

경계 버퍼를 이용한 볼륨 데이터의 실시간 절단

최임석, 김종용, 송주환, 최성희, 권오봉
전북대학교 컴퓨터과학과

Real-Time Cutting of Volume Data using Boundary Buffer

Y. S. Choe, J. Y. Kim, J. W. Song, S. H. Choi, O. B. Gwon
Dept. of Computer Science, Chonbuk National University

요 약

본 논문은 3차원 볼륨데이터의 실시간 절단에 관한 연구이다. 3차원 데이터를 가시화하는 방법에는 서피스 렌더링과 볼륨렌더링이 있는데, 서피스 렌더링은 처리시간이 적게 걸리는 장점이 있고, 볼륨 렌더링은 데이터의 내부를 가시화할 수 있는 장점이 있다.

본 논문에서는 위 두 가지 방법의 장점을 이용해서, 시점에 변화를 주면서 3차원 데이터를 절단하여 내부를 실시간으로 가시화 할 수 있는 방법에 관해서 기술한다. 또한 3차원 데이터의 저장 공간을 줄여서 효율적으로 메모리를 사용할 수 있게 하였다.

1. 서론

3차원 의료 데이터는 CT나 MR과 같은 장비를 이용하여 촬영된 영상을 3차원으로 재구성하여 사용한다. 3차원 데이터를 가시화하는 방법으로는 서피스 렌더링(Surface Rendering)과 볼륨 렌더링(Volume Rendering)의 2가지 방법이 있다. 서피스 렌더링은 데이터에서 하나의 경계면(surface)에 대해서 폴리곤 또는 패치를 이용하여 재구성한 후 그를 이용하여 가시화하는 방법으로 3차원 데이터중 경계면에 대한 정보만을 사용함으로써 처리시간이 적게 걸리는 장점이 있는 반면 하나의 표면 내부의 다른 표면을 나타낼 수 없다는 단점이 있다. 또한 볼륨 렌더링은 표면 내부를 표현할 수 있지만 데이터 전체를 처리하므로 처리시간이 많이 걸리는 단점이 있다. 본 논문에서는 3차원 데이터를 절단하여 그 내부를 볼 수 있도록 하고, 그 절단이 실시간으로 이루어지도록 하기 위하여 위 두 가지 방법의 장점들만을 이용하였다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안한 실시간 절단에 대하여 설명하고 3장에서는 구현한 내용과 그 결과에 대해서, 그리고 4장에서는 결론 및 향후 연구 방향을 제시하였다.

2. 실시간 절단

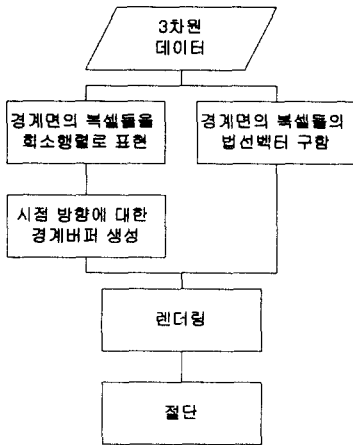
본 논문에서는 3차원 데이터를 가시화하고 그 결과를 실시간으로 절단하는 방법에 대해서 연구하였다. 실시간

으로 데이터를 절단하려고 볼륨 데이터 중에서 각 경계면에 대한 정보만을 추출하고, 그 결과만을 이용하여 절단하므로 많은 시간을 단축시킬 수 있게 되었다. 본 논문에서 사용하는 처리단계는 크게 3단계로 나누어진다. 첫째, 전처리 단계에서는 3차원 볼륨 데이터에서 각 경계면을 구하고 그에 대한 값을 2차원의 최소행렬로 표현하고, 경계면의 각 복셀의 법선 벡터를 구해 버퍼에 저장한다. 둘째, 렌더링 단계에서는 시점 방향에 변화가 있을 때마다 교차버퍼에 각 시점 방향에서 출발한 광선과 만나는 복셀의 인덱스를 저장한다. 마지막으로 근거리버퍼와 원거리버퍼를 두어 이 버퍼를 이용하여 절단된 면을 렌더링한다[그림 1].

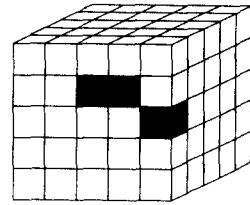
2.1 전처리 단계

3차원 의료 데이터는 매우 많은 용량의 저장공간을 필요로 한다. 예를 들어 x, y, z축으로 각각 256의 크기를 갖고, 각 복셀은 0~255의 값을 갖는다면 이 데이터의 크기는 16,776,2166Bytes(256×256×256)로 매우 커다란 저장 공간을 필요로 한다. 그러나, 본 논문의 절단 프로세스에서 필요한 값을 가지고 있는 부분은 전체 중 1/100정도로 매우 적다. 이를 위하여 3차원으로 저장된 데이터를 2차원의 최소행렬을 이용하여 표현한다[1].

이때 각 복셀이 가지고 있는 값은 그 복셀 주위의 값들을 이용하여 구한 법선벡터의 값과 함께 새로운 데이



[그림 1] 실시간 절단 프로세스



볼륨 데이터

index	X좌표	Y좌표	Z좌표	voxel_value & normal
0	2	1	0	voxel_value0, fx0, fy0, fz0
1	3	1	0	voxel_value0, fx1, fy1, fz1
2	4	2	0	voxel_value0, fx2, fy2, fz2
...
n	Xn	Yn	Zn	voxel_valuen, fxn, fyn, fzn

[그림 2] 최소 행렬로 표현된 경계 버퍼

터 구조로 저장하고 이 두 구조 사이를 인덱스를 이용하여 연결하였다. 이때, 3차원 데이터에서 최소행렬에 포함되는 값은 각 기관이나 조직의 경계면에 해당하는 복셀들을 저장한다.

