리 기반 번개 경로 생성 알고리즘보다 현저히 빠른 알고리즘을 제안한다.

## 2. 선행 연구

Niemeyer<sup>[1]</sup>는 1984 년 물리 기반으로 절연 파괴(Dielectric Breakdown) 현상을 설명하는 Dielectric Breakdown Model (DBM)을 제시한다. DBM 은 장면의 공간을 정규 격자로 나눈 후 각 격자가 가지는 전위(Electric Potential)를 경계 조건(Boundary condition)을 기반으로 계산한다. 절연 파괴가 시작되는 지점의 전위는 0 의 값을 가지며, 절연 파괴가 끝나는 지점은 1 의 값을 가진다. 주어진 경계 조건을 기반으로 라플라스 방정식(Eqn. 1)을 이용하여 모든 격자의 전위를 구한다.

$$\nabla^2 \phi = 0 \text{ where } \phi: \text{전위(Electric potential)} \quad (\text{Eqn. 1})$$

그 후 절연 파괴가 진행 될 다음 위치를 결정하기 위한 후보 격자들을 선택한다. 후보 격자들은 전위가 0 인 절연 파괴가 일어난 격자들(번개가 진행된 경로)에 인접한 격자들이다. 후보 격자들에서 다음 절연 파괴가 진행되는 격자가 선택될 확률 값을 Eqn. 2 를 이용하여 계산하며, 확률 값을 기반으로 랜덤하게 다음 격자를 선택한다.  $i, j$  는 각각의 후보 격자이고,  $n$  은 모든 후보 격자의 개수이다.



Figure 1. Lightning rendering result generated by our method. It uses a uniform grid of 128 by 128 and takes 1.01 s to construct the lighting path.

새로운 절연 파괴 격자가 선택되면 해당 격자의 전위를 0 으로 설정하고, Eqn. 1 을 이용하여 절연 파괴가 끝나는 지점에 도달 할 때까지 해당 과정을 반복하여 번개의 경로를 구한다. Eqn. 2 에서  $\eta$  는 조절 변수로  $\eta$  가 0 에 가까울수록 번개의 잔가지가 많이 퍼져나가며,  $\eta$  가 2 에서 3 사이일 때 실제 번개의 모양과 비슷한 형태를 가진다.

$$p_i = \frac{(\phi_i)^\eta}{\sum_{j=1}^n (\phi_j)^\eta} \quad (\text{Eqn. 2})$$

Reed 와 Wyvil<sup>[2]</sup>는 실험적 관찰에 기반한 번개 경로 생성 알고리즘을 제시한다. 관찰에 의하면 번개의 가지들은 부모 가지와 평균 16 도의 각도를 가지며, 이런 특성을 반영하여 번개의 경로를 선형 시간복잡도로 계산한다. 번개의 경로는 빠르게 결정할 수 있지만, ray tracing 렌더링 방법을 사용하여 많은 시간을 필요로 하며, 번개의 물리학을 기반으로 번개 경로를 생성하지 않기 때문에 결과물이 실제 번개와 다른 경우를 보인다.

2001 년 Sosorbaram<sup>[3]</sup>은 전위장(Electric Potential Field)에서 랜덤 번개 모양 트리의 확장을 위해 DBM 을 사용한다. 일정 수의 음전하와 양전하를 구름과 지상에 랜덤 샘플링 한 후 라플라스 방정식 대신 전위 공식을 사용하여 전위 값을 근사

한다. 번개의 렌더링에는 3D 텍스처를 사용한 볼륨 렌더링 기법을 사용한다.

2004 년 T. Kim<sup>[4,5]</sup>은 Conjugate Gradient Method (CGM)기법을 사용하여 DBM 을 구현하며, 사실적인 번개 렌더링을 위해 Atmospheric Point Spread Function 을 이용하는 알고리즘을 제시한다. 또한, 2007 년에는 정규 격자 대신 쿼드트리와 옥트리를 이용하여 번개 경로 생성 알고리즘 속도를 개선한 알고리즘을 제시한다. 그러나 Conjugate Gradient Method 는 전위를 반복적 연산을 통해 계산하므로 많은 시간을 요구하며, 번개의 렌더링 기법도 실시간 렌더링에는 적용할 수 없을 정도로 느린 편이다.

Nvidia<sup>[6]</sup>에서는 DirectX 10 예제를 통해 geometry shader 를 이용한 빠른 번개 경로 생성 알고리즘을 제시한다. Nvidia 의 알고리즘은 GPU 를 통해 번개 경로를 매우 빠르게 생성할 수 있지만, 번개의 물리적 특성과 장면의 장애물을 전혀 고려하지 않는다.

### 3. 본론

#### 3.1 번개 경로 생성

##### 3.1.1 번개의 특성

번개의 90 퍼센트는 구름의 음전하가 지상으로 이동하는 음전하 번개(Downward negative lightning)이다. 지구 표면과 높은 대기층 사이의 강한 자기장으로 인해 미세한 전류가 흐르고 음전하 번개를 통해 지구 표면에 음전하가 공급되어 지구는 일정한 전하균형 상태를 유지한다.

번개는 소나기 구름의 형성으로 구름의 위쪽에는 양전하가 아래쪽에는 음전하가 모여 전위차가 발생하고, 전기장이 4kV/cm 이상 강해지면 전하방전이 일어난다.

