가시 순차적 분할 렌더링 알고리즘을 이용한 3차원 텍스쳐† 기반의 볼륨 그래픽 구조

김정우[0] 이원종 박우찬 김형래 한탁돈
연세대학교 컴퓨터 과학과 미디어 시스템 연구실
{mosh[0], airtight, chan, kimhr, hantack}@kurene.yonsei.ac.kr

3D Texture-Based Volume Graphic Architecture using Visibility-Ordered Division Rendering Algorithm

Jung-Woo Kim[0] Won-Jong Lee Woo-Chan Park Hyung-Rae Kim Tack-Don Han
Media System Laboratory, Department of Computer Science, Yonsei University

요 약

3차원 텍스쳐 기반의 볼륨 렌더링 기법은 추가적인 하드웨어가 필요 없기 때문에 개발 비용이 적다는 장점이 있지만 다각형 기반 렌더링에 최적화 된 범용 그래픽 하드웨어를 그대로 사용하기 때문에 성능이 낮다는 단점이 있다. 이에 본 논문에서는 병렬 구조의 고성능 볼륨 렌더링 시스템에서 사용되던 볼륨 정보 분할 기법을 범용 그래픽 하드웨어에 적용하는 새로운 3차원 텍스쳐 기반 볼륨 그래픽 구조를 제안한다. 제안하는 구조를 통해 볼륨 정보를 분할하여 처리하면, 범용 그래픽 하드웨어가 갖고 있던 물리적 메모리 크기의 한계성을 극복할 수 있다. 또한 전체 해상도의 알파 블렌딩이 아닌 분할된 볼륨 정보 하나가 차지하는 크기 만큼의 작은 해상도로 알파 블렌딩을 수행 함으로서 렌더링 단계와 프레임 버퍼간의 데이터 전송량을 1/30로 줄이고 픽셀 캐시의 적중률을 99.9%에 근접하게 높일 수 있다.

1. 서 론

현재 3차원 그래픽스 분야에서는 이전에 목표로 했던 높은 정보 처리량 뿐 아니라 더욱 세밀하고 실감있는 영상을 만들기 위한 노력이 계속 되고 있다. 그러나 아무리 다각형 정보의 개수를 증가시켜 실감영상을 구현한다 해도 물체의 표면 정보만을 갖고 있는 다각형 정보를 통해서는 물체 내부를 표현할 수 없다는 근본적인 문제가 있었다.[1] 때문에 기존의 다각형 정보가 갖고 있는 한계를 극복하고 실감 영상을 구현하기 위해 공간 정보를 통해 영상을 생성하는 볼륨 렌더링 기법에 대한 많은 연구가 진행되기 시작했다. 현재에는 범용 그래픽 하드웨어에 내장된 텍스쳐 매핑 유닛을 통해 볼륨 정보의 처리가 가능해져서 3차원 게임 등의 일반 분야에 까지 그 적용 영역이 확대되고 있는 추세이다. 하지만 범용 그래픽 하드웨어를 사용하여 효과적으로 볼륨 렌더링을 구현하는데에는 아직도 많은 문제점이 있다. 첫째, 물체가 존재하는 공간상의 모든 색깔 정보를 담고 있는 볼륨 정보의 경우 그 양이 매우 크다. 따라서 다각형 정보를 처리하기 위한 기존의 그래픽 하드웨어의 경우 성능 및 메모리의 크기가 볼륨 정보를 처리하기에 부족하다. 둘째, 기존의 그래픽 하드웨어에 효율적인 영상 처리를 위해 내장된 알파 블렌딩 유닛(Alpha Blending Unit), 픽셀 캐시(Pixel Cache) 등의 장치가 다각형 정보를 처리하는데 최적화 되어 있어 볼륨 기반 렌더링시 효율이 급격히 떨어지는 문제가 있다.[2] 하지만 본 논문에서 제안하는 가시 순차적 분할 렌더링 알고리즘을 사용하면 알파 블렌딩 유닛 및 픽셀 캐시의 효율을 최대한 높이고 그래픽 메모리의 크기를 절약하여 전체적인 볼륨 렌더링 성능을 향상 시킬 수 있다.

