

그래픽스 API를 이용한 복셀화

정현배⁰ 송주환 최성희^{*} 권오봉
전북대학교 컴퓨터과학과
전주 공업대학 컴퓨터공학과

hbjeong@cs.chonbuk.ac.kr, jwsong@cs.chonbuk.ac.kr, obgwun@moak.chonbuk.ac.kr, shchoi@jtc.ac.kr

Voxelization based on graphics API(Application Program Interface)

Hyeon-Bae JeongYoung⁰ Ju-Whan Song Seong-Hui Choi^{*} Ou-Bong Gwon
Dept. of Computer Science, Chonbuk National Univ.
Dept. of Computer Engineering, Chonju Engineering Collage

요 약

본 논문은 3차원 그래픽스 시스템이 기본적으로 제공하는 프리미티브인 면(surface), 솔리드(solid)의 복셀화 및 이들을 결합한 CSG 표현의 복셀화에 대하여 고찰한다. 복셀화는 서피스로 표현된 물체의 빈 내부를 복셀로 표현하여 내부를 채우는 과정이다. 본 방법의 특징은 PC 플랫폼이 제공하는 그래픽스 API인 OpenGL을 이용하여 면, 솔리드 등을 고속으로 복셀화하여 이들을 기본 물체로 하여 폴리언연산에 의해서 간단히 새로운 물체를 만들고 볼륨 환경에서 이들을 단일화된 방법으로 렌더링할 수 있는 것이다.

1. 서 론

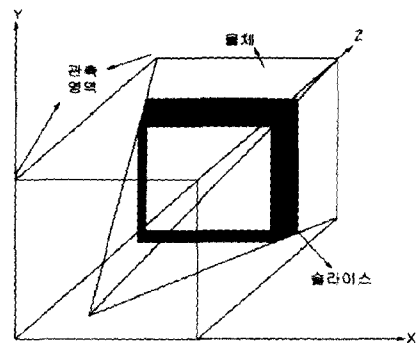
일반적으로 볼륨 가시화는 CT나 MRI 같은 단층의료영상촬영 기구나 유한요소법 등의 수치 시뮬레이션을 이용하여 획득한 대량의 데이터에서 필요한 정보를 추출하여 가시화하는 것이다 [1]. 볼륨 가시화에서는 프리미티브로서 복셀을 사용하기 때문에 가시화하기 위해서는 물체가 복셀로 구성되어 있다는 전제 조건이 필요하다. 그러나 대부분의 인간이 만든 인공물 즉 가위, 핀셋 등은 서피스로 모델링 된다. 따라서 가위 등의 인공물과 촬영기구로 획득한 심장 등을 함께 표현하기 위해서는 서피스로 모델링 된 물체를 복셀화할 필요가 있다[2].

기존의 복셀화에 관한 연구는 전처리 단계에서 이루어진다는 특징 때문에 고속처리에 관하여 그다지 힘을 쏟지 않아 매우 느렸다[3]. 그래서 대략적으로 서피스로 구성된 물체를 복셀화 하여 원래 복셀로 구성된 물체와 합하여 렌더링 하는 것은 쉽지 않았다. 본 논문에서는 OpenGL[4]을 이용하여 서피스 및 솔리드로 구성된 물체를 고속으로 복셀화하는 방법에 관하여 소개한다. 본 방법은 고속으로 복셀화가 가능하기 때문에 서피스 및 솔리드는 데이터를 복셀로 변환하여 저장할 필요가 없고 본래의 형식으로 저장하여 볼륨렌더링이 필요한 경우에만 변환 하여 이용할 수 있고 렌더링 된 결과를 복셀 데이터로 저장하기 때문에 텍스처 맵핑을 이용하면 렌더링 결과를 그대로 3차원으로 복원할 수 있다는 장점이 있다.

본 논문은 2 장에서 OpenGL을 이용한 복셀화의 개념, 서피스로 구성된 물체의 복셀화, 솔리드로 표현된 물체의 복셀화, CSG로 표현된 물체의 복셀화에 대하여 서술하고, 3 장에서 구현 결과를 보이고 고찰을 하고, 4 장에서 결론을 내린다.

2. 그래픽스 API를 이용한 복셀화

본 논문에서는 임의의 물체를 슬라이스로 나누어 3차원으로 화면에 차례로 렌더링하고 이것을 모두 메모리에 저장하는 방식으로 복셀화를 한다. <그림 1> 과 같이 복셀화된 물체의 모양을 유지하기 위하여 직교투상법(orthographic projection)을 이용한다. 물체의 주 방향을 설정하여 이 방향을 관측 영역의 Z축 방향에 맞추어 관측 공간의 깊이를 조절하여 물체를 슬라이스로 나눈다.