번개는 몇 번의 단계를 거쳐 지상을 타격한다. 첫 번째 단계는 stepped leader이다. 음전하의 작은 덩어리로서 잔류 저항이 작은 길을 따라 구름에서 지상으로 내려오며, 그 후 몇 번 더 지상으로 내려오는 번개는 dart leader라 부른다. stepped leader는 많은 잔가지를 가지며, dart leader는 잔가지가 적고, 이전 leader를 따라 내려오며 어두운 편이다. 각각의 leader가 지상의 양전하와 만나면 순간적으로 양전하가 구름으로 이동하는 return stroke가 발생하며 강한 빛을 낸다. 번개는 stepped leader가 지상으로 내려온 후 몇 번의 return stroke와 dart leader를 반복하다 사라진다.

또한 관찰에 의해 번개의 가지들은 부모의 가지와 평균 16도 각도를 이루며, 얼음 결정, 이끼 등의 다른 자연 현상과 같이 약

1.7의 프랙탈 차원<sup>[1]</sup>을 가진다. 이런 프랙탈 특성을 가지는 패턴은 라플라시안 성장(Laplacian growth<sup>[7]</sup>) 현상으로 알려져 있다.

##### 3.1.2 번개 경로 생성 알고리즘

사실적인 번개의 렌더링은 T. Kim<sup>[4,5]</sup>의 방법이 가장 좋은 결과를 보여준다. 실제 물리학에 기반을 둔 DBM을 사용하며, 라플라스 방정식의 해를 Conjugate Gradient Method를 사용하여 근사하고 있다. 그러나 T. Kim의 알고리즘은 쿼드트리를 사용하여 성능을 개선하기는 했지만, 복잡한 계산과 너무 많은 연산으로 게임과 같은 실시간 응용 프로그램에 적용하기에는 시간이 많이 걸려 적합하지 않다. 일반적인 PC에서 128 x 128 격자 맵의 번개 경로 생성에 약 200초가 걸린다.

게임에 사용할 수 있는 빠른 방법으로 GPU를 사용하는 Nvidia의 알고리즘이 있지만, 번개 경로 생성 시 장애물의 충돌을 전혀 고려하지 않으며, 단순히 번개 모양을 랜덤 요소를 이용해 모방하는 기법이라 실제 번개와 다른 모양이 생성될 수도 있다. 이 방식은 번개가 그다지 중요하지 않은 게임이나, 하드웨어 성능이 떨어져서 적은 성능으로 빠른 결과물을 얻는 곳에서는 적합할지 모르지만, 번개가 중요한 요소가 되는 고품질 게임에는 적합하지 않다.

우리는 사실적인 번개의 렌더링을 고품질 게임에 사용할 수 있도록 DBM을 빠르게 근사하는 방법을 제안하고자 한다. 기존의 DBM을 사용한 번개 경로 생성 알고리즘은 전위장 갱신과 번개 경로 선택의 2가지 과정을 반복적으로 수행한다. 전위장 갱신 과정에서는 라플라스 방정식의 해를 근사하여, 전위장의 각 격자의 전위를 구하고, 번개 경로 선택 과정에서는 각 후보 격자의 전위 값을 기반으로 확률적 랜덤으로 다음 격자를 선택한다. 새로운 번개 경로가 선택되면 새로운 경계조건(Boundary condition)하에서 다시 전위장 갱신 과정을 반복하며, 번개가 최종 목적지에 다다를 때까지 해당 과정을 반복한다. Figure 2는 DBM 방식의 번개 경로가 선택되는 과정을 보여준다.

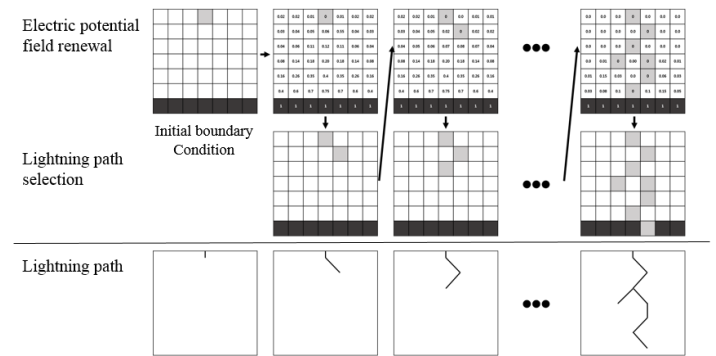


Figure 2. Lightning path generation process

T. Kim은 Conjugate Gradient Method를 이용하여 라플라스 방정식을 근사해 전위장의 값을 계산하며, 이 과정의 시간 복잡도는  $O(G^{1.5})$ <sup>[8]</sup>이고, G는 격자의 수이다. 번개 경로 선택 과정은 후보 격자에서 확률적 랜덤으로 선택하며, 이 과정의 시간 복잡도는  $O(g)$ 이고, g는 후보 격자의 수이다.

우리는 전위장 갱신을 라플라스 방정식을 사용하여 근사하지 않고, Sosorbaram<sup>[3]</sup>이 사용했던 전위 공식을 활용하여 전위장의 특징을 빠르게 근사하며, 번개 경로 선택 과정에서도 불필요한 후보 격자를 줄여 빠르게 번개 경로를 구하고자 한다. 또한 빠른 근사를 위해 번개가 도달하는 양전하 격자의 개수를 1개로 한정한다. 비행 슈팅 게임의 번개 무기는 자신에게 가장 가까이 있는 적을 목표로 날아가며, 디아블로3 마법사의 번개 마법 또한 마법이 적중할 적의 정해진 위치로 날아간다. 이처럼 대부분의 게임에서 번개가 도달하는 목적지가 상황에 따라 변할 수는 있지만 개수는 1개로 정해져 있는 경우가 많기 때문이다.

