



불의 애니메이션을 위한 복합적 기법

민경하
상명대학교 디지털미디어 학부
minkh@smu.ac.kr

A Hybrid Technique for Fire Animation

Kyungha Min
Division of Digital Media, Sangmyung Univ.

요 약

본 논문에서는 불을 애니메이션하고 가시화하는 새로운 방법을 제시한다. 불을 애니메이션하기 위해 해결해야 하는 문제들 중에서 가장 어려운 문제는 불이 빛과 열을 주위로 방사하는 과정을 시뮬레이션하는 것이다. 이 방법을 해결하기 위해서 연료가 빛과 열을 발산하고 연소물이 되는 연소 과정을 유체 방정식이 구현되는 복셀화된 공간에서 시뮬레이션하는 복합적인 방법을 제안한다. 불의 움직임은 다른 유체의 경우와 같이 유체 방정식을 이용해서 시뮬레이션하며, 시뮬레이션의 각 단계마다 연소 과정에서 발생한 열이 유체의 온도를 높이고, 이렇게 높아진 온도가 불의 복잡한 움직임을 더욱 가중시키는 과정을 반복시키면서 불에 대한 애니메이션이 처리된다. 또한, 불과 불이 주변을 밝히는 효과를 처리하기 위해서 널리 알려진 포톤 매핑 알고리즘의 효율성을 개선한 새로운 가시화 알고리즘을 구현한다.

ABSTRACT

In this paper, we present a new fire animation and visualization scheme. The most difficult problem in creating fire animation is how to simulate the mechanism of emitting lighting and heat of fire. We attack the difficulty by presenting a hybrid scheme that combines the simulation scheme and the combustion process in voxelized space where the numerical solution of the classical fluid equations is implemented. Therefore, the combustion process is simulated at each voxel and the amount of heat generated at the voxel is estimated. The generated heat will increase the temperature at the voxel, where results in the increase of turbulent motion of fire. We also propose a visualization scheme that modifies the photon mapping algorithm in order to render fire and various lighting effects of fire to the environments.

Keyword : hybrid algorithm, fluid simulation, fire, natural phenomena, photon mapping

1. 서론

애니메이션, 영화, 게임 등 최근 각광받고 있는 다양한 디지털 엔터테인먼트용 콘텐츠들을 제작하기 위해서 필요한 기술들 중에서 사실적인 영상을 생성하는 기술은 매우 중요한 기술이다. 이러한 사실적인 영상들 중에서 물, 불, 연기 등과 같은 자연 현상에 대한 영상은 가장 생성하기 어려운 것으로 인식되고 있다. 이러한 자연 현상에 대한 사실적인 영상을 얻기 위해서는 매우 복잡한 수학적 모델에 대한 수치적 해법을 구해야 함과 동시에 정교한 가시화 알고리즘을 적용해야 한다는 어려운 점이 존재한다. 특히, 불에 대한 애니메이션은 여러 자연 현상들 중에서 특히 어려운 대상으로 인식되는데, 그 이유는 불은 물리적이면서 화학적인 두 가지 성질을 동시에 갖는 복합적인 모델에 의해서 설명될 수 있기 때문이다. 불의 애니메이션이 다른 유체의 애니메이션에 비해서 어려운 이유는 불은 유체 역학의 지배를 받는 유체인 동시에 화학적 반응인 연소 반응에 의해서 그 움직임이 지배된다는 점이다. 화학적으로 연소는 특정한 온도 아래에서 연료와 산소가 결합하여 연소물이 되면서 빛과 열을 발산하는 화학적인 반응으로 정의된다. 본 연구에서는 사실적이면서 효율적인 불의 애니메이션을 위해서 연소 과정의 화학적인 성질을 모델링하여 시뮬레이션하는 기법을 개발하여 이를 기존의 유체 방정식에 근거한 유체 애니메이션 기법과 결합시킴으로써 새로운 불의 애니메이션 방법론을 제시한다.

본 논문에서 제안하는 연구 방법의 가장 중요한 특징은 복셀화된 공간에서 유체 방정식의 수치적 해법을 구하는 방법론과 연료의 연소를 시뮬레이션하는 방법론이 서로 협력하는 형태로 구현되는 복합적인 방법론 (hybrid method)을 제안한다는 점이다. 이를 위해서 본 연구에서는 기존의 유체 방정식에서 이용하는 유체가 연료에서 연소물로 변환되면서 열을 발산하는 방법론을 제시한다. 즉, 연소의 초기에는 모든 유체가 연료이지만, 연소가 진행됨에 따라 연료의 비율이 감소하고 연소물의 비율이 증가하게 되어서 결과적으로는 모든 유체가 연소물이 되는 과정을 겪게 된다. 유체 방정식이 시뮬레이션되는 각 복셀에서는 연소 과정이 시뮬레이션되어서 열을 발생시키고, 발생된 열은 복셀의 온도를 올리는 과정을 통해서 유체 방정식과 서로 협동적으로 불의 움직임에 대한 애니메이션을 수행하게 된다. 이러

한 시도가 Feldman 등이 수행한 연구와 구별되는 점은 입자 시스템을 도입하지 않고, 기존의 유체 방정식을 이용한 시뮬레이션 모델만으로 연소를 시뮬레이션하는 기법을 개발하였다는 점이다. 이를 통해서 더 효율적으로 불에 대한 애니메이션을 처리할 수 있다는 장점을 얻게 되었다.

유체 애니메이션의 또 다른 어려운 점은 불과 불로부터 발산된 빛이 주변 환경을 밝히는 효과를 가시화하는 점이다. 여러 연구자들이 후측 복사에 근거하여 불의 온도로부터 불꽃의 색을 추정하는 연구를 수행해왔다 [15,11]. 본 연구에서는 Nguyen 등 [10,15]이 이용했던 포톤 매핑에 근거한 불꽃의 가시화 방법을 효율적으로 구현하는 방법을 제안한다. 기존의 포톤 매핑 방법에서 포톤을 효율적으로 관리하기 위해서 사용하던 k-d 트리 구조를 복셀로 바꾸고, 포톤을 샘플링하는 방법의 시간 복잡도를 $O(n^3)$ 에서 $O(n^2)$ 으로 향상시킴으로써 본 논문에서는 기존의 가시화 알고리즘보다 훨씬 빠른 속도의 가시화 알고리즘을 구현하였다.

본 논문은 다음과 같이 구성된다. 먼저, 2장에서는 불의 애니메이션에 대한 기존 연구를 소개하고, 3장에서는 불의 움직임을 처리하는 유체 모델을 설명하며, 4장에서는 불의 연소 과정을 시뮬레이션하는 연소 모델을 제시한다. 그리고 5장에서는 불을 가시화하는 방법을 소개하며, 6장에서는 구현 결과를, 그리고 7장에서는 결론 및 향후 연구 방향을 서술한다.