<그림 1> 관측 영역과 슬라이스 개념도

여기서 복셀의 대상이 되는 물체는 곡면이나 이차원 평면으로 이루어진 서피스, 구, 육면체 폴리곤들로 이루어진 부피를 갖는 솔리드, 솔리드를 프리미티브로 하는 CSG 물체이다. 이들의 구체적인 복셀화 방법은 다음과 같다.

2.1 서피스로 구성된 물체

서피스 복셀화는 OpenGL의 서피스 프리미티브로 이루어진 물체들에 대한 볼륨 표현을 만들어 내는 것이다. 서피스 프리미티브는 삼각형, 사각형을 포함한 여러 가지 폴리곤과 곡면, NURBS 등이 있는데 이러한 물체들의 내부는 정의되지 않는다. 복셀화는 서피스 그래픽스가 2차원 스캔-컨버전을 이용하여 서피스를 화면에 그려준다는 점을 활용한다. 물체가 들어있는 관측 영역(view volume)전체를 Z축 방향에 대하여 일정한 간격으로 나누어 화면에 그린다. 나누어진 부분은 일정한 간격으로 Z축 두께를 갖고 X와 Y축 공간에 대해서도 폴리곤 형태의 조각이 전부 존재한다. 물체를 위와 같이 나누어 놓은 것을 슬라이스라고 하는데 면으로만 되어 있는 물체를 슬라이스로 나누면 선으로 표현된다. 서피스는 내부 정보를 가지고 있지 않기 때문에 각각의 면 슬라이스들에 대한 정보를 전부 메모리에 저장하고 이들을 조합하면 복셀화와 이의 복원은 간단히 이루어진다.

이것은 실제로 어떤 서피스로 이루어진 물체에 대하여 일정한 간격으로 앞쪽에서 뒤쪽으로 이동하면서 생성되는 슬라이스의 단면들을 화면에 출력하고 그 출력된 면을 메모리에 저장함으로써 이루어진다. 문제점은 관측 영역 안에 표현되는 서피스 물체와 관측 영역의 Z축과 이루는 각도가 작은 경우에는 단면으로 구성된 복셀은 그 물체를 연결된 물체의 형태로 나타내지 못하는 것이다. 이런 경우 물체 자체의 폭이 충분히 두꺼워야 연결된 형태로 표현된다. 이 문제는 제한된 해상도로 3차원 물체를 표현하기 때문에 발생하는데 OpenGL의 엔티앨리어싱 기능을 활성화 시킴으로써 완화시킬 수 있다.

2.2 솔리드로 표현된 물체

솔리드는 폴리곤, 곡면, NURBS 등으로 외부가 완전히 둘러싸여 내부를 가지고 있는 물체이다. 솔리드 복셀화 알고리즘에 대한 필요 조건은 서피스 물체들이 경계면을 완전히 둘러싸야 하는 점이다. 이것은 하나의 솔리드 뿐 아니라 겹쳐지지 않은 모든 솔리드에 대해서도 적용된다. 솔리드 물체는 내부를 가지고 있기 때문에 복셀화도 내부를 채울 필요가 있다. 그러나 일반적인 솔리드 물체를 2.1 절에서 설명한 방법으로 복셀화를 하면 내부가 텅 비게 된다. 따라서 내부를 채우는 방법을 고안해야한다. 솔리드로 표현된 물체의 복셀화는 2.1 절의 방법을 기본으로 하여 물체를 채우는 방법을 첨가한다.

솔리드로 표현된 물체의 복셀화는 서피스로 구성된 물체의 복셀화에서처럼 관측 영역을 Z축 방향으로 일정한 간격으로 앞에서 뒤로 이동하면서 슬라이스를 만들고 이 슬라이스를 렌더링하여 메모리에 저장한다. 그러나 이 슬라이스는 경계면(폐곡선)만을 가지고 있어 내부를 채워야 된다. 본 논문에서는 OpenGL의 블렌딩 함수와 비트 단위의 XOR 연산을 이용하여 내부를 채운다. 블렌딩 함수는 두 개의 픽셀 면(pixel plain)의 색을 혼합하는 함수이다. 솔리드의 제일 앞면은 항상 내부가 차

있고 모든 앞면의 크기는 뒷면보다 작다고 가정한다. 그 다음 면에서부터는 내부가 비어 있는데 이것을 앞면과 뒷면을 XOR 연산과 블렌딩 함수를 이용하여 뒷면을 채워나간다. 설명의 편의상 빈 공간의 픽셀 값을 0이라고 하고 물체의 경계 면을 1 이라고 하면 XOR 연산에 의해서 빈 공간의 값이 1이 되어 채워진다.