### 3.1.3 전위 값 근사

번개 현상과 같은 정전기장에서 전위(Electric potential)는 전기장 내의 단위 전하가 가지는 위치에너지이다. 전위는 Eqn. 3과 같이 전위 공식을 이용하여 구할 수 있으며, 전하의 크기에 비례하고, 전하들간의 거리에 반비례한다.  $\Phi_r$ 은 위치 r에서의 전위이고,  $\epsilon_0$ 은 진공 유전율, q는 전하량,  $r_i$ 는 위치 r에서 i번째 전하까지의 거리, n은 전기장 내 전하의 개수이다.  $\hat{r}_i$ 는 전기장 내 위치 r에서 i번째 전하와의 단위 방향 벡터로 음전하와 양전하가 만들어내는 척력과 양력을 나타낸다.

$$\Phi_r = \frac{1}{4\pi\epsilon_0} \sum_{i=1}^n \frac{q_i}{r_i} \hat{r}_i \quad (\text{Eqn. 3})$$

우리는 라플라스 방정식을 직접 풀지 않고, Eqn. 3의 전위 공식을 이용하여 격자들의 전위를 빠르게 계산하고, 전위장이 가지는 특성을 근사 하려고 한다.

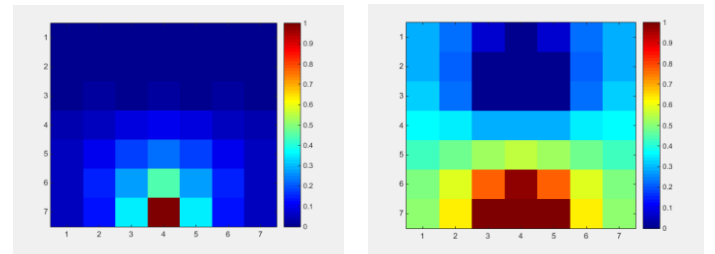
예로서 이차원 상단에 음전하와 하단에 양전하를 두는 간단한 이차원 공간을 고려해 보자. 전위장의 값들은 음전하로부터 멀수록 값이 커지며, 양전하에 가까울수록 값이 커지는 특성을 가지고 있다. 격자의 경계 외부는 번개가 지나갈 수 없는 곳으로 0의 전위 값을 가지는 음전하로 가정한다. 또한 번개의 시작 지점에서 도착 지점으로 향하는 방향으로 전위 값이 약 2배씩 증가하는 특성을 보인다. 특정 환경에서 Conjugate Gradient Method를 이용하여 계산한 전위장의 값을 Table 1에 나타내었다.

Figure 3은 Conjugate Gradient Method와 전위 공식(Eqn. 3)을 이용하여 계산한 전위장을 열전도 그래프로 보여준다. 전위 공식을 이용한 결과가 CGM 결과보다 예러 값이 더 크지만, 음전하

주변에서는 작은 값을 가지고, 양전하 방향으로 갈수록 큰 값을 가지는 전위장 특성을 만족하고 있다. 다만, 격자 외곽의 경계에 갈수록 값이 작아져야 하는 특성은 예러 값으로 인해 만족하지 못하고 있다. 따라서, 우리는 Eqn. 3의 결과 값을 바로 사용하지 않고, 격자 경계 음전하, 번개 경로의 음전하, 번개 도착 지점의 양전하에 대해 각각 분리하여 계산 한 후 전위장의 특성을 만족하도록 전위 값을 근사 한다. 분리된 각각은 모두 전위장의 특성을 만족하며, 고정된 위치의 음전하에 대한 전위 값은 미리 계산하여 번개 경로 생성 시간을 단축할 수 있다.

Table 1. Electric potential field solved by CGM, where 1.0 in the dark bottom cell is the destination cell and cells with 0.0 indicate a lighting path.

0.002	0.003	0.001	0.0	0.001	0.003	0.002
0.006	0.008	0.002	0.0	0.002	0.008	0.006
0.015	0.021	0.0	0.027	0.0	0.021	0.015
0.032	0.062	0.086	0.106	0.086	0.062	0.032
0.051	0.109	0.176	0.226	0.176	0.109	0.051
0.062	0.148	0.281	0.447	0.281	0.148	0.062
0.05	0.138	0.355	1.0	0.355	0.138	0.05



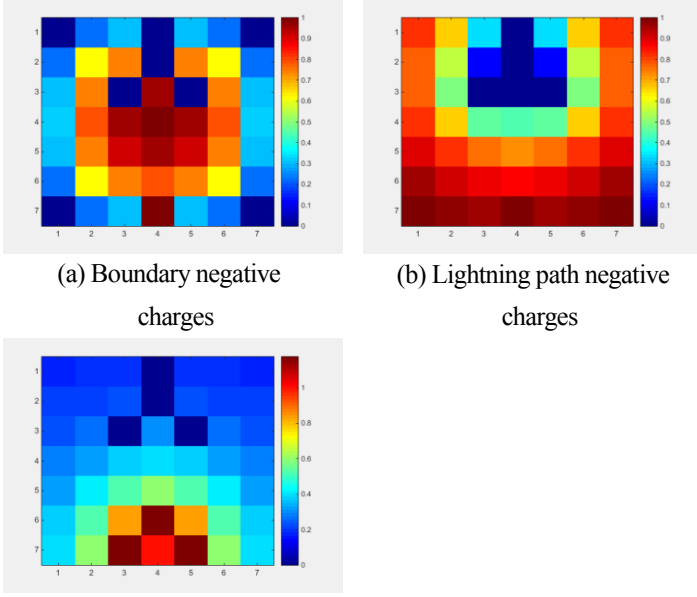
(a) Solved by CGM (b) Solved by Eqn. 3

Figure 3. Heat map visualization of electric potential field

Figure 4는 격자 경계 음전하, 번개 경로의 음전하, 번개도착 지점의 양전하에 대해 계산한 전위장 열전도 그래프이다. Figure 4의 (a)는 격자 경계의 음전하에 대한 전위장으로 음전하에서 멀어질수록 값이 커지는 특성을 만족하여 경계에서 안쪽으로 갈수록 큰 값을 가진다. (b)와 (c)는 번개 경로의 음전하와 번개의 도착 지점인 양전하에 대한 전위장으로 음전하에서 멀어질수록 값이 커지고, 양전하에 가까울수록 값이 커지는 전위장의 특성을 만족한다.