2. 기존 연구

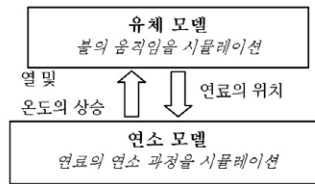
기존의 불의 애니메이션에 관한 연구는 크게 절차적 모델을 이용한 연구와 유체 역학에 근거한 연구로 구분할 수 있다. 절차적 모델을 이용한 불의 애니메이션 기법 중에서 첫 번째 연구는 Inakage 가 제안한 텍스처를 불꽃 형태의 음함수형 곡면에 적용하여 볼륨 렌더링을 수행하는 단순한 계층적 불꽃 모델이다 [9]. Chiba 등은 공간을 작은 셀로 나누고 각각의 셀들이 연소할 연료에 대한 정보를 가지고 있는 2차원 불꽃 전파 모델을 제시하였다 [4]. 이 연구에서 각 셀은 자신의 온도가 점화 온도보다 높은 상태에서 열과 빛을 발산하게 된다. Perry와 Picard는 다각형 메쉬 위에서 불꽃이 전파되는 과정을 시뮬레이션하는 방법을 연구하였다 [16]. 이 방법에서 불꽃의 전단면은 입자 시스템으로 표현되

며, 불꽃이 확장되는 과정은 새로운 입자들이 추가되는 것으로 처리된다. Bukowski와 Sequin은 건물의 화재 안전 시스템을 위한 설계 환경에 기반한 단순하고 대화식의 불의 시물레이션 기법을 제안하였다 [3]. Beaudoin 등은 불꽃을 음함수형 곡면으로 표현하고 시간에 근거한 벡터 장으로 불꽃을 조절하는 방법을 발표하였다 [1]. 이 방법에서는 삼각형 메쉬 위에서 불꽃이 전파되는 과정을 시물레이션하였으며, 불륨 렌더링 기법을 이용해서 불을 렌더링하였다. Lee 등은 임의의 복잡한 물체 표면에서 불꽃이 전파되는 것을 시물레이션하였는데, 불꽃의 전파는 물체의 표면에서의 측지곡선을 계산함으로써 시물레이션된다 [13]. Lamorlette와 Foster는 불꽃의 중심을 3차원 매개 변수 곡선으로 표현하는 불꽃의 모델링 방법을 제안하였다 [12]. 불꽃에서 눈에 보이는 부분은 중심 곡선에 대한 원기둥 단면으로 생성되며 절차적인 노이즈와 Kolmogorov 주파수 스펙트럼을 추가해서 불꽃을 복잡하게 보이도록 하였다.

불꽃에 대한 유체 역학에 근거한 연구의 첫 시작은 Stam과 Fume이 제안한 확산 방정식에 근거해서 불과 연기 등과 같은 유체를 시물레이션한 연구이다 [17]. 이 연구에서 확산 방정식의 수치적 해를 구하기 위해서 변형된 불륨 방법을 제안하였으며 곡면 위에서 불꽃이 전파되는 것을 처리하기 위해서 물체의 표면 위에 연료 입자를 분사하고 점화점으로부터 그 입자들을 태우는 방법을 개발하였다. Nguyen 등은 물리학에 근거한 불의 모델링과 애니메이션 방법을 발표하였다 [15]. 이 방법에서는 불을 중심, 불꽃, 연기의 세 구성 요소로 나누어서 모델링한다. 먼저, 중심은 등위 집합으로 표현되는데, 그 경계면은 분출되는 유체의 흐름에 따라서 그 형태가 변형된다. 불꽃은 ghost-fluid 방법을 이용해서 중심의 밖으로 방사되는 유체를 이용해서 표현된다. 그리고 유체의 움직임은 Euler 방정식을 이용해서 제어되며, 온도는 시간-온도 그래프를 통해서 제어된다. 마지막으로 연기는 온도가 특정한 경계 온도 아래로 내려간 유체를 의미한다. 이 연구에서는 불을 렌더링하기 위해서 포톤 매핑(photon mapping)과 광선 진행(ray marching)을 이용한 매우 사실적인 렌더링 방법을 제안하였다. Feldman 등은 연소를 이용해서 불과 폭발을 시물레이션하는 모델을 제시하였다 [6]. 정지된 입자 시스템으로 표현되는 연소의 연료는 Euler 방정식을 이용한 유체 모델을 이용해서 조절된다. 매 시각마다 연료는 연소되며 그 연료를 포함한 셀에

열과 빛을 방사하고, 이를 통해서 불과 폭발이 처리된다. Ihm 등은 화학적 반응을 시물레이션하는 방법을 이용해서 유체를 애니메이션하는 방법을 제안하였다 [8]. 이 방법은 불에 대한 애니메이션 기법은 아니지만, 화학적 반응을 포함했다는 점에서 불과 연관된 연구라고 할 수 있다. Lasasso 등은 불꽃의 전파를 애니메이션하는 방법을 제시하였다 [14]. 이 방법에서는 삼각형 메쉬로 표현된 물체를 격자로 표현된 유체로 변환하는 방법을 통해서 물체 위에서 불꽃이 전파되는 것을 구현하였다.

3. 유체 모델



[그림 1] 유체 모델과 연소 모델의 상호작용

본 연구에서 제시하는 불의 애니메이션 방법은 유체의 움직임을 시물레이션하는 유체 모델과 연소 반응을 시물레이션하는 연소 모델로 구성된다. 두 모델간의 상호 작용은 [그림 1]에서 제시되어 있다. 유체 방정식에 근거해서 불과 그 주위의 유체의 흐름을 제어하는 역할을 수행하는 유체 모델은 고전적인 유체 방정식을 이용해서 설계되는데, 본 연구에서는 낮은 점도를 갖는 Navier-Stokes 방정식을 이용한다 [7]. Navier-Stokes 방정식은 Euler 방정식 [5]에 발산을 처리하는 항이 추가된 방정식으로 더 자연스러운 유체의 움직임을 처리할 수 있다는 장점이 있다. Navier-Stokes 방정식은 다음과 같다.

$$\nabla \cdot \mathbf{u} = 0,$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} - \frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

위의 식에서 \mathbf{u} 는 유체의 속도를 나타내며, t 는 시간을, ν 는 유체의 운동학적 점성, ρ 는 유체의 밀도를, 그리고 \mathbf{f} 는 외부에서 가해지는 힘을 나타낸다. 이 방정식에서 이항력을 나타내는 항은 semi-Lagrangian 방법으로 해를 구한다

[18]. 또한, 외부에서 가해지는 힘은 유체를 상승시키는 부력과 유체를 회전시키는 상세력으로 구성된다. 유체의 밀도와 온도는 다음의 방정식에 의해서 제어된다.

$$\frac{\partial \rho}{\partial t} = -(\mathbf{u} \cdot \nabla) \rho$$

$$\frac{\partial T}{\partial t} = -(\mathbf{u} \cdot \nabla) T - c_T \left(\frac{T - T_{air}}{T_{max} - T_{air}} \right)^4 + \frac{dH}{dt}$$

이 방정식에서 T 는 유체의 온도랄 나타내며, T_{air} 는 상온의 온도, T_{max} 는 최대 온도, c_T 는 유체의 온도를 낮추는 냉각 상수이다. 그리고 dH/dt 는 연소에 의해서 발생한 열을 나타내는 항인데, 이 항의 $H(v)$ 에 대해서는 3.2 절에서 상세히 설명한다.