솔리드가 주어진 관측 영역 내에 완전히 들어가는 경우에는 위의 방법으로 복셀화가 가능하나 물체가 들어가지 않는 경우에는 물체의 중간에서부터 렌더링을 해야하기 때문에 위의 방법을 적용할 수 없다. 물체가 관측 공간 내에 완전히 포함되는 경우에는 처음 만들어지는 슬라이스가 물체의 한쪽 끝에서부터 만들어지기 때문에 내부가 완전히 채워지기 때문이다. 물체가 주어진 관측 영역 내에 완전히 들어가지 않는 경우에는 물체를 Z축 방향의 맞추어 가상의 채워진 슬라이스를 하나 만들어 이 곳으로부터 XOR 연산을 수행하면 물체가 관측 영역에 완전히 들어가 있는 것처럼 복셀화를 수행할 수 있다.

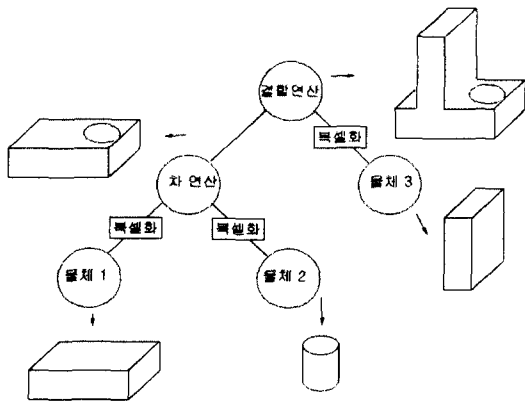
솔리드로 표현된 물체도 해상도의 제한 때문에 경계면에서 손실이 발생하는데 이것은 엔티앨리어싱을 이용하여 해결한다. 그리고 XOR 연산을 할 때 두 면이 살짝 겹치는 경계에서 손실이 발생하는데 이것은 복셀화가 이루어지지 않은 슬라이스와 복셀화한 슬라이스를 OR 연산하여 해결한다.

2.3 CSG로 표현된 물체의 복셀화

CSG 표현은 간단한 프리미티브를 이용하여 불리언(boolean) 연산과 어핀(affine) 변환을 이용하여 복잡한 3차원 솔리드 물체를 만들 수 있다. 이 방식은 조작자가 원하는 물체를 스크린 상에서 대화적으로 만들 수 있기 때문에 산업 디자인 분야에서 호평을 받고 있다. 솔리드를 볼륨으로 변환하면 볼륨으로 변환된 솔리드를 임의로 불리언 연산으로 조합하여 복잡한 물체를 만들 수 있다. 모든 CSG 모델들에 대하여 표현의 단일성을 유지하기 위하여 CSG 표현으로 이루어진 물체들의 단말 노드(leaf node)를 복셀화하여 볼륨 표현으로 바꿀 수 있다. 볼륨으로 된 CSG 트리는 복셀에 불리언 연산을 쉽게 적용할 수 있는 특징이 있어 아주 다양한 물체를 손쉽게 만들 수 있는 장점이 있다.

<그림 2>에서와 같이 복셀화된 볼륨으로 된 기본 물체들을 결합(union), 차(subtract)와 같은 불리언 연산을 이용하여 복잡한 물체를 구성할 수 있다. 또 이 방법은 기하학적으로 정의된 물체와도 쉽게 결합시킬 수 있다. 볼륨을 합성할 때 사용하는 연산 중 일부는 다음과 같다. 단 여기서 C1은 물체 1의 색 C2는 물체 2의 색 C는 합성 후의 물체의 색이라고 가정한다.

- 결합(Union) 연산
 $C = \text{Max}(C1, C2)$: Max는 C1, C2 중 큰 것을 의미한다.
- 교차(Intersection) 연산
 $C = \text{Min}(C1, C2)$: Min는 C1, C2 중 작은 것을 의미한다.



<그림 2> 볼륨 CSG 모델의 트리구조도

합(Add) 연산

$C = \text{Add}(C1 * F1) + (C2 * F2)$: C1과 C2의 색이 F1:F2 비율로 C에 혼합된다.