Conjugate Gradient Method는 전역 함수(global function)이므로 모든 격자의 값을 계산하고 반복적인 계산을 통해 예러 값을 줄여나가는 방식이지만, 우리는 Eqn. 3을 사용하여 다음 번개 경로가 될 후보 격자의 전위만 계산할 수 있으므로 CGM방식에 비해 많은 시간을 단축할 수 있다.





(a) Boundary negative charges  
(b) Lightning path negative charges  
(c) Positive charge  
Figure 4. Heat map visualization for each part of electric charges. The sum of these terms equals to (b) of Fig. 3

또한 격자 경계의 음전하에 대한 전위 값들은 고정되어 있는 값이므로 미리 계산을 해두고 사용할 수 있다. 게임에서 성능 향상을 위해 미리 계산할 수 있는 부분은 미리 계산하여 리소스 파일로 저장한 후 실제 렌더링 시에 파일에서 읽어 들여 사용하는 기법을 자주 사용한다. 우리는 Eqn. 3을 이용하여 격자 경계의 음전하에 대한 값을 미리 계산한 후 정규화하여 경계 맵(Boundary map)으로 저장하여 사용한다. Eqn. 4는 격자 경계의 음전하에 대한 값을 나타내며, 전위장의 값은 B에 비례한다. 우리는 다음에 설명할 번개 경로의 음전하와 번개 도착 지점의 양전하에 대한 값과 통일성을 유지하기 위해 전위장이 B의 값에 반비례하도록 Eqn. 4로 계산한 값을 역으로 정규화한 값을 저장한다.

$$B = \text{Normalize} \left( \frac{1}{4\pi\epsilon_0} \sum_{i=1}^n \frac{q_i}{r_i} \right) \quad (\text{Eqn. 4})$$

다음으로 번개의 도착 지점인 양전하에 대한 전위 계산은 번개의 도착 지점인 양전하가 1개뿐이므로 각 후보 격자와의 거리를 매번 계산한다. 번개의 도착 지점인 양전하에 대한 전위 값은 Eqn. 5를 사용하여 근사한다. r은 후보 격자의 위치이고, p는 양전하 격자의 위치이다.

$$P = \text{Distance}(r, p) \quad (\text{Eqn. 5})$$

마지막으로 번개 경로의 음전하는 새로운 번개 경로가 선택되는 과정을 통해 개수가 동적으로 늘어나므로 미리 값을 구할 수 없고, 계산 시간도 늘어난다. 번개의 경로가 늘어날수록 후보

격자의 수도 늘어나므로 우리는 이 부분의 연산을 더 줄이도록 한다. 격자의 전위는 음전하에서 멀리 있는 경우에는 영향을 덜 받고, 가까이 있는 경우에는 영향을 많이 받기 때문에 우리는 멀리 있는 번개 경로 음전하의 영향은 버리고, 후보 격자와 인접해 있는 번개 경로 음전하만 고려하여 값을 빠르게 근사한다.

후보 격자에 인접한 번개 경로 음전하와의 거리는 1 혹은  $\sqrt{2}$  이므로 단순히 인접한 번개 경로 음전하의 개수로 근사하여 연산도 줄이며, 구현도 간단하게 만든다. 2D 격자에서 최대 8개의 인접한 격자를 가지므로 [1, 8] 값이 된다. Eqn. 6은 번개 경로 음전하에 대한 전위를 근사하며, 번개 경로 음전하에 인접할수록 큰 값을 가지므로, 전위장의 값에 반비례한 값을 가진다.

$$N = \text{Number of adjacent negative charges} \quad (\text{Eqn. 6})$$

분리하여 근사한 B, P, N 값을 이용하여 각 격자의 전위 값을 근사한다. B, P, N 값 모두 전위장의 전위 값에 반비례하는 값을 가진다. 그리고 인접한 격자들 간의 전위 차이를 크게 만들기 위해 Eqn. 7과 같이 값을 곱하여 전위를 근사한다. 인접한 격자들의 전위 차이가 크지 않으면, 비슷한 확률 값을 가지므로 잔가지가 번개의 방향과 무관하게 전체적으로 퍼져나가게 된다.

$$P_i = \frac{1}{B \times N \times P} \quad (\text{Eqn. 7})$$

Eqn. 7을 이용하여 전위를 계산하면 마지막으로 선택된 번개 경로 격자 주변의 전위는 변하지만, 그 외의 후보 격자들의 전위는 변하지 않는다. 실제로 전기장의 전위 값은 번개가 진행될수록 늘어나는 음전하의 영향으로 값이 작아지며, 번개의 시작 지점에서 번개가 도착하는 양전하의 방향으로 약 2배씩 값이 커지는 특성이 있다. 이런 특성을 근사하기 위해 Eqn. 8과 같이  $2^G$ 를 곱하여 값이 약 2배씩 커지도록 만든다. G는 후보 격자가 번개의 시작 위치에서 도착 위치로 향하는 벡터를 격자 크기로 나눈 단계의 어느 단계에 속해 있는지를 나타내며, 후보 격자의 위치를 번개의 시작 위치에서 도착 위치로 향하는 벡터에 투영시켜 계산한다.

$$P_i = \frac{1}{B \times N \times P} \cdot 2^G \quad (\text{Eqn. 8})$$

마지막으로 격자의 크기가 작은 경우에는 직접 지수 연산을 수행할 수 있지만, 격자의 크기가 커지는 경우 지수 연산 결과가 변수에 저장할 수 없을 정도로 커지기 때문에 우리는 로그 연산을 사용해 Eqn. 8을 구현한다. 번개 경로 선택 과정에서는  $P_i$ 를 확률 값으로 적용하여 랜덤으로 다음 번개 경로를 선택한다.