4. 연소 모델

3장에서는 연료가 산소와 결합하면서 연소물로 변환되고, 그 과정에서 열과 빛을 방사하는 연소 반응을 시뮬레이션하는 연소 모델에 대해서 소개한다. 이를 위해서 연소 반응에서 이용되는 연료의 성질을 표현하는 연료 모델에 대해서 먼저 설명한다.

4.1 연료 모델의 설계

4.1.1 연료의 성질

연료는 매우 다양한 성질을 가지고 있는데, 본 연구에서는 그 중에서 중요하다고 판단되는 다음의 성질들을 이용해서 연료 모델을 설계한다.

- 잠재열

: 연료의 잠재열은 연소 반응 중에 연료가 발생하는 열의 양을 의미한다. 연소 이론에 따르면 서로 다른 화학적 성질을 지닌 연료들은 각각 서로 다른 잠재열을 지니고 있다. 본 논문에서는 유사한 화학적 성분을 지닌 가연성 물질들은 유사한 잠재열을 가지고 있다고 가정한다. 따라서 복합적인 화학 성분으로 구성된 가연성 물질은 다양한 잠재열을 지닌 연료들로 변환되어 연소에 참여하게 되는 것으로 처리한다.

- 연소 속도

: 연료의 반응 속도 역시 연료의 화학적인 성질에 의해

서 결정된다. 본 논문에서는 연소 과정을 이산화된 시각에서 시뮬레이션함으로써 유체 방정식과 같은 환경에서 연소 과정에 대한 시뮬레이션을 수행한다. 따라서 연료의 연소 속도는 각 시각에 연소되는 연료의 양을 의미하는 연료의 연소비율로써 구현된다. 연료가 연소가 수행되는 영역에 투입되면서 높은 연소 속도를 가진 연료는 빠른 속도로 연소되며, 낮은 연소 속도를 가진 연료는 서서히 연소된다.

- 필요 산소량

: 연소 과정에서는 산소가 요구된다. 연료는 이 산소와 결합함으로써 연소물로 변환되며 그에 따른 빛과 열을 방사하기 때문이다. 서로 다른 성분을 가진 연료는 연소 과정에서 각각 다른 양의 산소를 요구한다. 본 논문에서는 이 성질을 이용해서 연소 지역을 조절한다. 낮은 필요 산소량을 가진 연료는 불꽃의 중심부에서 연소되며 높은 필요 산소량을 가진 연료는 불꽃의 변경부에서 연소된다.

- 경계 온도

: 연료는 그 연료의 온도가 특정한 경계 온도 아래의 상태에서는 고체 상태나 액체 상태를 유지한다. 그 경계 온도를 열분해 온도라 하는데, 이 온도 이상의 온도에서 연료는 기체 상태가 되며 연소될 준비가 된다. 또한, 이렇게 열분해된 연료들은 점화 온도라 불리는 경계온도보다 높은 환경에서 연소된다.

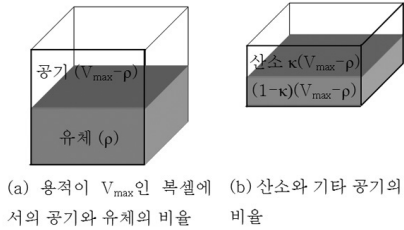
본 논문에서는 유체의 상태를 연료에서 연소물로 변화하는 것으로 표현하며, 위에서 제시한 연료의 여러 가지 성질을 다음과 같이 표현한다. 유체가 N 가지의 서로 다른 연료로 구성된다고 가정할 때, x 번째 연료의 성질을 다음과 같이 표현한다.

- h^x : x 번째 연료의 잠재열
- v^x : x 번째 연료의 연소 속도
- O^x : x 번째 연료의 필요한 산소량
- T^x_1 : x 번째 연료의 열분해 온도
- T^x_2 : x 번째 연료의 점화 온도

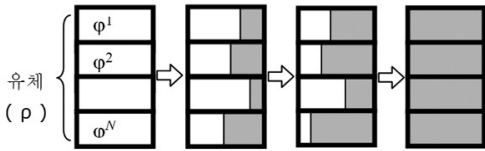
4.1.2 복셀화된 공간에서의 연료의 구현

연료를 유체 방정식이 구현되는 복셀화된 공간에서 구현하기 위한 기본 전제로 본 연구에서는 각 복셀들이 유체 또는 공기로 가득 차 있다고 가정한다. 시뮬레이션을 시작하기 전에 최대 용적이 V_{max} 인 복셀은 공기로 가득 차 있다고

가정될 수 있다. 만약 공기에서의 산소의 비율을 κ 라고 한다면, 그 복셀에 있는 산소의 양은 κV_{\max} 로 계산될 수 있으며, 산소 이외의 다른 기체의 양은 $(1-\kappa) V_{\max}$ 로 계산된다. 복셀의 구성은 [그림 2]에서 제시되어 있다.



[그림 2] 복셀의 구성: 유체 및 공기 (공기는 산소와 기타 기체로 구성됨)



[그림 3] 유체의 상태 변화: 연료에서 연소물로 변화한다.

시물레이션 도중에 복셀은 밀도가 ρ 인 유체로 가득차게 된다. 유체의 밀도 ρ 는 복셀의 체적이 일정한 상황에서는 유체의 질량을 의미하는데, 이 유체는 N 가지의 연료로 구성되어 있으며 각각의 비율은 ϕ^x 로 표시된다. 초기에 연소되기 이전의 유체는 $\phi^x=1.0$ 의 값을 갖는다. 시물레이션 중에 ϕ^x 의 값은 연소가 진행됨에 따라 감소하게 되며, 연소가 완전히 끝난 뒤에는 ϕ^x 는 0의 값을 갖게 된다. 연료의 변화는 [그림 3]에서 제시되어 있다. ϕ^x 는 연료 중에서 x 번째 종류의 연료의 비율을 나타내기 때문에 x 번째 종류의 연료의 양은 $(\phi^x/\phi)\rho$ 로 계산될 수 있다. 이 식에서 ϕ 는 모든 종류의 연료들의 ϕ^x 의 합으로 계산된다. ϕ^x 의 값은 유체 방정식에서의 밀도를 나타내는 방정식을 이용해서 표현된다.

$$\frac{\partial \phi^x}{\partial t} = -(\mathbf{u} \cdot \nabla) \phi^x, \text{ for } 1 \leq x \leq N$$

4.2 연소 모델의 구현

유체 시물레이션의 매 시각에 모든 복셀에서 연소 반응이 시물레이션된다. 연소 과정을 간략하게 표현하면 다음과 같다.

- [1] 복셀 내에서 온 연료에 대해서 이용가능한 산소량과 필요한 산소량을 측정한다.
- [2] 각 연료들에 대해서 다음의 과정을 통해서 발생되는 열의 양을 계산한다.
 - [2.1] 연소될 연료의 양을 계산한다.
 - [2.2] 연소된 연료로부터 발생하는 열의 양을 계산한다.
 - [2.3] 연소에 따라서 소모된 연료와 남은 연료의 양을 계산한다.
- [3] 복셀에서 발행한 열의 총 합을 계산한다.

이 과정은 다음의 단계들을 통해서 구현된다.