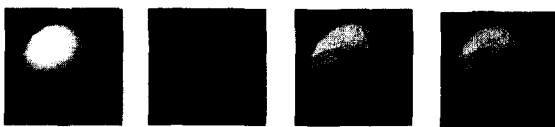
차(Subtract) 연산

$C = \text{Max}(C1 - C2, 0)$: C2가 0이면 C1의 색이 C에 할당되고 C2가 0이 아니면 C2에 최대값을 할당하여 C1-C2가 음수가 되게 한다.

3. 구현 및 결과

2장에서 기술한 방법으로 간단한 복셀라이저를 IBM-PC pentium III 550MHz, RAM 256MB의 플랫폼에서 VC++와 OpenGL을 이용하여 구현하였다.

이것은 CSG 볼륨화를 위한 예비 실험으로 구 볼륨의 생성 결과를 <그림 3>에 표시한다. 복셀화된 볼륨의 해상도는 $256 \times 256 \times 256$ 이다. 서피스로된 구 한 개를 선택하여 OpenGL로 렌더링하여 화면에 표시하여 XOR연산을 하여 메모리에 저장하는 방식으로 복셀화를 하였다. <그림 3-a>는 다각형으로 구성된 구의 렌더링 결과를 표시하고 <그림 3-b>는 구에서 한 개의 슬라이스를 렌더링한 결과를 표시한다. <그림 3-c>는 복셀화된 구 일부를 레이캐스팅을 이용하여 렌더링한 결과를 표시하고 <그림 3-d>는 복셀화된 구 전체를 레이캐스팅을 이용하여 렌더링한 결과를 표시한다. <그림 3-c>에서는 내부가 채워진 것을 육안으로 확인할 수 있다.



(a) 구 (b) 슬라이스 (c) 복셀단면 (d) 전체복셀

<그림 3> 서피스로 이루어진 구 및 결과 볼륨

본 논문에서의 방법으로 구를 슬라이스로 분할하고 XOR 연산을 하여 복셀화하는데 소요시간은 13.35초 걸렸는데 종래의 알고리즘보다는 개선된 것이라 생각된다. 물체가 복잡해지면 더욱더 시간차가 많이 나리라 기대된다.

구현할 때 PC에서 OpenGL의 해상도가 한계가 있어 슬라이스 간격을 어느 정도 줄이면 단면이 렌더링 되지 않는 경우가 발생하여 간격을 임계 간격 이상으로 하였다.

4. 결론 및 향후 연구

본 논문에서는 그래픽스 API인 OpenGL이 제공하는 기능을 이용하여 서피스 및 솔리드를 대화적으로 복셀화하는 방법에 대하여 기술하였다. 본 방법은 서피스와 솔리드로 된 물체를 슬라이스로 나누어 렌더링하여 이를 XOR 연산과 블렌딩 함수를 이용하여 결합하여 복셀화 한다. 본 방법은 고속으로 복셀화가 가능하기 때문에 서피스 및 솔리드로 된 물체를 대화적으로 복셀 데이터와 함께 사용할 수 있고 렌더링 된 결과를 그대로 복셀 데이터로 저장할 수 있기 때문에 텍스처 매핑을 이용하면 렌더링 결과를 그대로 3차원으로 복원할 수 있는 것이 장점이다. 단점은 OpenGL의 해상도의 한계 때문에 슬라이스 간격을 너무 좁히면 단면이 나타나지 않는 경우가 발생하여 이에 대한 처리 방법을 고안할 필요가 있다.

향후 연구로는 복셀화된 솔리드를 볼리언 연산을 이용하여 결합시켜 보다 복잡한 물체를 만들어내는 시스템의 구현을 계획하고 있다.

<참고 문헌>

[1] Richard S. Gallagher, "Computer visualization," CRC, 1994.
 [2] J. Huang, R. Yagel, V. Filippov, and Y. Kurzion, "An Accurate to Voxelize Polygonal Meshes." *IEEE, ACM SIGGRAPH*, 119-126. 1998.
 [3] M. Sramek, and A. E. Kaufman, "Alias-Free Voxelization of Geometric Objects," *IEEE, Transaction on Visualization and Computer Graphics*, 5(3): 251-267, 1999.
 [4] J. Neider, T. Davis, and M. Woo, "OpenGL Programming Guide," Addison Wesley, 1993.



폴리곤으로 구성한 체스 말을 본 논문의 방법으로 복셀화한 후 레이 캐스팅으로 렌더링 한 결과