

비행시뮬레이션에서 프레임율과 현실감을 고려한 계층형 구름 구현 방안

강석윤 · 김기일*

Implementation of Layered Clouds considering Frame Rate and Reality in Real-time Flight Simulation

Seok-yeon Kang · Ki-il Kim*

Department of Informatics, Engineering Research Institute, Gyeongsang National University, Jinju 660-701, Korea

요 약

비행 시뮬레이션 소프트웨어 개발 시 구름효과 구현을 위한 대표적인 연구로는 파티클 시스템을 이용한 구름 모델링과 텍스처 맵핑을 이용한 구름판 모델링이 있다. 하지만 전자의 경우 낮은 프레임율을 야기하는 반면, 후자의 경우는 비행 시 현실감이 떨어지는 문제가 있다. 이를 해결하기 위해 본 논문에서는 구름판 모델링 방법에 현실적 효과를 더 할 수 있도록 카메라에 안개효과를 적용하는 방안을 제안한다. 제안한 방안을 OpenSceneGraph를 사용한 대용량의 맵 데이터베이스 환경에서 실험 한 결과 기존의 구름판 모델링 방법에 비해 1~2Hz 정도로써 프레임은 거의 차이가 없었고, 현실적 효과 측면에서는 파티클을 이용한 구름 모델링 방법과 비슷하게 자연스럽게 현실적인 효과를 보였다.

ABSTRACT

There are two main technologies to implement cloud effect in flight simulator, cloud modeling using particle system and texture mapping. In former case, this approach may cause a low frame rate while unrealistic cloud effect is observed in latter case. To Solve this problem, in this paper, we propose how to apply fog effect into camera to display more realistic cloud effect with high frame rate. The proposed method is tested with massive terrain database environment through implemented software by using OpenSceneGraph. As a result, compared to texture mapping method, the degree of difference on frame rate is 1 or 2Hz while the cloud effect is significantly improved as realistic as particle system.

키워드 : 비행 시뮬레이터, 계층형 구름, 프레임율, 현실감, OpenSceneGraph

Key word : Flight simulator, Layered clouds, Frame rate, Reality, OpenSceneGraph

접수일자 : 2013. 08. 27 심사완료일자 : 2013. 09. 16 게재확정일자 : 2013. 10. 01

* **Corresponding Author** Ki-Il Kim (E-mail:kikim@gnu.ac.kr, Tel:+82-55-772-1373)

Department of Informatics, Engineering Research Institute, Gyeongsang National University, Jinju 660-701, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2014.18.1.72>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

항공기 조종과 같은 모의훈련을 위한 비행시뮬레이션 소프트웨어를 개발함에 있어서 파일럿에게 실제와 같은 비행 훈련 환경을 제공하기 위해서 기상이나 충돌과 같은 효과가 필요하다. 그 중 구름 효과의 구현은 실시간 애플리케이션에서 사용자의 몰입도를 높이고, 자연스러운 비행 장면의 생성에 중요한 역할을 한다[1].

구름 효과는 구름의 모델링 방법에 따라 프레임율과 현실감에 영향을 받기 때문에 시뮬레이션의 용도에 따라 적절한 모델링 방법을 선택해야 한다. 구름을 모델링하는 대표적인 방법에는 파티클 시스템을 이용한 모델링 방법과 텍스처 맵핑을 이용한 구름판 모델링 방법이 있다. 파티클 시스템을 이용한 모델링 방법의 경우 구름 입자를 구름의 모양에 맞춰 모델링하는 기법으로 구름을 구성하는 모든 입자들에 대해 그 물리적인 변화 과정을 추적하여 시뮬레이션 함으로써 정확하게 구름의 형태를 모델링할 수 있다고 하더라도 구름의 규모가 매우 크기 때문에 오랜 계산시간으로 인한 성능저하가 발생한다. 이에 반해 텍스처 맵핑을 이용한 구름판 생성 방법의 경우 구름 입자 연산 시뮬레이션 과정이 필요 없기 때문에 상대적으로 빠른 모델링이 가능하고 물리적인 변수의 조작을 통해 모델링 될 구름의 형태를 예측하는 것이 아니라 구름판의 텍스처 조작을 통해 구름의 형태를 지정할 수 있기 때문에 물리기반 모델링에 비해 선호하기는 하나 매우 낮은 현실감으로 인한 문제가 있다.

