대한산업공학회/한국경영과학회 2002 춘계공동학술대회
한국과학기술원(KAIST) 2002년 5월 3일-4일

Stencil Buffer를 이용한 형상의 복셀화
Hardware accelerated Voxelization using a Stencil Buffer

장동고, 김광수
경상북도 포항시 남구 효자동 산31번지 포항공과대학교 산업공학과

Abstract

We propose a hardware accelerated voxelization method for various 3D object models such as surface models, solid models, and volumetric CSG models. The algorithm utilizes the stencil buffer that is one of modern OpenGL graphics hardware features. The stencil buffer is originally used to restrict drawing to certain portions of the screen. The volumetric representations of given 3D objects are constructed slice-by-slice. For each slice, the algorithm restricts the drawing areas with the inner region of 3D objects using the stencil buffer, and generates slices of the volumetric representation for target objects. As a result, we can provide volume graphics support for various engineering applications such as multi-axis machining simulation, collision detection and finite element analysis.

1. 서론

지금까지 물체 외형의 형상화에 많은 연구가 수행되었고 이의 결과로서 다양한 분야에서 기술들이 개발되었다. 최근에는 사물의 표면뿐만 아니라 내부까지도 형상화의 대상으로 하려는 시도들이 있다. 이러한 시도의 하나인 볼륨 모델링은 현실의 공간을 아주 작은 단위공간인 복셀(voxel)의 집합으로 가정하고 관심의 대상이 되는 공간의 특징을 이 단위공간이 가지게 하여 전체 공간의 성질을 나타내려는 것이다. 초기의 볼륨 모델링(volume modeling)은 대량의 데이터를 저장하고 처리하는데 어려움을 겪었으나 최근 컴퓨터 하드웨어의 발달로 인하여 대용량의 정보를 빠른 시간 내에 처리하는 것이 가능해졌다.

공간상의 물체를 복셀로 표현할 때 가장 기본적인 방법으로 물체의 내부와 경계부분에 해당하는 복셀은 1의 값을 가지게 하고 외부에 해당하는 복셀에 0의 값을 가지게 하는 이진데이터로 표현하는 것이다. 이러한 형태의 볼륨 데이터(volumetric data)를 얻기 위하여 선형 및 이차곡면, 매개변수 곡면 및 곡선, 다각형등의 다양한 종류의 형상에 대한 복셀화(voxelization) 알고리즘[1-6]이 개발되었다.

그러나 이러한 알고리즘은 주로 이론적인 특징, 즉 정확성(accuracy), 분리성(separability or tunnel-free), 그리고 최소성(minimality)을 만족하기 위해 개발되었다. 따라서 공학적인 응용이 필요한 수행속도에 대한 고려는 간과되어 왔다. 이는 복셀화의 과정이 기하학적 물체에 대하여 렌더링(rendering)의 목적으로 사용되어 실제 실질적인 작업의 수행이 아닌 부수적인 과정으로

인식되었기 때문에 사전작업(pre-processing)으로 별개로 처리되어 왔다. 이러한 이유로 형상이 동적인 변화하는 환경에는 직접적인 응용이 불가능하였다.

또한 솔리드 모델(solid model)의 경우 물체의 경계곡면은 다각형등의 기본도형으로 표현되어있어 기존을 알고리즘으로 복셀화가 가능하였지만 실제로는 물체의 내부역시 복셀로 변환되어야 한다. 모든 복셀에 대하여 물체의 내부와 외부를 판단해야 하기 때문에 곡면의 복셀화 알고리즘이 직접 적용되기는 어렵다. 이를 해결하기 위해 점 분류 알고리즘(point classification algorithm)[7]이 개발되었으나 너무 느린 수행속도로 인해 직접적인 사용이 어렵다. 솔리드 모델의 또 다른 표현 방법인 CSG 모델을 대상으로는 점 추출 알고리즘(point sampling algorithm)[8], 볼륨 추출 알고리즘(volume sampling algorithm)[9], 거리장 알고리즘(distance volume algorithm)[10] 등이 개발되었다. 그러나 이들 알고리즘은 전체 물체를 대상으로 복셀화가 수행되기 때문에 계산시간이 많이 필요하게 된다.