† 본 연구는 한국과학기술평가원의 '99 국가 지정 연구실 과제 (2000-N-NL-01-C-133)에 의해 수행되었음

2. 3차원 텍스쳐 기반 볼륨 렌더링

3차원 텍스쳐 기반 볼륨 렌더링은 범용 그래픽 하드웨어의 3차원 텍스쳐 매핑 유닛을 사용하여 볼륨 렌더링을 수행하기 위한 기법이다.[3] 이 기법은 별도의 추가된 하드웨어가 필요 없어 개발 비용이 낮고, 다각형과 볼륨 정보가 혼합된 영상을 만들 수 있다는 장점이 있다. 하지만 기존의 그래픽 하드웨어를 그대로 이용하는데서 오는 다음과 같은 몇가지 단점이 있다. 3차원 텍스쳐 기반 볼륨 렌더링을 위해서는 일반적으로 볼륨 정보를 한꺼번에 그래픽 메모리에 저장시켜둔 뒤 생성하려는 영상의 해상도와 동일한 크기의 단면들을 3차원 텍스쳐 매핑을 거쳐 차례로 알파 블렌딩 한다. 따라서 대용량의 물리적 메모리가 요구되는 것은 물론이고 블렌딩 하려는 단면의 해상도가 크기 때문에 보통 4KBytes 내외의 크기를 가지는 픽셀 캐시의 효율은 낮아지게 된다. 볼륨 렌더링의 경우 잦은 알파 블랜딩 때문에 픽셀 캐시의 효율이 매우 중요한데 이것은 프레임 버퍼의 접근 지연 시간이 캐시 또는 내부 유닛의 수행 속도에 비해 비교적 길기 때문이다. 만약 캐시의 적중율이 낮다면 알파 블랜딩을 위해 어쩔 수 없이 프레임 버퍼 접근이 잦아지게 되고 전체적인 렌더링 수행 속도는 느려지게 된다.[4]

2.1 가시 순차적 분할 렌더링 기법

볼륨 정보를 분할하여 렌더링 할 경우 분할된 볼륨 정보를 어떤 순서로 읽어들여 처리할 것인지도 중요한데 이는 볼륨 렌더링의 주요 작업이 작업 대상들의 순서 유지가 필수적인 알파 블렌딩 기

법을 통해서 수행되기 때문이다.[5] 가시 순차적 볼륨 렌더링의 경우 일반적으로 다음과 같은 방법에 따라 분할된 볼륨 정보들의 순서를 유지하며 분할된 볼륨 정보들을 처리하게 된다.

1) 시점에서 본 볼륨 정보의 방향에 따라 Case를 판별한다.
2) 평행 시점(Parallel View) 또는 전경 시점(Perspective View)에 따라 세부 순서를 결정한다.
3) 결정된 순서에 따라 분할된 볼륨 정보를 처리한다.

전체적인 기준이 되는 Case는 크게 세가지로 나눌 수 있다. 평행 시점에서 볼륨 정보의 한 면이 시점과 평행한 경우(Case 1), 볼륨 정보의 한 선(Edge) 또는 점(Point)이 시점을 향하고 있는 경우(Case 2), 전경 시점의 경우(Case 3) 가 그것이다. 세부적으로는 시점의 종류가 평행 시점인 경우 중심이 되는 꼭지점을 중심으로 아래 또는 옆으로 순서가 결정되지만 전경 시점인 경우 시점의 위치를 중심으로 가까운 차례로 순서가 결정된다.[6]