본 논문에서는 비행시뮬레이션 소프트웨어 상에서 대용량의 맵 데이터베이스 환경에서도 끊어짐이 없고 자연스러운 비행을 할 수 있도록 앞서 제시한 두 개의 방법의 문제점을 개선하는 방안을 제안한다. 제안하는 방안은 렌더링 속도가 높은 구름판 모델링 방법을 사용하여 두 개의 구름판을 만들고 현실감을 높이기 위하여 특정 구간에서 카메라에 안개효과를 적용하여 구현하였다. 실험은 오픈소스 라이브러리인 OSG와 대용량의 지형 데이터베이스 환경에서 실험하였고, 기존의 구현 방법과 비교 분석함으로써, 프레임율과 현실성 개선에 대한 검증은 제시한다.

본 논문의 구성은 다음과 같다. 1장에서는 본 논문의 배경을 설명하였고 2장에서는 모델링 방법 및 관련 예제 소개하고 3장에서는 구현을 위한 소프트웨어 구조

및 구현 방법을 설명하였고, 4장에서는 실험결과 분석 및 구현 검증을 보여주고 마지막 5장에서는 결론 및 향후 연구를 제시한다.

II. 구름 모델링 관련 연구

본 장에서는 기존의 시뮬레이션 소프트웨어에서 구름 모델링 시 쓰이는 일반적인 두 가지 방법을 소개하고, 그에 관련된 예제를 소개한다.

2.1. 파티클을 이용한 구름 모델링 방법

많은 입자들의 애니메이션을 표현하거나 유동적인 장면을 시뮬레이트하기 위한 그래픽 객체를 생성할 때 사용하는 파티클 시스템은 기존의 렌더링 기술들보다 시뮬레이션 소프트웨어에서 더 쉽게 렌더링 할 수 있도록 도와주는 컴퓨터 그래픽 기술이다[2]. 기존의 파티클 시스템을 사용하여 가상환경에서의 구름 모델링이나 실시간 구름 모델링에 관한 연구들이 있다[3-6]. 이 연구들에 따르면 입자를 사용하여 객체의 움직임이나 모양을 가지기 때문에 매우 현실적인 렌더링이 가능하다. 그렇기 때문에 그림 1과 같은 구름을 모델링 할 때에도 파티클 시스템을 사용하면 쉽게 현실적인 구름효과를 구현 할 수 있다.

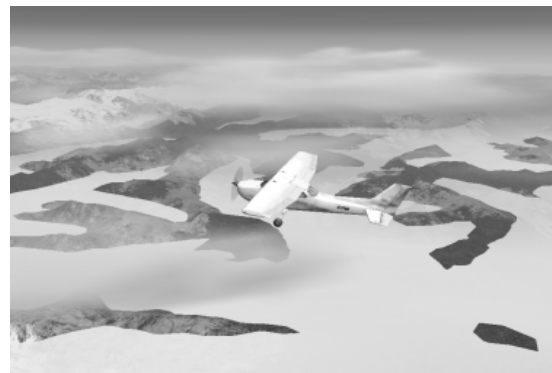


그림 1. Microsoft Flight simulator의 구름효과
Fig. 1 Cloud effect of Microsoft Flight simulator

하지만 파티클 시스템의 단점 중 하나는 하나의 입자가 가지는 물리적인 애니메이션에 의한 연산으로 인해 그 수가 많아지면 렌더링 속도에 많은 영향을 미치게

된다는 것이다. 특히 구름효과의 경우는 많은 수의 입자를 동시에 렌더링 해야 하기 때문에 높은 렌더링 속도와 대용량 데이터베이스를 요구하는 실시간 시뮬레이션 소프트웨어에서 사용하기에는 부적합하다.