이러한 문제점을 해결하기 위해 최근에 들어서야 그래픽스 하드웨어(graphics hardware)를 이용한 빠른 수행속도를 동반한 복셀화 알고리즘[11]이 개발되었다. 이 알고리즘은 그리픽스 시스템의 가시영역(viewing volume)을 조절하여 물체를 렌더링하여 화면상의 영상을 획득하여 단면 단위로 복셀화를 진행한다. 곡면 모델의 복셀화 과정에서는 anti-aliasing이 추가적으로 적용되고 솔리드 모델의 복셀화 과정에서 기존의 화면에 새로운 화면을 XOR 도구를 이용하여 덮어쓰는 방법으로 단면의 복셀 표현을 만들어 낸다.

본 논문은 이러한 그래픽스 시스템을 이용한 하드웨어 가속 복셀화 알고리즘의 일종으로 현재 대중화되어 있는 OpenGL 시스템의 stencil buffer를 이용해 관심의 대상이 되는 영역에 대하여 빠르게 복셀화를 수행하는 알고리즘을 제시한다.

2. 본론

2.1. Stencil Buffer

OpenGL의 buffer는 color buffer, depth buffer, stencil buffer, accumulation buffer로 이루어진다. 이중 stencil buffer는 화면의 일정한 영역에 대하여 그림이 그려지도록 제한하는 목적으로 이용된다. 예를 들어 그림 1과 같이 창 너머로 보이는 배경을 나타내기 위해서는 우선 비행기의 내부를 그린 후 유리에 해당하는 부분을 stencil buffer에 저장한다. 그 후 배경을 그리면 유리를 통해 보이는 배경만 화면에 표현된다.

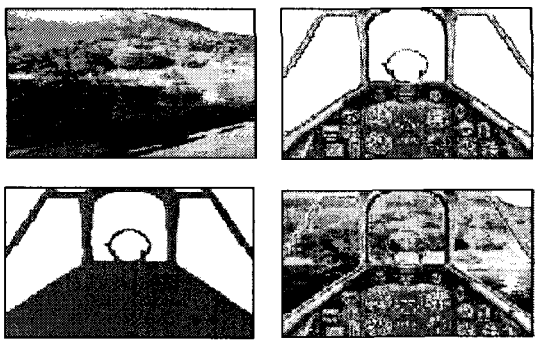

그림 1. Examples of stencil buffer usage

이러한 stencil buffer를 이용하여 쉽게 구현할 수 있는 기능이 CSG(Constructive Solid Geometry)의 렌더링이다. CSG의 기본 형상들의 Boolean 작업을 통해 생성되는 CSG 모델을 화면에 나타내기 위해서는 실제로 모델에 Boolean 작업을 수행하여야 하지만 OpenGL의 stencil buffer를 이용하여 CSG의 최종형상을 렌더링 할 수 있다. 그림 2는 두개의 기본도형에 대해 병합 (union), 교차(intersection), 공제(subtraction)등의 작업

을 수행한 후의 형상을 stencil buffer를 이용하여 렌더링한 것이다.

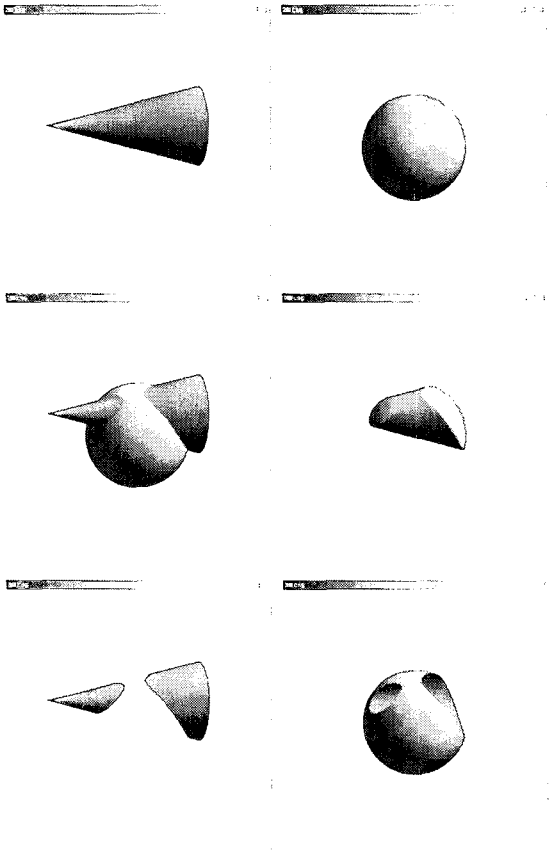

그림 2. CSG Rendering

2.2. 입체형상의 복셀화

본 논문에서 대상으로 하는 입체형상은 경계의 표현에 다면체, 이차곡면, 그리고 NURBS 곡면 등의 다양한 방법이 이용될 수 있다. 대상으로 하는 물체가 2개 이상인 경우 각각의 물체는 서로 교차하지 않아야 한다.