Table 2는 Eqn. 8를 통해 근사한 후보 격자의 전위 값을 보여준다. 각 전위 값들은 번개의 시작 지점에서 번개의 도착 지점으로 갈수록 증가하며, 격자 경계와 번개 경로 격자에 가까이 있을수록 작은 값을 가지는 전위장의 특성을 만족하며, 후보 격자의 전위만 계산하므로 Conjugate Gradient Method보다 빠르다.

Table 2. Electric potential field solved by our method

		0.004	0.0	0.004		
	0.017	0.015	0.0	0.015	0.017	
	0.067	0.0	0.074	0.0	0.067	
	0.224	0.584	0.500	0.584	0.224	
			1.0			

번개가 이미 지나간 주변의 후보 격자들은 아주 작은 전위를 가지기 때문에 다음 번개 경로로 선택될 확률이 낮아지며, 번개 경로가 늘어나면 음전하의 수가 늘어나서 더 낮은 확률을 가지게 된다. 이런 불필요한 후보 격자의 수를 줄이면 성능을 더욱 향상시킬 수 있으므로, 우리는 파티클 시스템(Particle system)과 같이 후보 격자의 생명 시간을 추가한다. 후보 격자들이 사용자가 정한 생명 시간을 넘어서면, 더 이상 후보 격자 리스트에 추가하지 않도록 하여 후보 격자의 수가 급격하게 증가되는 것을 막도록 한다. 생명 시간 조절 값이 너무 작으면 번개의 경로가 직선에 가깝게 잔가지의 길이가 작아지며, 너무 많으면 후보 격자의 수를 줄이는 효과가 없다. 우리는 이후의 실험과 결과에 생명 시간 조절 값으로 4를 사용하였다.

### 3.1.3 번개 경로 생성 결과

본 논문에서는 전위 값 근사 알고리즘을 이용한 번개 경로 생성과 물리 기반의 선행 연구인 T. Kim의 번개 경로 생성 알고리즘을 구현하였다. Figure 5는 2.6 GHz Intel Core i7 CPU를 사용하는 PC에서 본 논문과 T. Kim의 결과를 비교한 것이다. 그래프에서 볼 수 있듯이 우리가 제시한 번개 경로 생성 알고리즘이 64 x 64 격자에서는 약 50배, 128 x 128 격자에서는 약 200배 빠른 결과를 보인다. 128 x 128 격자 크기에서 약 1초의 시간은 실시간 게임에 바로 적용하기는 어렵지만, 기존의 물리 기반의 번개 경로 생성 알고리즘보다는 현저히 빠르다.

Figure 6은 본 논문의 번개 경로와 T. Kim의 번개 경로 결과를 비교한 것이다. 본 논문의 결과에서 잔가지의 길이가 조금 짧은 편이지만, 실제 번개의 모습과 유사한 결과를 보여주고 있다. 눈으로는 두 결과의 차이 비교하기 힘들기 때문에 우리는 두 결과

의 프랙탈 차원(Fractal dimension)을 값을 비교하였다. 두 결과 모두 약 1.56의 프랙탈 차원을 가지며 자연 현상의 프랙탈 차원인 1.7에 근접하여 실제 번개의 모습과 유사하다는 것을 보여준다.

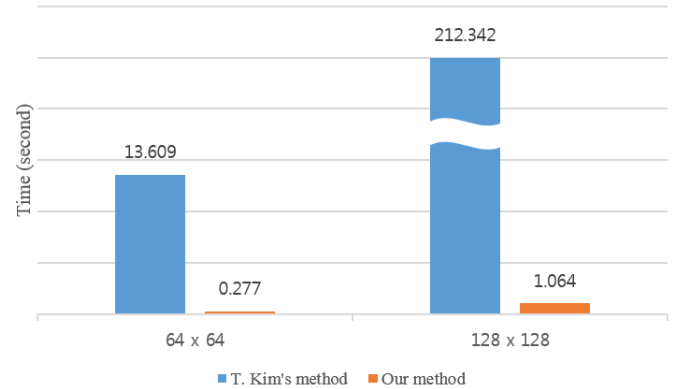
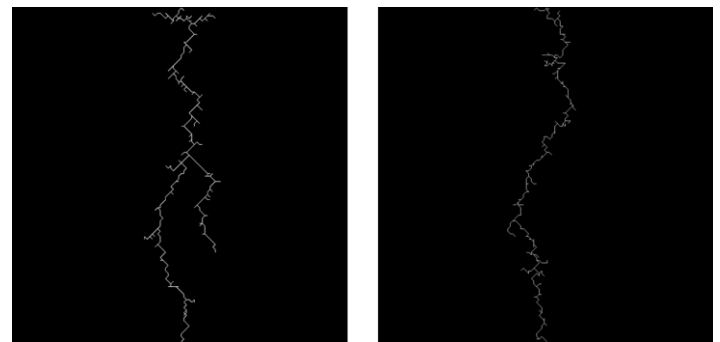


Figure 5. Comparison of lightning path generation time



(a) T. Kim's method

(b) Our method

Figure 6. Comparison of lightning path

## 3.2 장애물을 고려한 번개 경로 생성

### 3.2.1 장애물 회피를 위한 번개 경로 생성

본 논문의 전위 값 근사 알고리즘으로 번개가 피해야 할 많은 물체들을 가진 복잡한 장면에서 적용하면 로컬 미니마(Local Minima) 문제가 발생한다. 이는 우리가 전위 값을 근사하는 과정에서 번개의 도착 지점을 향하도록 값을 유도했기 때문이다. 도착 지점 방향을 물체가 가로 막고 있다면, 포텐셜 필드(Potential field<sup>9)</sup>)의 경로 탐색 기법이 가지는 로컬 미니마 문제에 빠져 제대로 된 경로를 찾지 못하고 모든 경로를 탐색하려고 한다.