2.2. 파티클을 이용한 구름 모델링 방법

텍스처 맵핑은 텍스처의 표면 또는 컴퓨터 그래픽의 색깔, 또는 3D 모델과 같은 세부 추가 사항을 위한 하나의 방법이다[7]. 일반적으로 현실성과 자연스러움을 주기 위해 그 객체나 폴리곤의 맞는 텍스처 맵을 표면에 적용하여 모델링 한다. 텍스처 맵핑 방식은 객체에 텍스처 맵을 입히는 것 외에 별다른 연산이 필요 없으므로 렌더링 속도에도 영향을 끼치지 않는다. 구름을 모델링 할 때도[그림 2와 같이 구름판 모델에 구름 텍스처 맵을 맵핑하여 모델링 하는 구현 예제가 있다[8]. 이러한 연구는 구름 근처나 구름 속에서 비행을 하는 시뮬레이션 소프트웨어의 경우 구름의 현실적 효과가 매우 낮지만 프레임율 측면에서는 높은 효율성을 보이고 있다. 또한 구름판을 이용한 방법의 현실성을 높이기 위한 방안으로 여러장의 텍스처를 쌓아 테스트한 연구가 있었지만, 이 방법을 사용하면 프레임 효율성이 떨어져 장점이 사라지게 된다[9].



그림 2. 구름 텍스처를 입힌 2D 모델
Fig. 2 2D model with cloud texture

2.3. 파티클을 이용한 구름 모델링 방법

일반적으로 시뮬레이션 소프트웨어에서 쓰이는 계층형 구름은 구름판을 특정위치에 렌더링 하는 구름층 구현 방법 중 하나이다. 계층형 구름을 모델링 하는 방법은 앞서 소개한 두 가지 모델링 방안을 모두 적용 가능하다. 본 논문에서는 계층형 구름 모델링 방법을 기

반으로 구름판 모델링 방식을 사용하여 천장과 바닥의 구름층 위치와 카메라 위치에 따라 안개효과의 농도를 다르게 적용하여 현실성을 높이는 방법을 적용하여 구현하였다.

Ⅲ. 계층형 구름 구현 방법

본 장에서는 OSG와 대용량의 지형 데이터베이스 환경에서 계층형 구름을 구현하기 위한 방안을 제안한다. 먼저 제안하는 방안의 참고 예제를 소개하고 실제 구현 방법에 대해 설명한다.

3.1. 기본 사용 모델

본 논문에서 제안하는 계층형 구름의 구현을 위해 프레임율이 높은 기존의 연구 예제인 텍스처 맵핑을 이용한 구름판을 기본 모델 사용하였다[8]. 기본 모델로 사용된 예제인 SigmaOsg는 오픈 소스인 OSG를 기반으로 구름이나 LOD(Level of Detail)와 같은 간단한 프로젝트들을 모아 놓은 애플리케이션이다. 2D의 사각 판 모델에 사용자가 원하는 구름의 농도에 따라 각각 다른 10장의 구름 텍스처를 맵핑하여 구름량 조절은 물론 바람에 의한 구름의 이동을 시각화하는 것이 가능하다. 위의 관련연구에서 보여준 그림 2는 SigmaOsg의 구름 예제를 구현한 결과이다.

3.2. 계층형 구름 설계 및 구현

본 절에서는 앞서 소개한 SigmaOsg의 구름 예제를 기반으로 계층형 구름 효과의 구현 방법을 설명한다. 구현 방법 설명에는 먼저 구름 생성 전에 사용자로부터 입력 받는 구름 상태의 on/off 신호 값과 구름의 고도 값, 구름층의 두께 값을 받아 처리하는 부분에 대해 설명하고 다음으로 구름과 카메라의 위치에 따른 안개효과 설정방법에 대해 설명한다.

본 논문에서는 구름효과 구현을 설명하기에 앞서 구름효과를 내기 위한 변수를 표 1과 같이 정의 하였다. 또한 구현하고자 하는 계층형 구름의 전체적인 구조는 그림3과 같이 설계하였다. 계층형 구름의 효과를 카메라(항공기)의 위치에 맞게 조절하기 위해 카메라의 고도 값과 안개효과의 농도를 저장할 변수를 각각 선언하고, 구름층의 바닥 위치와 천장 위치를 저장할 변수들

을 각각 선언하여 LCloudsDensity()함수의 인자 값으로 넘겨서 구름의 안개효과 농도를 처리하도록 설계하였다. 각 변수들은 각 프레임 마다 아래의 그림 4과 같은 처리 과정을 가진다. 가장 먼저 카메라의 고도 정보에 따른 효과 처리를 해야 되기 때문에 고도 값을 불러온 후 그 고도 값의 위치가 구름판 아래로 진입 층 값만큼 고도 값과 구름판 위로 진입 층 값을 더한 고도 값의 사이일 때 구름의 안개효과를 적용한다. 만약 그 사이의 값이 아닐 경우 구름의 안개효과를 제거 하도록 설계 하였다.