본 논문에서 제시하는 복셀화 알고리즘은 공간상의 물체에 대하여 관심의 대상이 되는 영역(target volume)에 국한하여 내부를 포함하는 복셀표현을 얻어낸다. 관심의 대상이 되는 영역은 공간상의 직육면체 형태로 표현되고, 이는 일반적인 렌더링 시스템의 가시영역(viewing volume)을 이용하여 지정할 수 있다. 이 경우 직육면체의 옆면은 시각좌표계의 z축과 평행하도록 설정된다. 가시영역을 지정할 때 원근투영(perspective view)이 설정되어 있는 경우는 직교투영(orthogonal view)으로 변환하여 지정한다. 가시영역을 설정할 때 단면의 크기는 원하는 관심영역에 맞게 설정하고 깊이는 물체를 모두 포함할 수 있는 정도로 충분히 크게 선택한다. 깊이방향으로의 관심영역은 단면단위로 진행되는 복셀화 과정에서 반영된다. 일반적으로 관심영역은 화면전체에 렌더링되기 때문에 화면의 크기는 단면의 해상도를 결정한다. 또한 원하는 해상도 만큼의 단면으로 깊이 방향의 관심영역을 나누어 복셀화를 진행하는 방법으로 깊이방향의 해상도를 결정한다.

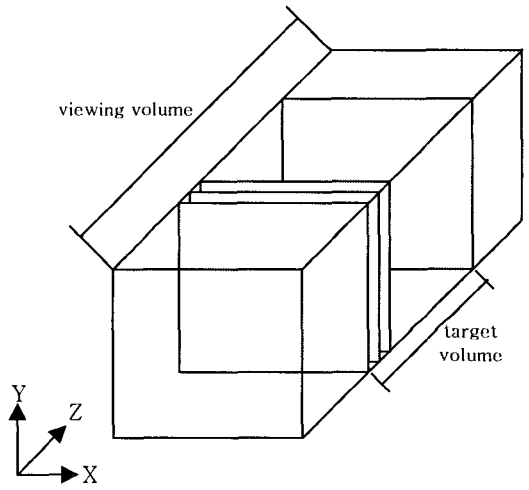

그림 3. Viewing volume and target volume

제시하는 알고리즘은 OpenGL시스템의 frame buffer중 color buffer는 이용하지 않고 stencil buffer와 depth buffer만을 이용하기 때문에 현재 화면상에 있는 영상에 영향을 주지 않고 복셀화를 진행 할 수 있다. 이를 위해 실제 단면단위의 복셀화 알고리즘이 진행되기 이전에 color buffer에 쓰기를 금지하고 시작한다. 각 단면을 대상으로 stencil buffer를 이용하여 복셀 영상을 얻어내기 위해서는 stencil buffer와 depth buffer를 초기화 한다. 이때 stencil buffer는 대상이 되는 단면의 깊

이 값으로 초기화를 진행한다. 이 후에 복셀로 변환하고자 하는 대상을 그린다. 이 과정에서 대상 단면의 depth 값보다 작거나 같은 값을 가진 pixel만이 depth test를 통과하게 되고 이 pixel에 대해서만 stencil buffer의 값을 증가시킨다. 이 때 color buffer와 depth buffer에 쓰기 금지를 설정 해 놓았기 때문에 화면에는 나타나지 않는다. 여기까지가 진행되면 stencil buffer는 홀수번 증가된 pixel과 짝수번 증가된 pixel로 나눌 수 있다. 그림 4는 구의 복셀화 결과를 XZ 평면에서 바라본 것이다. A에 해당하는 부분은 홀수번 증가된 부분이고 B에 대항하는 부분은 짝수번 증가된 부분이다. 홀수번 증가된 pixel의 집합이 바로 대상 단면에서의 물체의 내부가 된다.