이 문제를 해결하기 위해 우선 번개가 피해야 할 장면의 오브젝트들 또한 음전하로 가정한다. 장면의 고정된 음전하로 인한 전위 값은 격자 경계 음전하의 경우와 마찬가지로 미리 계산을 할 수 있다. Figure 7은 장애물이 포함된 장면에서 격자 경계 외부와 장애물의 음전하에 대해 계산한 전위 값을 열전도 그래프로 보여준다. (a)와 (b)에서 전위 값들은 격자 경계에서 멀어질수록,

오브젝트에서 멀어질수록 큰 값을 가지는 전위장 특성을 만족한다. (c)는 격자 경계와 오브젝트를 동시에 고려한 열전도 그래프이며, 전위는 이 값에 반비례하므로 우리는 (d)와 같이 반대 값을 가지는 경계 맵(boundary map)을 사용한다.

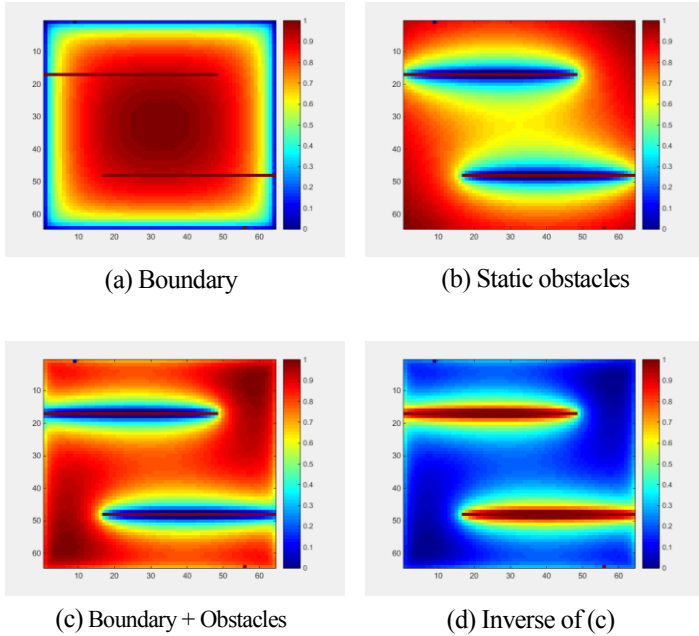


Figure 7. Heat-map visualization for boundary and static obstacle negative electric charges

로컬 미니마 문제를 해결하기 위해 우리는 경로 탐색 알고리즘인 A\* 알고리즘을 적용하여, 번개 경로의 가이드 경로를 적용한다. 대부분의 게임에는 실시간 경로 탐색 알고리즘이 대부분 구현되어 있고, 이를 쉽게 활용할 수 있다. 우리는 경로 탐색 결과를 활용하여, 경로의 방향이 바뀌는 정보만을 활용하여 전체 번개 경로를 여러 개의 서브 번개 경로의 연결로 구현한다. Figure 8은 전체 번개의 경로의 시작과 끝을 여러 개의 서브 번개 경로의 시작과 끝의 연결로 표현하는 것을 보여준다.

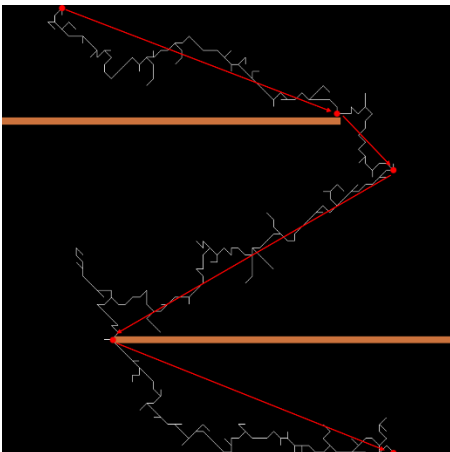


Figure 8. Guide path for sub-lightning path

서브 번개 경로의 도착 지점은 다음 서브 경로의 시작 지점이 되지만, 가이드 경로의 도착 지점을 반드시 지난다면, 다양한 번개 경로를 만들 수 없다. 그래서 서브 번개 경로의 도착 지점에 인접하거나, 다음 서브 번개의 가이드 경로가 마지막에 추가된 번개 경로 격자에서 물체에 가려져 있지 않다면 다음 서브 번개의 가이드 경로를 사용하도록 하여 정형화된 번개 경로를 생성하지 않도록 한다

### 3.2.2 장애물을 고려한 번개 경로 결과

Figure 9는 장애물이 있는 장면에서 본 논문과 T. Kim의 번개 경로 생성 시간을 보여준다. 장애물이 없는 알고리즘에 비해 가이드 경로 생성과 가시화 검사가 추가 되었지만, T. Kim의 결과에 비해 월등히 빠른 속도를 보여준다.