표 1. 구름효과에 사용되는 변수 목록
Table. 1 List of Variables used in Proposed Scheme

변수 명(변수 형)	저장하는 값
wZ(Double)	카메라(항공기)의 고도 값
Density(Float)	안개효과의 농도를 결정하는 값
cloud_BOTTOM_Pos(Int)	계층형 구름의 바닥 구름판 위치 값
cloud_TOP_Pos(Int)	계층형 구름의 천장 구름판 위치 값
cloud_outside(Int)	계층형 구름의 진입 층 두께 설정 값

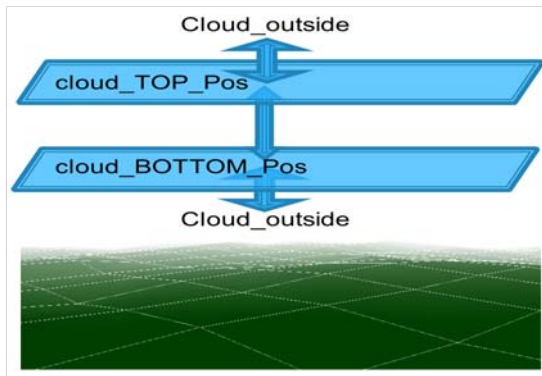


그림 3. 제안하는 계층형 구름형상
Fig. 3 Proposed Layered Cloud

구름층의 구름효과 유효 범위 내에서 고민해야 할 것은 구름속의 안개효과이다. 안개효과는 안개의 농도에 따라 다른 효과를 보이기 때문에 농도 값을 설정하는 것은 매우 중요하다. 본 논문에서는 카메라가 구름층의 구름효과 유효 범위 밖에서 안으로 진입할 때 그 위치에 따라 각각 다른 안개효과의 농도를 적용하기 위해

그림 5와 같이 LCloudsDensity() 함수를 정의하였다. LCloudsDensity() 함수의 기능은 파라미터 변수에 안개 효과의 농도를 저장하는 변수(Density)의 포인터 주소 값과 구름판의 바닥과 천장 위치 값(cloud_TOP_Pos, cloud_BOTTOM_Pos), 진입 층 두께 값(cloud_outside)을 각각 받아 구름판의 천장의 고도 값과 바닥의 고도 값 사이일 때는 농도를 높여 뿌옇게 처리한다. 또한 천장과 바닥 구름판의 위치 값과 진입 층의 두께 값을 사용하여 구름 진입 층의 범위를 계산하고 카메라의 고도가 진입 층 범위인지를 판단한다. 그리고 난 후 진입 층 시작 범위로부터 현재 카메라의 고도를 계산하여 그에 비례 하도록 농도를 증가시키는 방법을 사용하여 보다 자연스러운 구름층 진입 효과를 구현 하였다.

```

if(wZ >= cloud_BOTTOM_Pos - cloud_outside
&& wZ <= cloud_TOP_Pos + cloud_outside )
{ /*카메라 안개효과 적용 부분 소스 추가*/
    LCloudsDensity(&Density, cloud_BOTTOM_Pos,
                    cloud_TOP_Pos, wZ,
                    cloud_outside);
    eFog->setDensity(Density);
}
else
{ /*카메라 안개효과 제거 부분 소스 추가*/
    Density = 0.0f;
}
    
```