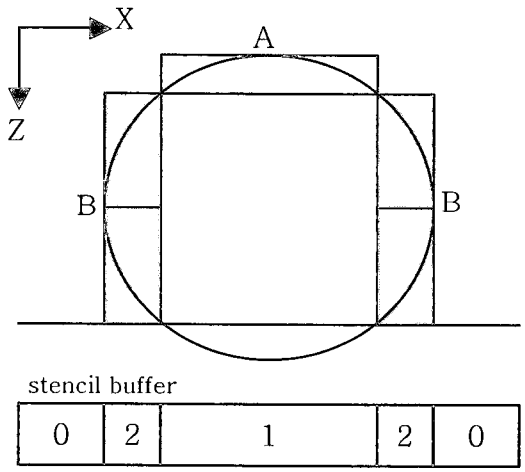

그림 4. stencil buffer

따라서 단면을 평면으로 다시 그리면서 홀수번 증가된 pixel의 값을 1로 짝수번 증가된 값을 0으로 바꾸어 주면 stencil buffer는 주어진 단면에 대하여 물체의 내부는 1로 외부는 0으로 채워지게 된다. 하나의 단면에 대해 이상의 작업이 끝나면 주어진 단면의 복셀화는 완료되고 이때의 stencil buffer index의 값을 메모리로 전송한 후 다음 단면을 진행한다. 그림 5는 이상의 과정을 나타낸 것이다.

```
glDepthFunc(GL_LEQUAL);
disable writing to color buffer
for each slice
  set clear depth to target slice
  clear depth buffer and stencil buffer

  disable writing to depth buffer
  enable depth test and stencil test

  glStencilFunc(GL_ALWAYS, 0, 0);
  glStencilOp(GL_KEEP, GL_KEEP,GL_INCR);
  draw object

  glStencilFunc(GL_EQUAL, 1, 1);
  glStencilOp(GL_ZERO, GL_REPLACE,
              GL_REPLACE);
  draw slice plane

  dump stencil index to main memory

  enable writing to depth buffer
  disable depth test and stencil test
end for
enable writing to color buffer
```

그림 5. algorithm for voxelization

2.3. 결과

이상의 알고리즘은 512M 메모리의 Pentium-III 800MHz PC에서 ATI Radeon 32M DDR을 그래픽 하드웨어로 사용하여 개발되었다. 그림 6과 그림 7은 구(sphere)를 대상으로 복셀화 알고리즘을 적용한 것으로 128x128x128의 해상도로 구를 복셀화 하는데 소요된 시간은 1.9초이다. 그림 8와 그림 9는 OpenGL에서 제공하는 teapot에 대해 복셀화 알고리즘을 적용한 것이다. 계산시간은 128x128x128의 해상도에서 2.4초가 소요되었다. 알고리즘에서 가장 많은 시간을 소모하는 부분은 stencil buffer에서 메모리로 전송하는 부분이다. 메모리로의 전송 시간을 제외하고 실제 복셀화에 소요된 시간은 각각 0.15초, 0.6초로 실시간에 변환이 가능하다. 이는 3D texture memory를 지원하는 고급사양의 그래픽 하드웨어를 이용한다면 해결할 수 있다.

3. 결론

본 논문은 다양한 형상모델을 대상으로 원하는 부분에 대하여 하드웨어를 이용한 볼셀화 알고리즘을 제시한다. 본 알고리즘은 현재 대중화 되어 있는 OpenGL 그래픽 가속장치의 기능 중 stencil buffer를 이용하여 형상을 단면 단위로 복셀화를 진행시켜 나간다. 복셀화 과정에서 물체의 표현법에 관계 없이 단일화된 방법을 제시하고, 관심의 대상이 되는 영역에 국한하고 하드웨어를 이용하여 소요되는 계산시간을 단축한다. 이러한 계산시간이 짧은 복셀화 알고리즘은 다축 가공 시뮬레이션, 간섭파악, 유한 요소 해석등의 다양한 공학적 응용분야에 적용될 수 있다.