- [1] 각 복셀을 v 로 표현하면, v 복셀에서 x 번째 연료가 사용할 수 있는 산소의 양과 x 번째 연료가 완전히 연소하기 위해서 필요한 산소의 양을 각각 $O_a^x(v)$ 와 $O_r^x(v)$ 로 표시할 수 있다. $O_a^x(v)$ 를 계산하는 과정에서 본 논문에서는 복셀 v 에서의 x 번째 연료가 사용할 수 있는 산소의 양은 복셀 v 에 있는 연료에서 x 번째 연료가 차지하는 비율이라고 가정한다. 따라서 $O_a^x(v)$ 는 다음과 같이 계산된다.

$$O_a^x(v) = O(v) \frac{\phi^x(v)}{\Phi}, \text{ for } \Phi = \sum_{n=1}^N \phi^n(v)$$

이 식에서 복셀 v 에서의 사용 가능한 산소의 양인 $O(v)$ 는 $\kappa(V_{\max}-\rho(v))$ 로 계산된다. 즉, 복셀 v 의 전체 용적에서 연료가 차지하는 것을 뺀 용적을 공기의 용적이라고 계산하고, 이 공기의 용적에서 산소의 비율을 곱한 값을 산소의 양이라고 계산하는 것이다. 그리고 x 번째 연료의 단위량이 완전히 연소하는데 필요한 산소량을 O_r^x 이라고 표시할 경우에 $O_r^x(v)$ 는 다음과 같이 계산된다.

$$O_r^x(v) = O_r^x \phi^x(v)$$

- [2] 각 복셀에서 발생하는 열의 양은 그 복셀에서 각각의 연료들이 연소하면서 발생하는 열의 합으로 계산된다. 각 연료들의 연소로부터 발생하는 열을 계산하기 전에, 각각의 연료들에 대해서 각 시물레이션 시각마다 복셀 v 에서 연소될 수 있는 연료의 양을 계산할 필요성이 있다. 복셀 v 에서 연소될 수 있는 x 번째 연료의 양을 나타내는 $\phi^x(v)$ 는 그 연료의 연소 속도와 복셀 v 에서 이용할 수 있는 산소의 양에 의해서 결정된다. 따라서 그 복셀에서 이용할 수 있는 산소의 양이 필요한 산

소의 양보다 많은 경우와 적은 경우로 나누어서 처리한다. 만약 이용할 수 있는 산소의 양이 필요한 양보다 더 많은 경우에는 $\phi^x(v)$ 는 연소 속도와 연료의 양을 곱한 값으로 계산된다.

$$\phi^x(v) \leftarrow \gamma^x \phi^x(v)$$

위의 식에서 γ^x 는 x번째 연료의 연소 속도이다. 이 값이 1인 경우에는 완전히 연소를 하며, 0인 경우에는 전혀 연소하지 않는다. 만약 이용 가능한 산소의 양이 필요한 양보다 더 적은 경우에는 $\phi^x(v)$ 는 이용 가능한 산소의 양에 의해서 다음과 같이 제한된다.

$$\phi^x(v) \leftarrow \frac{O_a^x(v)}{O_r^x(v)} \gamma^x \phi^x(v)$$

따라서 x번째 연료가 복셀 v에서 연소되어 발생하는 열의 양을 $H^x(v)$ 라고 표시한다면, 이 값은 다음의 식에 의해서 계산된다.

$$H^x(v) = h^x \phi^x(v) \rho_v \delta t$$

이 식에서 h^x 는 x번째 연료의 잠재열을 나타내며 δx 는 시뮬레이션을 하기 위한 시간 간격을 나타낸다. 최종적으로 연소 후에 남은 연료의 양은 다음과 같이 계산된다

$$\phi^x = \phi^x - \phi^x(v), \text{ for } 1 \leq x \leq N$$

[3] 복셀 v에서 발생하는 열의 총합을 H(v)로 표시하면 이 값은 다음과 같이 각 연료로부터 발생하는 열의 합으로 계산된다.

$$H(v) = \sum_{i=1}^N H^x(v)$$

이 값은 3장의 방정식 (1)에서 사용된다.

5. 가시화 알고리즘

본 논문에서는 불과 불이 주변을 밝히는 효과를 가시화하기 위해서 다음의 단계를 수행한다.

[1] 온도를 색에 대응

: 불의 온도에 대해서 적절한 색을 대응하기 위해서 본

논문에서는 온도-색 그래프를 제시한다.

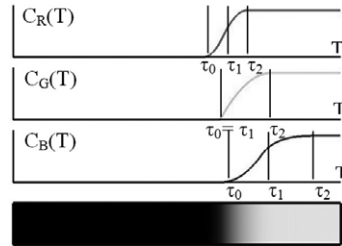
[2] 포톤을 방사하고 관리

: 온도가 특정한 경계값보다 큰 복셀에서는 다수의 포톤이 방사된다. 이 포톤의 색은 온도-색 그래프를 통해서 결정된다. 방사된 포톤은 공간을 떠돌다가 적절한 위치에 저장된다. 이렇게 저장된 포톤들은 적절한 자료 구조를 통해서 관리된다.

[3] 통합

: 공간에 저장된 포톤들은 광선 탐사 방법을 적용해서 통합될 수 있으며, 이렇게 통합된 포톤들의 값을 이용해서 렌더링하는 영상의 각 픽셀값을 결정한다.

5.1 온도-색 그래프



[그림 4] 온도-색 그래프. 붉은색, 녹색, 푸른색에 대한 그래프와 그 결과로 얻어진 불의 색 스펙트럼

흑체 복사 이론에 따르면 온도가 높은 유체의 색은 그 온도에 의해서 결정된다. 본 연구에서는 이러한 관계를 이용해서 불의 색을 결정하기 위해서 온도-색 그래프를 제시한다 (그림 4). 사용자는 이 온도-색 그래프를 제어함으로써 자신이 원하는 불의 색을 표현할 수 있다. 이 온도-색 그래프는 음함수형 곡면에서 이용되는 필드 함수를 이용해서 설계된다 [2].

$$f(d) = \begin{cases} 1 - \frac{(3d^2)^2}{p + (4.5 - 4p)d^2}, & 0 < d \leq \frac{1}{2} \\ \frac{(1 - d^2)^2}{0.75 - p + (1.5 + 4p)d^2}, & \frac{1}{2} < d \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

위의 식에서 p는 함수의 기울기를 조절하는 매개 변수이다. 이 필드 함수는 세 가지 기본 색에 대해서 각각 설계되며, 최종적인 불의 색은 이 세 가지 기본 색에 대한 필드 함수의 값을 합함으로써 결정된다. 필드 함수를 사용하는 가

장 큰 장점은 필드 함수의 매개 변수 p 를 이용해서 함수의 곡선의 기울기를 제어할 수 있다는 점이다. 온도를 색에 대응시키는 과정에서 각 색에 대해서 τ_0, τ_1, τ_2 이라는 세 가지 경계 온도를 제시한다. 이 각 경계값은 각각 필드 함수의 입력인 d 의 0, 0.5, 1인 값에 대응된다. 따라서 온도 T 가 어떤 값을 갖느냐에 따라서 그 T 에 해당되는 색이 $C(T)$ 는 다음과 같이 결정된다.

$$C(T) = \begin{cases} 0, & \text{if } T < \tau_0 \\ f(d_1), & \text{if } \tau_0 \leq T < \tau_1 \\ f(d_2), & \text{if } \tau_1 \leq T < \tau_2 \\ 1, & \text{otherwise} \end{cases}$$