그림 4. 계층형 구름 효과 구현
Fig. 4 Implementation of Layered Clouds Effect

```

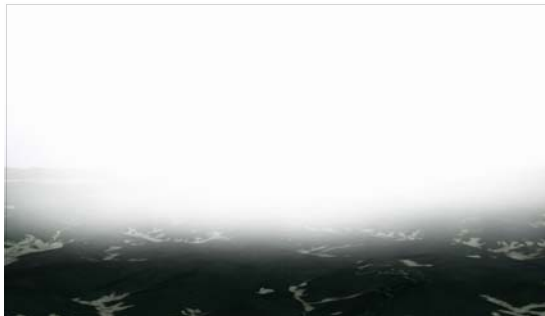
void LCloudsDensity(float *Density, int bottom, int top,
                    double wZ, int cloud_outside)
{
    if(wZ >= bottom && wZ <= top )
        *Density = 0.0005f;
    else if(wZ > (bottom-cloud_outside) && wZ < bottom)
        *Density = (wZ-(bottom-cloud_outside)) *
        0.000001f;
    else if((wZ < (top+cloud_outside)) && wZ > top)
        *Density = ((top+cloud_outside) - wZ) *
        0.000001f;
}
    
```

그림 5. LCloudsDensity 함수 정의
Fig. 5 Definition of LCloudsDensity Function

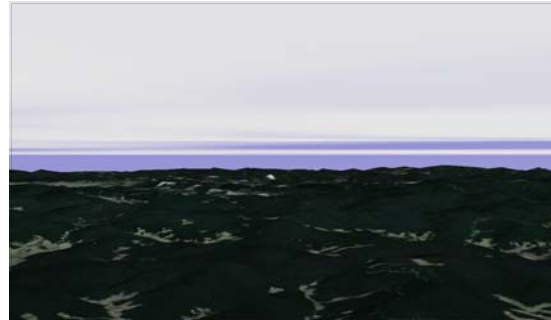
IV. 성능평가

본 장에서는 기존의 파티클을 이용한 구름 모델링 방법과 구름판 텍스처 맵핑을 이용한 모델링 방법을 본 논문에서 제안하는 계층형 구름과 비교하여 프레임율과 현실성을 비교하고자한다. 시뮬레이션하기 위해 i7-3770 3.40GHz, 8GB의 시스템 메모리, NVIDIA GeForce GTX 680을 장착한 시스템에서 각각의 모델링 방식을 OSG를 사용하여 구현하였고, 3.17GB 대용량 지형 환경에서 고정 프레임율 60으로 설정하여 실험하였다. 같은 위치에서 각각의 모델링 구현 방법을 적용하여 실험하였고, 각각의 실험마다 시스템의 성능측정을 위해 프레임율을 측정하고 총 10번의 프레임율을 캡처하여 평균값을 구해 표로 정리하였다. 또한 시뮬레이션에서 현실적 효과를 비교하기 위해 구름의 고도를 똑같이 설정하고 카메라가 구름근처로 진입 시 보여지는 각각의 모델링 방법에 따른 구름 효과 시뮬레이션 결과를 보여준다.

각각의 모델링 방법에 따라 테스트 한 결과 그림 6과 같이 각각의 모델링 방법에 따라 현실성에서 많은 차이를 보였다. 파티클을 이용한 구름 모델링 방법은 구름의 위치가 조정 가능해 원하는 위치에 구름을 위치시킬 수 있으며, 각 구름에 접근하여도 구름 속 안개효과가 자연스럽다. 그에 비해 구름판을 이용한 텍스처 맵핑 방법은 땅에서 보았을 때 그 현실적 효과는 실제 구름과 비슷하지만 구름 근처나 구름 속 비행 시 얇은 구름판으로 인해 부자연스러운 효과를 보게 된다. 하지만 이러한 구름판 모델링 방법에 제안하는 방안을 적용하면 그림 6의 가장 하단의 그림과 같이 보다 자연스러운 구현이 가능하다.



(a)



(b)



(c)

그림 6. 모델링 방법에 따른 시뮬레이션 결과 (a) 파티클 시스템 (b) 구름판 텍스처맵핑 (c) 제안된 방법
Fig. 6 Simulation results according to modeling methods (a) particle system (b) texture mapping cloud board (c) the proposed method

이 같은 현실적 효과와는 반대로 프레임율면에서는 표 2와 같은 차이를 보였다. 파티클을 이용한 모델링 방법은 구름판을 이용한 텍스처 맵핑 방법에 비해 프레임율이 2배정도 떨어져 시뮬레이션에 부하가 생겼다.