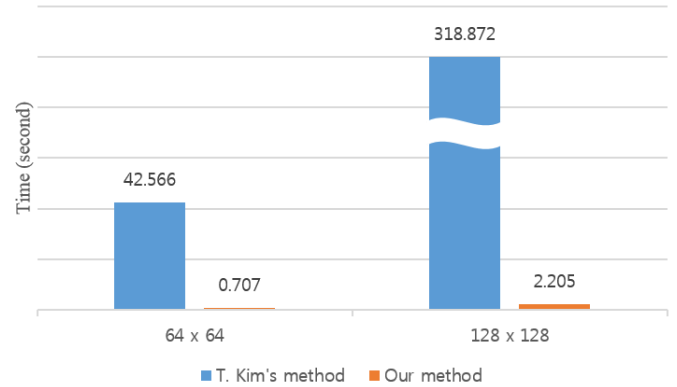
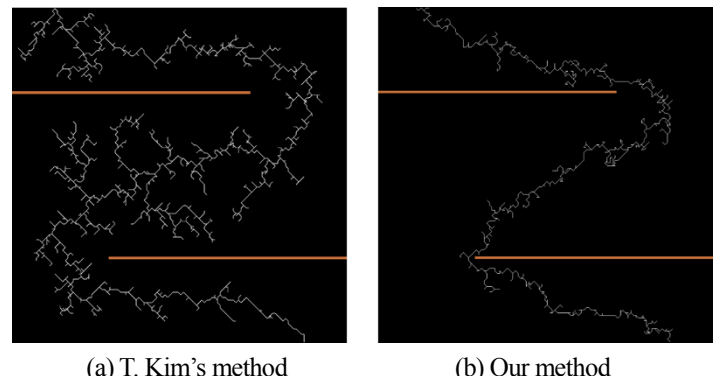


Figure 9. Comparison of lightning path generation time in the complex scene

Figure 10은 128 x 128 격자의 장애물이 있는 맵에서 T. Kim과 우리가 제안한 알고리즘으로 생성된 번개 경로 결과를 보여준다. 우리가 제안한 알고리즘은 가이드 경로를 따라가려고 하기 때문에 T. Kim의 결과에 비해 여러 방향으로 뻗어나가는 잔가지가 적게 보이지만 로컬 미니마 문제를 해결하면서 적절한 번개 경로를 보여준다.



(a) T. Kim's method (b) Our method  
Figure 10. Comparison of lightning path in the complex scene

### 3.3 번개의 렌더링

#### 3.3.1 렌더링 알고리즘

번개의 렌더링은 T. Kim과 Reed의 선행 연구에서 제시한 방법을 참고하여 번개의 밝기와 번개 가지 주변 빛의 산란을 구현한다.

T. Kim은 번개 경로를 메인 채널(Main channel)과 주변 채널(Secundary channel)로 구분한다. 메인 채널은 번개의 시작점과 도착점을 연결하는 경로이고, 주변 채널은 메인 채널에서 뻗어 나온 잔가지이다. T. Kim은 메인 채널은 고정된 밝기를 사용하고, 주변 채널은 메인 채널에서 멀어질수록 밝기가 줄어들도록 하였다. 하지만, 번개 주변 대기의 빛의 산란을 표현하기 위해 APSF기법을 사용하여, 실시간 렌더링은 불가능하다.

Reed는 메인 채널의 두께는 주변 채널의 2배라는 관찰에 의한 특성을 사용한다. 번개의 렌더링에는 가우시안 블러를 사용하지만, ray tracing 렌더링 기법을 사용하여 T. Kim의 방법과 마찬가지로 실시간 렌더링은 불가능하다.

우리는 T. Kim과 Reed가 제시한 메인 채널과 주변 채널의 두께와 밝기의 물리적 특성은 그대로 사용하고, 번개의 렌더링은 웨이더를 이용하여 빠른 2단계 가우시안 블러(Two-Pass Gaussian Blur) 필터링 기법을 적용한다.

렌더링은 OpenGL을 이용한 Deferred 렌더링 방식을 사용한다. 한 개의 프레임 버퍼 텍스처에는 장면을 렌더링하고, 또 다른 프레임 버퍼 텍스처에는 가우시안 블러 필터링을 적용할 번개 경로를 렌더링 한다. 이때 번개 메인 채널과 주변 채널의 밝기와 두께를 고려하여 렌더링 한다. 번개 경로의 프레임 버퍼 텍스처에 웨이더를 이용하여 가로 방향으로 가우시안 블러를 적용하고, 그 결과를 다시 세로 방향으로 가우시안 블러를 적용하는 2단계 가우시안 블러 필터링을 기법을 적용하여 최종 번개의 텍스처를 만들어 낸다. 마지막으로 장면 텍스처와 최종 번개 텍스처를 혼합하여 화면에 렌더링 한다. Figure 11은 본 논문의 렌더링 과정을 보여준다.

#### 3.3.2 지터링

번개 렌더링 결과에서 눈에 띄는 결함 중 하나는 격자 맵의 해상도가 낮아서 결과물이 사각형 형태를 보여주는 것이다. 번개 경로의 라인이 격자의 중심 좌표들끼리 연결되어 있기 때문에 낮은 해상도의 격자에서는 이런 문제가 발생한다. 이 문제를 해결하는 가장 좋은 방법은 격자의 해상도를 화면 해상도와 같도록 맞추는 것이지만, 격자의 해상도가 높아지면 번개 경로 생성 시간도 증가하기 때문에 실시간 게임에서는

사용할 수 없다. 우리는 T. Kim이 결과에 적용한 것과 동일하게 지터링(jittering) 샘플링 기법을 이용하여, 번개의 라인이 격자 내부의 랜덤한 위치와 연결되도록 하여 더욱 사실적인 번개를 렌더링 할 수 있도록 한다.