위의 식에서 d_1 과 d_2 는 다음과 같다.

$$d_1 = 1 - \frac{1}{2} \left(\frac{T - \tau_0}{\tau_1 - \tau_0} \right)$$

$$d_2 = \frac{1}{2} \left(\frac{T - \tau_1}{\tau_2 - \tau_1} + 1 \right)$$

온도-색 그래프의 한 예는 그림 3에서 제시되고 있다. 이 그래프는 세가지 기본 색인 붉은 색(R), 녹색(G), 푸른 색(B)에 대한 각각의 그래프 ($C_R(T)$, $C_G(T)$, $C_B(T)$)로 구성된다. 그래프의 아랫쪽에는 이 그래프로부터 얻어지는 색의 스펙트럼이 제시되어 있다. 사용자는 τ_0, τ_1, τ_2 의 값들과 그래프의 p 값을 제어해서 그래프의 형태를 조절할 수 있으며, 이를 통해서 다양한 불의 색을 표현할 수 있다.

5.2 포톤의 방사와 관리

불과 불에 의한 조명 효과를 가시화하기 위해서 본 논문에서는 포톤 매핑 알고리즘을 효율적으로 개선하는 방법을 제안한다. 포톤을 방사하는 과정은 다음과 같다.

[1] 포톤을 생성

: 불의 내부에 놓인 복셀은 포톤을 생성한다. 생성된 포톤은 색과 강도를 갖는다. 생성된 포톤의 색은 5.1절에서 제시한 온도-색 그래프에 의해서 결정된다.

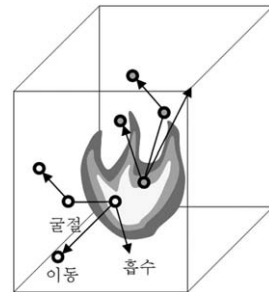
[2] 포톤을 방사

: 불의 내부에 있는 복셀에서 생성된 포톤을 외부로 향해서 방사된다. 포톤은 불의 외부로 향해서 방사되는 것으로 제한하는데, 이 방향은 다음과 같이 결정된다. 먼저 임의의 방향을 설정한다. 그리고 선택된 방향으로

로 복셀의 한 변의 길이만큼 떨어진 점을 샘플링한다. 만약 샘플링된 점의 온도가 복셀의 온도보다 더 높다면, 그 방향은 불의 내부를 향한 방향이라고 간주하고 포기되며, 새로운 방향을 다시 선택해서 위의 과정을 반복해서 불의 외부로 향한 방향을 선택한 다음, 그 방향으로 포톤을 방사한다.

[3] 포톤의 여행

: 방사된 포톤은 공간을 여행한다. 이 포톤의 여행은 포톤의 임의의 한 점에서 미리 정의된 거리만큼 이동한 다음 다음의 세 가지 작용 중에 한 가지를 처리하는 과정을 반복해서 수행된다. 그 세 가지 작용은 이동, 굴절, 그리고 흡수이다 (그림 5). 이 세 가지 작용 중에 어떤 것을 선택할 지는 러시아 룰렛 알고리즘을 통해서 결정된다. 포톤이 새로운 점으로 이동할 때마다 포톤의 강도는 감소된다. 포톤이 각 점에서 이동하거나 굴절될 경우에는 그 점에서 포톤을 저장하며, 흡수되는 경우에는 더 이상 포톤을 여행시키지 않는다. 이러한 포톤의 여행은 포톤의 강도가 특정한 경계값 아래로 떨어질 때까지 계속 수행된다.



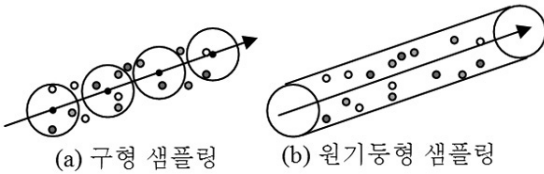
[그림 5] 불에서 방사된 포톤의 여행에서 발생할 수 있는 세 가지 사건: 굴절, 이동 및 흡수

[4] 포톤의 저장

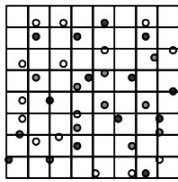
: 포톤은 어떤 곳에서 저장되어있느냐에 따라서 다음의 두 가지로 구분될 수 있다. 공간 중에 저장된 포톤을 공간 포톤이라고 하고, 물체의 표면에서 저장된 포톤을 표면 포톤이라고 한다. 기존의 연구에서는 이 포톤을 k-d 트리 구조를 이용해서 저장하였는데, 본 연구에서는 복셀화된 공간에 저장하는 방법을 제시한다. 그 이유는 유체 방정식의 수치적 해법을 위해서 이미 공간이 복셀로 분할되어 있기 때문에 그 구조를 이용함으로써 새로운 자료 구조를 도입하는 복잡함을 피할 수

있다. 또한, 복셀화된 공간에 포톤을 저장할 경우, 포톤의 위치를 계산하는 식이 k-d 트리 구조보다 더 단순해 진다는 장점이 있다.

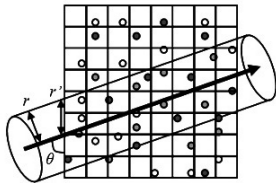
5.3 포톤의 통합



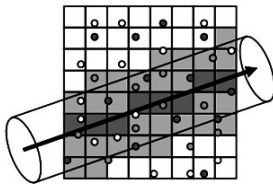
[그림 6] 두가지 포톤 샘플링 방법의 비교



(a) 포톤들이 복셀화된 공간에 저장됨



(b) 시선으로부터의 반직선으로부터 거리 r 이내의 포톤들을 샘플링



(c) 반직선에서 거리 r 이내의 복셀을 샘플링함

[그림 7] 복셀화된 공간을 이용한 포톤의 샘플링: 반직선에서 거리 r 이내의 복셀을 샘플링하고, 그 복셀 내부의 포톤들을 샘플링함.

불과 그 효과를 렌더링한 영상의 각 픽셀의 색은 시점으로부터 불을 향해서 투사한 반직선 주변에 놓인 포톤들의 색들을 통합함으로써 결정된다. 기존의 연구에서는 구형 샘플링을 이용해서 포톤들을 통합하였다. 즉, 반직선 위의 몇몇의 점들을 샘플링하고, 그 샘플링된 점에서 적당한 크