그에 비해 제안된 방법은 구름판을 이용한 텍스처 맵핑 방법에 비해 1~2Hz정도의 미묘한 프레임율 저하가 있었지만, 시뮬레이션 하는데 있어 프레임율 저하를 체감하지 못 할 정도였다.

표 2. 모델링 방법에 따른 성능측정 표
Table. 2 Frame Rate According to Modeling Methods

테스트	시뮬레이션시 프레임평균		
	파티클	구름판	제안된 방법
1차	29.36 Hz	58.84 Hz	56.48 Hz
2차	32.82 Hz	57.82 Hz	57.76 Hz
3차	30.45 Hz	59.46 Hz	54.37 Hz

V. 결 론

본 논문에서는 구름효과 구현 방법 중 대용량 고 해상도의 지형 데이터베이스 환경에서도 사용자에게 현실적 효과를 줄 수 있는 layered clouds 모델링 방안을 제안하였다. 기존의 파티클을 이용한 구름 모델링 방법과 구름판 텍스처 맵핑을 이용한 구름 모델링 방법을 테스트하고 제안하는 방안과 비교하였다.

향후 연구과제로서 이번 연구와는 반대로 기존의 파티클을 이용한 구름 모델링 방법을 토대로 프레임율을 높일 수 있는 모델링 방법을 개발 하는 것뿐만 아니라 이번 연구에서는 하지 못했던 실제와 같은 구름의 움직임 모델링할 수 있도록 하고자 한다.

감사의 글

본 논문은 교육부와 한국연구재단의 지역혁신 인력양성사업(No. 2011-02-기-01-010) 및 미래창조과학부 및 정보통신산업진흥원의 고용계약형 SW 석사과정(NIPA-2013-HB301-13-1004)의 연구결과

REFERENCES

- [1] J. Y. Do, N. H. Baek, C. W. Lee, K. W. Ryu, "Implement of Cloud Modeling and Rendering for Flight Simulation," *The KIPS transaction, Part A. Part A v.16A No.5*, pp.307 - 318, 2009.
- [2] WIKIPEDIA Particle system, Available : http://en.wikipedia.org/wiki/Particle_system
- [3] Mark J. Harris, "Real-Time Cloud Simulation and Rendering," The University of North Carolina at Chapel Hill, 2003.
- [4] Cui, H., Qi, M., & Li, D. "3D cloud modeling base on fractal particle method." *Electrical and Control Engineering(ICECE), International Conference on. IEEE*, pp. 5639-5643, 2011.
- [5] Muhammad Azam Rana, Mohd Shahrizal Sunar, Siti Mariyam Shamsuddin, "Particles Cloud Modeling Algorithm for Virtual Environment," *Asian Journal of Information Technology*, 5(5), pp.555-565, 2006.
- [6] Liu, Z., Wang, Z., & Zhang, C, "Scheme of Dynamic Clouds Generation for 3D Real Time Flight Simulation," *ICCMS'10. Second International Conference on. IEEE*, pp. 370-374, 2010.
- [7] WIKIPEDIA. Texture mapping, Available : http://en.wikipedia.org/wiki/Texture_mapping
- [8] sigmaosg, Available : <http://code.google.com/p/sigmaosg/>.
- [9] Tarantilis, Georgios E, "Simulating clouds with procedural texturing techniques using the GPU," NAVAL POSTGRADUATE SCHOOL MONTEREY CA, 2004.
- [10] Upchurch, E. M & Semwal, S. K, "Dynamic cloud simulation using cellular automata and texture splatting," in *Proceedings of the 2010 Summer Computer Simulation Conference. Society for Computer Simulation International*, pp. 270-277, July. 2010.



강석윤(Seok-Yoon Kang)

2012.08: 경상대학교 학사
2012.09 ~ 현재: 경상대학교 정보과학과 석사과정
※관심분야 : 항공 소프트웨어, 비행 시뮬레이션, 임베디드,



김기일(Ki-II Kim)

2002.02: 충남대학교 이학석사
2005.02: 충남대학교 이학박사
2006.03 ~ 현재: 경상대학교 정보과학과 부교수
※관심분야 : 센서네트워크, 이동네트워크, 항공기소프트웨어