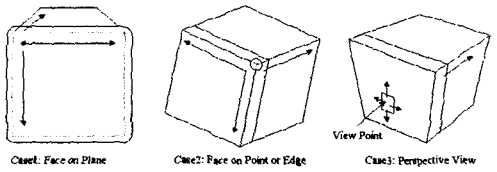

그림 1. 분할 렌더링 순서의 3가지 경우

3. 제안하는 볼륨 렌더링 구조

본 논문에서는 아래 그림과 같이 일반적인 기하 기반 그래픽 하드웨어의 구조에, 볼륨 정보를 분할하여 보이는 순서대로 그래픽 메모리에 넘겨주는 볼륨 분할기(Volume Divider), 볼륨 분할기로부터 분할된 영역에 대한 위치 정보를 받아 해당 영역에 대한 단면을 생성하는 단면 생성기(Slice Generator)가 추가된 새로운 볼륨 렌더링 구조를 제안한다.

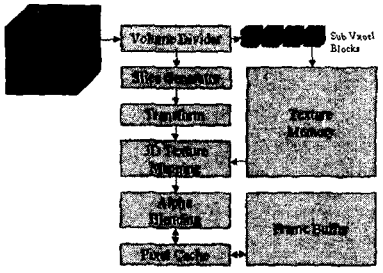

그림 2. 제안하는 구조의 블록 다이아그램

제안하는 구조는 다음과 같이 동작한다.

1) 볼륨 분할기(Volume Divider)는 시점 평면 및 볼륨 정보의 크기와 위치 정보를 통해 분할 볼륨 영역의 크기 및 처리될 순서를 결정한뒤 차례로 분할 영역에 대한 3차원 텍스쳐 맵을 생성한다.
2) 단면 생성기(Slice Generator)는 볼륨 분할기로부터 받은 해당 분할 영역의 경계 정보를 통해 분할 영역에 대한 단면을 생성하여 기하 연산 단계로 보낸다.
3) 기하 연산을 마친 각 단면들은 텍스쳐 매핑 유닛을 통해 미리 생성된 3차원 텍스쳐로 매핑된 후 하나씩 알파 블랜딩 유닛을 통해 블랜딩 된다.
4) 하나의 분할 영역에 대한 단면이 들어오기 시작하면 최초의 단면에 대한 프레임 버퍼 메모리 요청은 모두 캐시 적중이 실패하여 프레임 버퍼로부터 블랜딩 할 위치의 픽셀 정보를 읽어와 캐시에 쓰게 된다.
5) 해당 분할 영역 내에 한하여 최초의 단면을 제외한 나머지 단면들의 픽셀 캐시 적중이 보장되기 때문에 이후의 블랜딩 작업은 프레임 버퍼에의 접근 없이 픽셀 캐시와 알파 블랜딩 유닛 사이에서 일어난다.
6) 분할 영역의 단면에 대한 블랜딩 작업이 종료되면 다음 분할영역의 단면이 블랜딩 유닛에 들어오게 된다. 모든 분할 영역의 첫 번째 단면은 캐시 적중이 실패하기 때문에 이 때 현재 픽셀 캐시에 저장된 정보를 프레임 버퍼에 쓰게 된다.

3.1 볼륨 분할기

볼륨 분할기(Volume Divider)는 볼륨 정보의 분할 및 처리 순서에 대한 전체적인 관리를 하기 위한 추가 장치 이다.

제안하는 구조에서, 분할될 볼륨 영역의 크기는 해당 영역이 시점 평면에 투영되었을 때 차지하는 해상도, 다시 말해 해당 영역의 텍스쳐가 매핑될 단면의 크기가 픽셀 캐시의 크기보다 작도록 결정된다. 이러한 방식을 사용하는 것은 하나의 분할 영역 내의 단면들에 대한 픽셀 캐시 적중률을 높이기 위해서이다. 일반적인 픽셀 캐시의 크기가 4KBytes인 것을 감안할때 32bit의 크기를 가지는 픽셀의 경우 가능한 최대 단면의 해상도는 32×32 가 된다. 볼륨 분할기는 어플리케이션으로부터 얻어낸 시점 정보 및 전체 볼륨의 크기 정보를 통해 분할된 볼륨 정보가 투영될 프레임 버퍼상의 영역이 32×32 픽셀 해상도를 벗어나지 않도록 볼륨 영역의 크기를 결정한다.