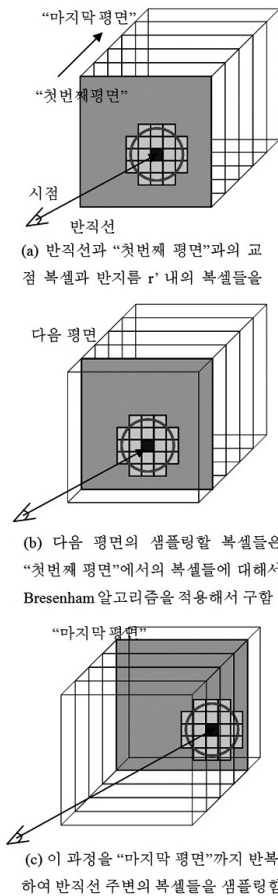
기의 반지름을 가진 구를 생성하고, 이 구의 내부에 포함된 포톤만을 통합에 이용하는 방법을 적용하였다. 이러한 방법은 비교적 빠르게 통합에 이용될 포톤을 결정할 수 있지만, 반직선 주변의 포톤들이 고르게 통합에 참여하기 힘들다는 단점이 있으며, 또한 포톤들과 샘플링된 점들 사이의 거리를 계산해야 한다는 계산상의 부담이 크게 작용한다. 이를 개선하기 위해서 본 논문에서는 원기둥형 샘플링 방법을 제안한다 (그림 6). 포톤의 통합 과정에서 요구되는 계산량의 대부분은 포톤과 반직선 사이의 거리를 계산하는 연산이 차지한다. 따라서 반직선 주변의 포톤들만을 샘플링하여 이 포톤들과 반직선 간의 거리를 구하는 효율적인 알고리즘들이 제시되는데, 기존의 포톤 매핑 방법에서는 k-d 트리 구조를 이용하고 있다 [10]. 본 연구에서는 유체 시물레이션에서 사용되는 복셀 구조를 이용한다. 그 과정은 [그림 7]에서 제시되어 있다. 이러한 복셀화된 공간 구조를 이용할 경우에는 Bresenham의 알고리즘을 응용한 효율적인 샘플링이 가능하다는 장점이 있다.

5.3.1 반직선 주변의 복셀을 샘플링

준비 단계로써, 포톤을 수집할 영역을 나타내는 변수인 샘플링 거리 r을 설정한다. 그리고 가시화에서의 시점은 유체가 시물레이션되는 복셀들의 외부에 존재한다고 가정한다. 반직선 주변의 복셀을 샘플링하는 과정은 다음과 같이 수행된다 (그림 8).

- [1] 위의 가정에 의해서 시점으로부터 출발하는 반직선이 복셀들 중에서 처음으로 만나는 평면을 “첫번째 평면” (first plane)이라고 표시한다면, 이 평면은 다음의 여섯 평면들 중의 하나이다 $\{x = 0, x = N_x - 1, y = 0, y = N_y - 1, z = 0, z = N_z - 1\}$. 여기서 N_x, N_y, N_z 는 복셀의 x, y, z 차원을 나타낸다.
- [2] “첫번째 평면”에서의 복셀들 중에서 반직선과 처음으로 만나는 “첫번째 복셀” (first voxel)을 계산한다. 그리고 “첫번째 평면”의 복셀들 중에서 반지름이 r' 이내에 들어있는 모든 복셀들을 샘플링한다. 이 r' 은 $r \cos \theta$ 로 계산되는데, θ 는 반직선과 “첫번째 평면” 사이의 각이다. “첫번째 평면”에서 반지름 r' 내에 들어있는 복셀들은 다음 평면에서의 복셀 샘플링에 이용된다.
- [3] 다음 평면에서 샘플링할 복셀을 결정하기 위해서 Bresenham 알고리즘을 이용한다 (그림 8). 본 논문에서

서는 임의의 기울기를 가진 직선이 지나가는 픽셀을 직선과 픽셀의 교차를 계산하지 않고 효율적으로 찾아낼 수 있는 기존의 Bresenham 알고리즘을 3차원으로 확장하여 “첫번째 평면” 위에서 샘플링된 복셀들을 지나가는 시선 방향의 반직선이 지나게 될 다음 평면에서의 복셀을 복셀과 반직선의 거리를 계산하지 않고 찾아낼 수 있다. 따라서 이 방법은 “첫번째 평면”에서만 복셀과 반직선 간의 거리를 계산하고, 다음 평면들에 대해서는 계산을 하지 않기 때문에 샘플링에 걸리는 시간 복잡도를 $O(n^3)$ 에서 $O(n^2)$ 로 감소시킨다.



[그림 8] Bresenham 알고리즘을 이용한 복셀들의 샘플링

5.3.2 원기둥형 샘플링

원기둥형 샘플링 방법은 반직선을 중심으로 특정한 반지름 내에 있는 포톤들을 통합하는 것으로 원기둥 내의 포톤을 통합하는 것과 유사하다. 이 방법의 장점은 포톤과 반직

선과의 거리를 계산하지 않고 포톤들을 통합할 수 있다는 점이다. 이러한 특징 때문에 기존의 구형 샘플링 방법에 비해서 크게 효율적인 계산이 가능하다는 장점이 있다.

5.3.3 통합

각 픽셀의 색은 반직선을 축으로 특정한 반지름으로 형성된 원기둥 내부의 포톤의 색을 통합함으로써 얻어진다. 여기서 반직선은 시선 방향의 벡터를 의미한다. $C(p_i)$ 를 포톤 p_i 의 색, $d(p_i)$ 를 포톤 p_i 와 반직선간의 거리, n 을 샘플링된 포톤의 수라고 가정할 때, 해당되는 픽셀의 밝기 I 는 다음의 식에 의해서 계산된다.

$$I = \frac{1}{n} \sum_{i=1}^n e^{-k_1 d^2} d^{-k_2} C(p_i) + e^{-k_3 l^2} C(q)$$

이 식에서 k_1 과 k_2 는 각각 시점으로부터의 거리와 시선 방향의 반직선으로부터의 거리가 밝기에 영향을 미치는 것을 제어하기 위한 변수이다.

5.4 불의 조명 효과

불의 조명 효과는 불이 주변 환경을 밝히는 효과를 나타내는데, 이는 불에서 방사된 포톤들 중에서 물체의 표면에 저장된 포톤들을 통합함으로써 처리된다. 이 효과는 위의 식 (3)에서 제시된 통합 방정식은 다음과 같이 변형해서 구현할 수 있다.

$$I = \frac{1}{n} \sum_{i=1}^n e^{-k_1 d^2} d^{-k_2} C(p_i)$$

이 식에서 q 는 시선 방향의 벡터와 물체와의 교점이며 $C(q)$ 는 q 에서의 색, 그리고 $l(q)$ 는 시점과 q 와의 거리이다. 식 (4)에서 거리가 멀어짐에 따라 $C(q)$ 의 값도 감소하게 되는데, 이를 처리하기 위해서 k_3 라는 변수를 추가하였다. $C(q)$ 는 다음의 식에 의해서 계산된다.

$$C(q) = C_q(P(q) + A)$$

이 식에서 C_q 는 물체의 q 점에서의 색을 나타내며 $P(q)$ 는 샘플링 거리 내에 있는 표면 포톤들의 색을 통합한 값이다. 또 A 는 자연광의 색을 나타낸다.

6. 구현 및 결과

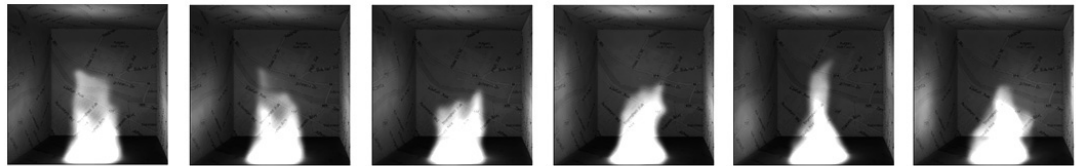
본 논문에서 제시된 알고리즘을 Pentium 3.06 GHz의 속도를 가지는 CPU와 2.0 GByte의 주기억 용량을 가진 개인용 컴퓨터에서 구현되었다. 유체 시뮬레이션을 위해서는 공간을 150 x 150 x 150 크기의 복셀로 분할하였으며, 시뮬레이션에서의 시간 간격은 0.00005초이다.