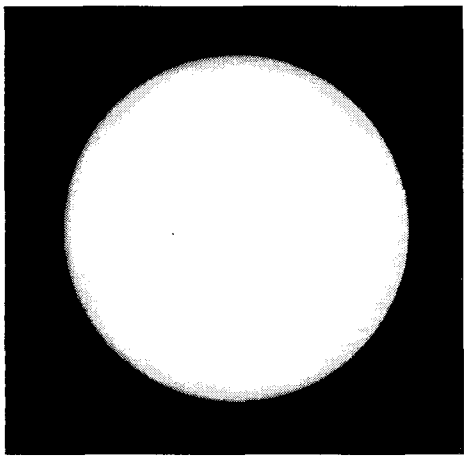

그림 6. Sphere

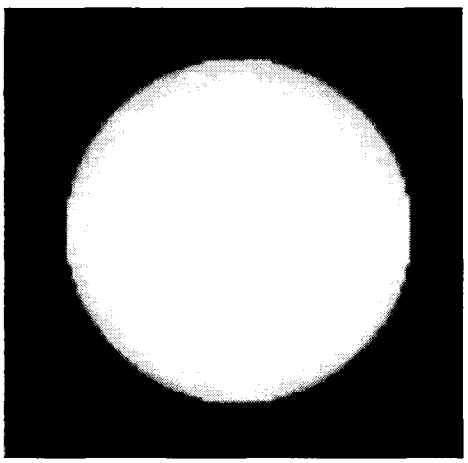

그림 7. Voxel Sphere

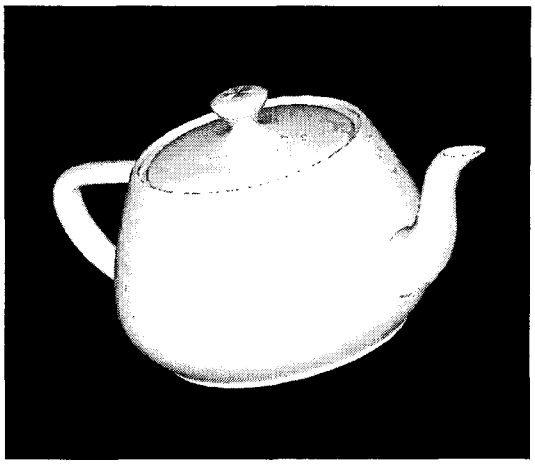

그림 8. Teapot

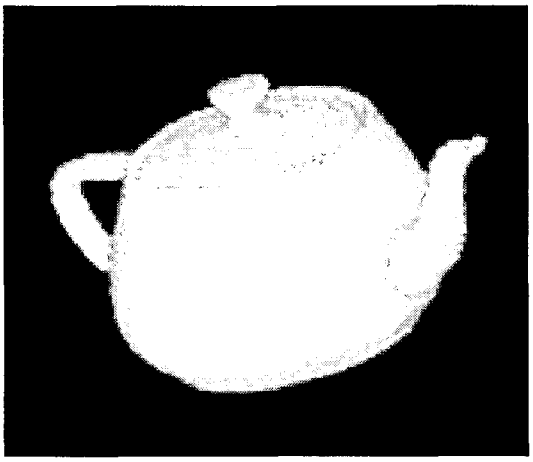

그림 9. Voxel Teapot

참고문헌

1. Kaufman, A., "Efficient algorithms for 3D scan-conversion of parametric curves, surfaces, and volumes," ACM SIGGRAPH, pp. 302-310, 1987
2. Cohen, D. and Kaufman, A., "Scan-conversion algorithms for linear and quadratic objects," In Kaufman, editor, Volume Visualization, pp. 280-301, IEEE Computer Society Press, 1991
3. Kaufman, A. and Shimony, E., "3D

scan-conversion algorithms for voxel-based graphics," Proceedings of ACM Workshop on Interactive 3D Graphics, 1986

4. Kaufman, A., "Efficient algorithms for scan-converting 3D polygons," Computer & Graphics, Vol. 12, No. 2, pp. 213-219, 1988
5. Cohen, D. and Kaufman, A., "Fundamentals of surface voxelization," Graphical Models and Image Processing, Vol. 57, No. 6, pp. 453-461, 1995
6. Juang, J., Yagel, R. Filippov, V. and Kurzion, Y., "An accurate method for voxelizing polygon meshes," IEEE Symposium on Volume Visualization, pp. 119-126, 1998
7. Lee, T. and Requicha, A., "Algorithms for computing the volume and other integral properties of solids," Communication of the ACM, Vol. 25, No. 9, pp. 635-650, 1982
8. Breen, D., "Constructive cube CSG evaluation for display using discrete 3D scalar data sets," Proceedings of the Eurographics'91, pp. 127-142, 1991
9. Wang, S. and Kaufman, A., "Volume-sampled 3D modeling," IEEE Computer Graphics and Applications," Vol. 14, No. 1, pp. 281-292, 1995
10. Breen, D., Mauch, S., and Whitaker, T., "3D scan conversion of CSG models into distance volumes," Proceedings of the 1998 IEEE/ACM Symposium on Volume Visualization, pp. 7-14, 1998
11. Fang, S. and Chen, H., "Hardware accelerated voxelization," Computers & Graphics, Vol. 24, pp. 433-442, 2000