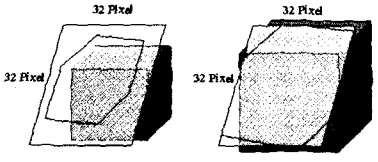

그림 3. 볼륨 정보 분할 크기의 결정

또한 볼륨 분할기는 앞서 설명한 가시 순차적(Visibility - Ordered) 방식을 사용하여 각 분할 영역이 처리될 순서를 결정한다. 이로서 볼륨 정보가 분할되었더라도 볼륨 분할기에서 초기에 결정된 순서에 따라 블렌딩 작업을 수행하면 그림4 와 같이 원본 볼륨 정보를 사용한 경우의 영상과 동일한 결과를 얻을 수 있다.

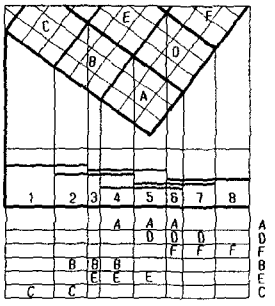

그림 4. 가시 순차적 블랜딩

임의의 분할 영역의 볼륨 정보를 통해 텍스쳐 맵이 생성되면 볼륨 분할기는 해당 영역이 투영될 시점 평면상의 위치 및 크

기를 미리 계산하여 분할 영역의 경계 정보(Boundary Box)를 생성한뒤 단면 생성기에 전달한다. 이 외에도 전체 단면의 개수 및 분할된 볼륨 영역의 개수를 통해 분할 영역당 생성되어야 할 단면의 개수를 결정하여 같이 단면 생성기로 보낸다.

3.2 단면 생성기

단면 생성기(Slice Generator)는 볼륨 분할기로부터 받은 볼륨 경계 정보 및 분할 영역당 단면의 개수 정보를 통해 텍스처를 매핑할 해당 영역에 대한 단면을 생성한다. 단면은 볼륨 경계의 전면에 평행하며 동일한 간격 및 크기로 생성된다. 생성된 단면의 크기는 볼륨 분할기에 의해 32×32 픽셀 이하의 크기로 조절되며 이 단면들은 시점 평면, 즉 생성하고자 하는 영상면에 평행하다. 이것은 픽셀 캐시의 효율을 높이고 구조의 복잡성을 피하기 위해서 이다.

4. 실험 및 결과 분석

실험은 표준 C/C++ 언어를 통해 작성된 시뮬레이터를 통해 이루어졌다. 실험을 위한 볼륨 정보는 256X256X74 개의 복셀 데이터를 256^3 복셀 로 재조정하여 사용하였으며 시점면의 넓이는 256X256, 렌더링에 사용된 단면의 개수는 256개로 하였다.

표 1. 일반 구조와의 캐시 효율 비교

	General	Proposed (16 PBlock)	Proposed (32 PBlock)
Cache Hit	15,728,643	16,744,438	16,760,832
Cache Miss	1,049,372	32,792	16,590
Data Transfer from Frame Buffer	64.3MB	2.2MB	2.1MB

픽셀 캐시는 4Way 방식의 4KBytes 크기를 가지는 것으로 설정하였으며 표에서 알 수 있듯이 캐시 블록의 크기가 커졌을 때 효율이 높아짐을 알 수 있다. 일반 구조와 비교했을 때, 제안하는 구조는 캐시 적중률이 약 3% 상승하여 99.9%에 근접하였으며 렌더링 단계와 프레임 버퍼간의 데이터 전송량이 1/30 로 줄어든 것을 알 수 있다.