본 논문의 결과는 두 가지 형태로 제시될 수 있다. 첫 번째 결과는 몇 가지 서로 다른 조건에서의 시뮬레이션이다. 즉, 연료가 불에 공급되는 과정이 고정된 것인지, 이동하는 것인지에 대한 조건을 바꾸면서 불에 대한 애니메이션을 생성한다. 본 논문의 결과로는 연료가 고정된 상태에서 공급되는 애니메이션과 회전하는 토러스 형태로 공급되는 애니메이션, 그리고 크기가 진동하는 토러스의 형태로 공급되는 애니메이션을 생성한다. 이 결과는 [그림 9]에서 제시되어 있다.

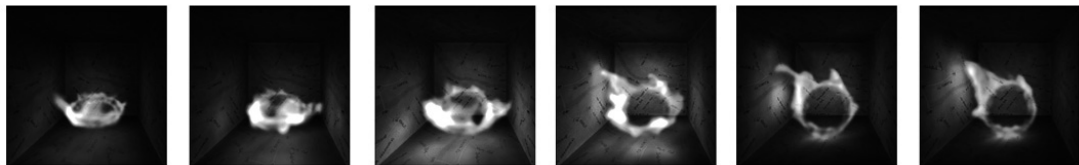
두 번째 결과는 동일한 애니메이션 결과에 대해서 각각 다른 렌더링 기법을 적용하는 것이다. 본 논문에서는 (i) 불에서 포톤을 외부로 방사하지 않고 온도-색 그래프에 근거해서 생성된 포톤을 통합한 단순한 렌더링 효과, (ii) 배경과 불을 합성한 효과, (iii) 주변의 빛이 없는 상황에서의 광원

으로써의 렌더링 효과, (iv) 포톤을 주변으로 방사하여 주변 공기를 밝히는 렌더링 효과, (v) 배경 빛이 있는 상황에서의 광원으로써의 렌더링 효과 등을 다양하게 렌더링한다. 이 결과는 [그림 10]에서 제시되어 있다.

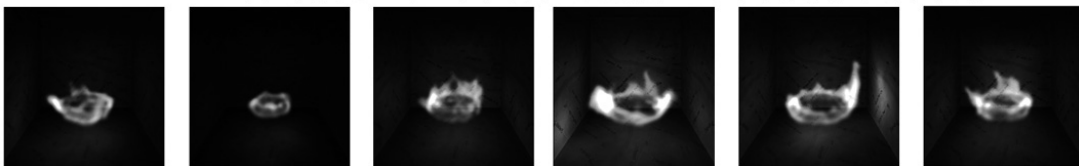
본 논문의 결과와 유사한 결과를 제시하는 중요한 기존 연구로는 [15]와 [6]을 들 수 있다. [15]는 유체 시뮬레이션을 이용해서 불의 애니메이션을 구현하는 최초의 연구로 사실적인 불의 애니메이션 및 가시화를 가능하게 하였다. 본 논문의 유사한 점은 유체 방정식을 이용하였고, 포톤 매핑 방법을 적용해서 가시화를 처리하였다는 점이고, 다른 점은 [15]에서는 박염 이론(thin flame theory)에 근거해서 불꽃의 중심을 등위 집합(level-set)으로 표현하였다는 점이다. [6]은 입자 시스템을 유체 방정식에 도입한 복합적 방법론을 제시하였으며, 이를 통해서 폭발 등과 같은 급격한 속도의 연소 반응을 처리할 수 있다는 점이 특징이다. 다만, [6]의 경우에는 IRT라는 Pixar 사의 Renderman 과 유사한 렌더링 라이브러리를 사용하였는데, 그 결과가 미흡하다는 단점이 있다.