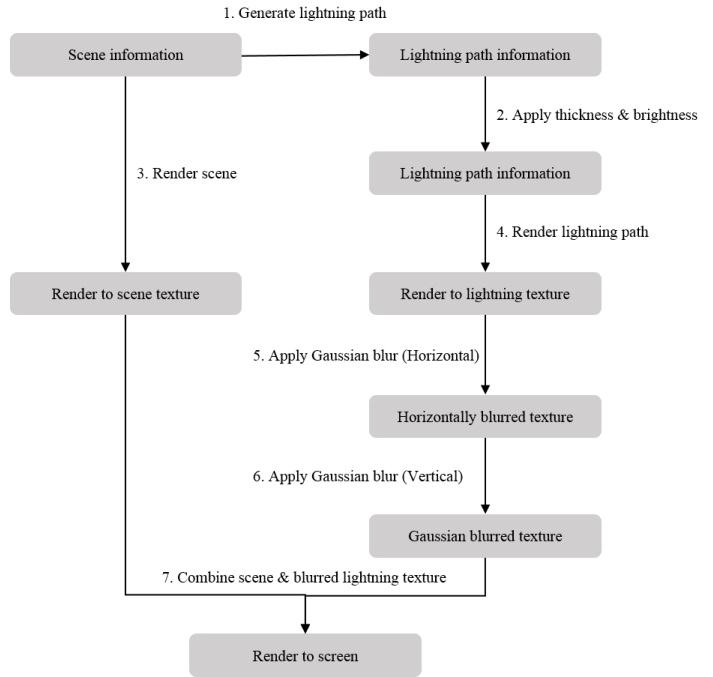


Figure 11. Lightning rendering process

#### 3.3.3 결과

Figure 1은 서울의 야경 장면에 우리가 제안한 번개가 추가된 렌더링 결과를 보여준다. 번개 경로 생성은 2.6 GHz Intel iCore 7 PC에서 128x 128 격자를 사용하여, 1.01초가 걸렸다.

### 4. 결론

본 논문은 기존의 물리 기반 번개 경로 알고리즘의 속도를 개선하여 향후 게임과 같은 실시간 응용 프로그램에서 사실적인 번개의 렌더링을 하는 것이 목표이다. 물리 기반의 번개 경로 알고리즘은 복잡한 연산 때문에 많은 시간을 필요로 하여 게임과 같은 실시간 응용 프로그램에는 활용하기 힘들며, 기존의 게임들은 빠른 번개 경로 생성을 위해 사실적이지 못한 랜덤 트리 기반의 알고리즘을 사용하였다.

사실적인 번개 경로 생성을 위해 절연 파괴 현상을 물리적으로 접근한 DBM을 사용하고, 빠른 전위 값 근사를 위해 라플라스 방정식 대신 전위 공식을 이용하여 전위장의 특성을 빠르게 근



사하였다. 번개의 렌더링은 T. Kim과 Reed가 제시한 번개의 메인 채널과 주변 채널의 특징을 이용하였으며, GPU를 활용하여 빠른 2단계 가우시안 블러 필터링 기법을 적용하였다.

T. Kim의 물리 기반 결과에 비해 잔가지의 길이가 짧은 편이지만, 약 1.56의 프랙탈 차원 (Fractal dimension) 값을 가져 자연 현상의 번개와 유사한 결과를 보이고, 기존의 물리 기반 시뮬레이션 알고리즘에 비해 월등히 빠른 결과를 보여준다. 다만, 아직 1초 이상의 시간이 걸려 당장 실시간 게임에 적용하기에는 무리가 있기 때문에 번개 경로 생성 알고리즘을 GPU를 활용하여 개선할 예정이며, 더욱 사실적인 번개의 잔가지 표현을 위해 CGM보다 에러 값은 크지만 빠르게 값을 근사할 수 있는 Solver를 적용해 볼 예정이다.

## 감사의 글

OpenGL 구현에 도움을 준 손명배, 최병윤, Pio Claudio 님과 리뷰를 도와주신 윤웅직, 권용선님에게 감사 드립니다. 본 연구는 미래창조과학부 및 정보통신기술연구진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음 [R0126-16-1108].

## References

- [1] L. Niemeyer, L. Pietronero, and H. J. Wiesmann, "Fractal dimension of dielectric breakdown", *Physical Review Letters*, vol. 52, pp. 1033–1036, 1984.
- [2] T. Reed and B. Wyvill, "Visual simulation of lightning", *Proc. Of the 21st annual conference on Computer graphics and interactive techniques*, pp. 359-364, 1994.
- [3] B. Sosorbaram, T. Fujimoto, K. Muraoka, and N. Chiba, "Visual simulation of lightning taking into account cloud growth", *Proc. Computer Graphics International 2001*, pp. 89-95. 2001.
- [4] T. Kim and M. Lin, "Physically Based Modeling and Rendering of Lightning", *Proc. Of pacific Graphics 2004*, pp. 267-275, 2004.
- [5] T. Kim and M. Lin, "Fast Animation of Lightning Using an Adaptive Mesh", *Visualization and Computer Graphics*, vol. 13, pp. 390-402, 2007.
- [6] Nvidia Corporation, "Lightning, NVIDIA Direct3D SDK 10 Code Samples", 2007.  
(See also [http://developer.download.nvidia.com/SDK/10/direct3d/Source/Lightning/doc/lightning\\_doc.pdf](http://developer.download.nvidia.com/SDK/10/direct3d/Source/Lightning/doc/lightning_doc.pdf))
- [7] M. Hastings and L. Levitov, "Laplacian growth as one-dimensional turbulence", *Physica D*, vol. 116, pp. 244–252, 1998.
- [8] J. Shewchuk, "An introduction to the conjugate gradient method

without the agonizing pain", Technical Report CMUCS-TR-94-125, Carnegie Mellon University, 1994.

(See also <http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.ps>)

- [9] Oussama Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", *The International Journal of Robotics Research*, vol. 5, pp. 90-98, 1986.