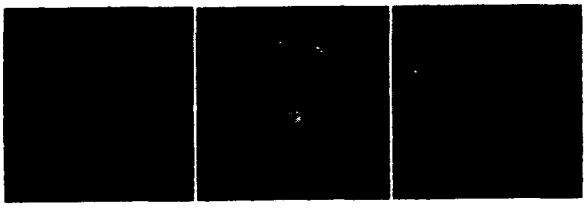

그림 5. 실험 결과 영상

5. 결론

볼륨 분할기로부터 적정 크기로 나뉘어진 분할 영역 내부에서의 알파 블랜딩은 앞서 말한 것처럼 픽셀 캐시의 높은 효율을 보장하게 되어 프레임 버퍼로의 접근을 최소화한다. 결과적으로 제안하는 구조를 사용하면 마치 각 분할 영역당 하나의 보조 이미지가 생성되고 이러한 보조 이미지들이 차례로 프레임 버퍼에 씌여지는 것[7]과 같은 보조 영상 합성 효과(Sub Image Composition Effect)를 낳게 된다.

이러한 경우 긴 지연 시간(Latency)을 가지는 프레임버퍼로의 접근 횟수가 분할된 볼륨 영역의 개수 만큼 줄어들게 되어 프레임 버퍼 접근으로 인한 성능 저하를 막을 수 있다. 또한 렌더링 종단에서 요구되는 버스 대역폭이 전체 볼륨 영역에 대한 모든 단면들의 정보량이 아닌 각 분할 영역에서 생성되는 보조 영상 정보량에 의해 결정되므로, 렌더링시 요구되는 대역폭의 크기를 줄일 수 있다. 결국 모든 렌더링 유닛의 성능을 높이지 않고도, 캐시의 성능만을 높임으로서 전체적인 렌더링 성능속도의 큰 향상을 기대할 수 있어 적은 비용으로 높은 효과를 낼 수 있는 장점이 있다.

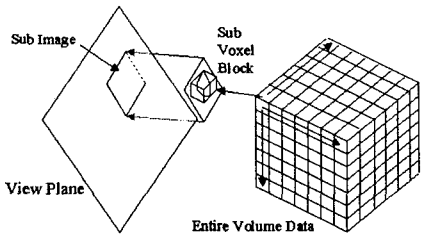

그림 6. 보조 영상 합성 효과

참고 문헌

[1] Barthold Lichtenbelt, Randy Crane, Shaz Naqvi, "Introduction to Volume Rendering," Hewlett-Packard Professional Books/A Prentice Hall Title, p5-6, 1998.

[2] Frank Dachille, Kevin Kreeger, Baoquan Chen, Ingmar Bitter, Arie Kaufman, "High-Quality Volume Rendering Using Texture Mapping Hardware," SIGGRAPH Eurographics Graphics Hardware Workshop, 1998.

[3] T. Todd Elvins "A Survey of Algorithms for Volume Visualization," Technical Report, San Diego Supercomputer Center, 1992.

[4] Orion Wilson, Allen Van Gelder, and Jane Wilhelms, "Direct volume rendering via 3D textures," Technical Report of University of California, Santa Cruz, UCSC-CRL-94-19, 1994.

[5] Kwan-Liu Ma, James S. Painter, Charles D. Hansen, Michael F. Krogh, "A data distributed parallel algorithm for ray-traced volume rendering," Proceedings of Parallel Rendering Symposium, p15-22, 1993.

[6] H. Ray and D. Silver, "The RACE II engine for real-time volume rendering," Proceeding of SIGGRAPH/Eurographics Workshop, p129-136, 2000.

[7] Gunter Knittel, "TriangleCaster: extensions to 3D-texturing units for accelerated Volume Rendering," Proceedings of the 1999 Eurographics/SIGGRAPH workshop on Graphics hardware, p25-34, 1999.