(a) 고정된 연료의 공급원으로부터 생성된 불의 애니메이션

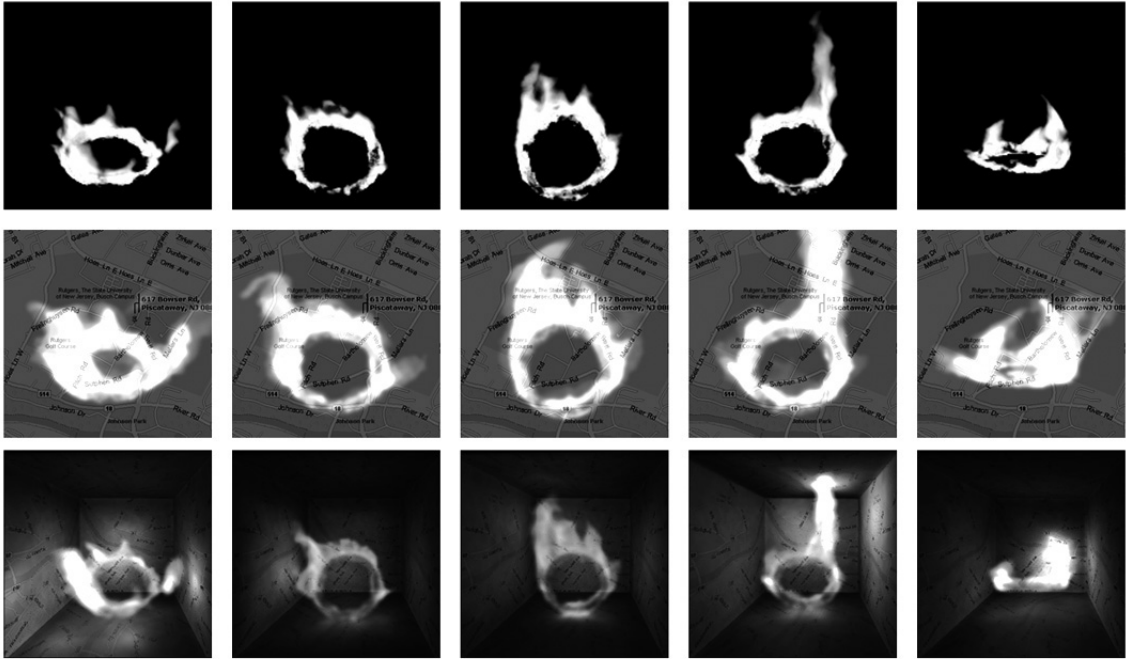


(b) 회전하는 토러스 형태의 연료의 공급원으로부터 생성된 불의 애니메이션

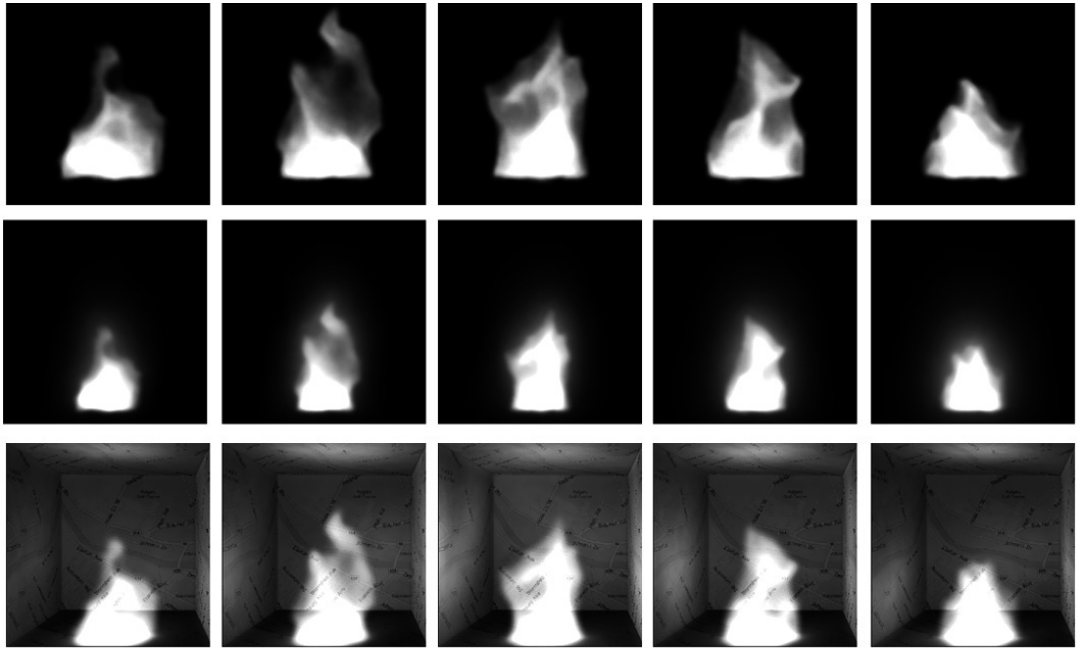


(c) 팽창/수축하는 토러스 형태의 연료의 공급원으로부터 생성된 불의 애니메이션

[그림 9] 세 가지 형태의 애니메이션 결과



(a) 회전하는 토러스 형태의 불꽃을 여러가지로 렌더링한 결과: 단순한 포톤 통합을 통한 렌더링 (윗줄), 배경과 합성한 렌더링 (중간), 그리고 어두운 환경에서의 광원으로써의 렌더링 (아랫줄).



(b) 고정된 연료의 공급원으로부터의 불의 애니메이션에 대한 다양한 렌더링: 단순한 포톤의 통합에 의한 렌더링 (윗줄), 불에서 나온 포톤이 주변 공기를 밝히는 렌더링 (중간), 배경 빛이 있는 상황에서의 조명 효과 렌더링 (아랫줄)

[그림 10] 다양한 렌더링 효과

7. 결론 및 향후 연구 방향

본 논문에서는 연소 과정의 시뮬레이션을 기존의 유체 시뮬레이션에 통합함으로써 불을 애니메이션하는 방법론을 제시하고, 포톤 매핑 알고리즘을 효과적으로 개선한 가시화 알고리즘을 개발하여 불을 다양하게 렌더링하는 방법을 구현하였다.

본 논문에서의 향후 연구 방향은 아직 미흡하게 제시되고 있는 게임 개발 환경에 본 논문에서 제시한 불의 애니메이션 기법을 적용하는 것이다. 이를 위해서 최근 게임 개발 환경에서 각광받고 있는 GPU 프로그래밍 환경에서 본 논문을 효율적으로 구현하는 방법론을 제시하고자 한다.

참고 문헌

- [1] P. Beaudoin, S. Paquet, and P. Poulin, "Realistic and controllable fire simulation", In *Proceedings of Graphics Interface 2001*, pp. 159-166, 2001.
- [2] C. Blanc and C. Schlick, "Extended field functions for soft objects", In *Proceedings of Implicit Surfaces 1995*, pp. 21-35, 1995.
- [3] R. Bukowski and C. Sequin, "Interactive simulation of fire in virtual building environments", In *Proceedings of SIGGRAPH 1997*, pp. 35-44, 1997.
- [4] N. Chiba, S. Ohkawa, K. Muraoka, and M. Miura, "Two-dimensional visual simulation of flames, smoke and the spread of fire", *Journal of Visualization and Computer Animation*, Vol. 5 No. 1, pp. 37-54, 1994.
- [5] R. Fedkiw, J. Stam, and H. Jensen, "Visual simulation of smoke", *Proceedings of SIGGRAPH 2001*, pp. 15-22, 2001.
- [6] B. Feldman, J. O'Brien, and O. Arikan, "Animating suspended particle explosions", *ACM Transactions on Graphics*, Vol. 22, No. 3, pp. 708-715, 2003.
- [7] N. Foster and D. Metaxas "Modeling the motion of a hot, turbulent gas", *Proceedings of SIGGRAPH 1997*, pp. 181-188, 1997.
- [8] I. Ihm, B. Kang, and D. Cha, "Animation of realistic gaseous fluids through chemical kinetics", In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation 2004*, pp. 203-212, 2004.
- [9] M. Inakag, "A simple model of flames", In *Proceedings of Computer Graphics International 1989*, pp. 71-81, 1989.
- [10] J. Jensen, "Global illumination using photon maps. Proceedings of Rendering Techniques", *Proceedings of the 7th Eurographics Workshop on Rendering*, pp. 21-30, 1996.
- [11] B. Kang, I. Ihm, and C. Bajaj "Extending the photon mapping method for realistic rendering of hot gaseous fluids", *Computer Animation and Virtual Worlds*, Vol. 16 No. 3-4, pp. 353-363, 2005.
- [12] A. Lamorlette and N. Foster, "Structural modeling of flames for a production environment", *ACM Transactions on Graphics*, Vol. 21, No. 3, pp. 729-735, 2002.
- [13] H. Lee, L. Kim, M. Meyer, and M. Desbrun, "Meshes on fire", In *Proceedings of Eurographics Workshop on Computer Animation and Simulation*, pp. 75-84, 2001.
- [14] F. Losasso, G. Irving, E. Guendelman, and R. Fedkiw, "Melting and burning solids into liquids and gases", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 12, No. 3, pp. 343-352, 2006.
- [15] D. Nguyen, R. Fedkiw, and H. Jensen, "Physically based modeling and animation of fire", *ACM Transactions on Graphics*, Vol. 21, No. 3, pp. 721-728, 2002.
- [16] C. Perry and R. Picard, "Synthesizing flames and their spreading", In *Proceedings of Fifth Eurographics Workshop on Animation and Simulation 1994*, pp. 105-117, 1994.
- [17] J. Stam and E. Fiume, "Depicting fire and other gaseous phenomena using diffusion processes", In *Proceedings of SIGGRAPH 1995*, pp. 129-136, 1995.
- [18] J. Stam, "Stable fluids", In *Proceedings of SIGGRAPH 1999*, pp. 121-128, 1999.



민 경 하 (Kyungha Min)

1988-1992. 한국과학기술원 과학기술대학 전산학과 학사
 1992-1994. 포항공과대학교 컴퓨터공학과 석사
 1994-2000. 포항공과대학교 컴퓨터공학과 박사
 2002-2006. 미 Rutgers 대학교 박사후 연구원
 2006 ~ 상명대학교 디지털미디어학부 전임강사