최소행렬의 자료구조와 저장되는 각 복셀의 자료구조는 다음과 같이 표현한다.

```

struct{
    int index;
    byte x, y, z;
    byte voxel_value;
    float fx, fy, fz;
}
    
```

여기에서 fx, fy, fz는 V(x,y,z) 복셀의 법선벡터 다음 식을 사용하여 구한다.

$$\begin{aligned}
 fx &= \frac{f(x_{i+1}, y_j, z_k) - f(x_{i-1}, y_j, z_k)}{2} \\
 fy &= \frac{f(x_i, y_{j+1}, z_k) - f(x_i, y_{j-1}, z_k)}{2} \\
 fz &= \frac{f(x_i, y_j, z_{k+1}) - f(x_i, y_j, z_{k-1})}{2}
 \end{aligned}$$

[그림 2]에 경계버퍼의 생성 과정과 자료구조를 표현하였다. 이때 경계 버퍼를 쉽게 검색할 수 있도록 x, y, z의 각각에 대하여 정렬하였다.

2.2 렌더링

3차원 데이터를 최소 행렬로 표현한 경계버퍼를 그대로 사용하는데는 시점의 방향에 대해서 재구성해야 하는 문제점이 발생한다. 이 문제점을 해결하고자 교차버퍼를 생성하는데, 이 교차버퍼는 스크린의 각 픽셀에서 출발한 각 광선이 볼륨 데이터를 통과하면서 만나는 경계면의 위치를 스크린의 각 픽셀에 대해 깊이에 대해 오름차순으로 정렬하여 만들어지는 리스트이다. 이때 각 광선이 볼륨 데이터를 통과하면서 경계면을 찾는 방법으로

300DA 알고리즘을 사용하여 처리시간을 감소시켰다[2] [그림 3].

```

list = NIL;
do{
    if(tMaxX < tMaxY){
        if(tMaxX < tMaxZ){
            X = X + stepX;
            if(X == justOutX)
                return(NIL);
            tMaxX = tMaxX + tDeltaX;
        }
        else{
            Z = Z + stepZ;
            if(Z == justOutZ)
                return(NIL);
            tMaxZ = tMaxZ + tDeltaZ;
        }
    }
    else{
        if(tMaxY < tMaxZ){
            Y = Y + stepY;
            if(Y == justOutY)
                return(NIL);
            tMaxY = tMaxY + tDeltaY;
        }
        else{
            Z = Z + stepZ;
            if(Z == justOutZ)
                return(NIL);
            tMaxZ = tMaxZ + tDeltaZ;
        }
    }
    list = ObjectList[X][Y][Z];
}while(list == NIL);
return(list);
    
```

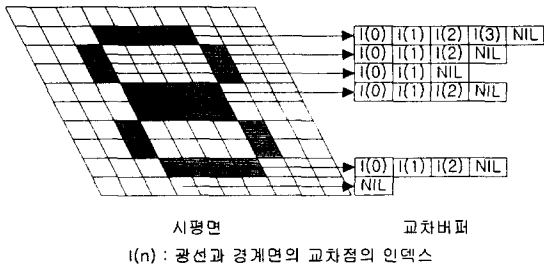
[그림 3] 3차원 ODA 알고리즘

이렇게 만들어진 교차버퍼의 구조는 [그림 4]에 나타내었고, 교차버퍼를 이용하여 가시화 할 때 음영은 품 음영법을 사용하였다.

$$\begin{aligned}
 c(x_i) &= c_p k_a + \frac{c_p}{k_1 + k_2 d(x_i)} \\
 &\times [k_d(N(x_i) \cdot L) + k_s(N(x_i) \cdot H)^n]
 \end{aligned}$$

2.3 절단

절단은 절단면들의 집합 C₁, C₂, ..., C_m으로 정의된



[그림 4] 교차버퍼

	경계면 수	볼륨 크기	경계면 자료량
구	3	256× 256× 256 (16MB)	226,846개
머리부분 (CT)	3	256× 256× 109 (6.8MB)	182,040개

[표 1] 사용 데이터

다. 절단을 위해서 추가로 2개의 버퍼 (D_{near} 와 D_{far})가 필요하다. 이 두 개의 버퍼 각각은 스크린과 같은 해상도를 갖고, 버퍼의 각 요소는 스크린의 각 픽셀과 일대일 대응한다. 버퍼 D_{near} 의 모든 값은 0으로 초기화되고 D_{far} 의 모든 값은 MAX_DEPTH로 초기화된다[3].

스크린의 각 픽셀 p에 대하여, 절단면들의 픽셀에 투사된 광선 R(p)을 따라서 다음과 같이 3가지의 형태로 분류되고 각 오퍼레이션은 근거리버퍼와 원거리버퍼에 대해 다음과 같이 수행한다.

1) 절단면이 광선에 평행 : $R(p) \cdot N_i = 0$

$$D_{near}(p) = MAX_DEPTH$$

$$D_{far}(p) = 0$$

2) 절단면이 광선과 같은 방향 : $R(p) \cdot N_i > 0$

$$D_{near}(p) = \max(D_{far}(p), depth)$$

3) 절단면이 광선과 반대 방향 : $R(p) \cdot N_i < 0$

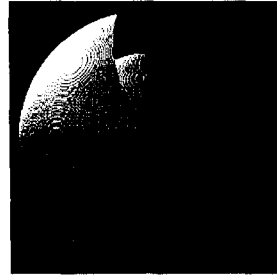
$$D_{far}(p) = \min(D_{far}(p), depth)$$

이렇게 구한 두 개의 버퍼는 렌더링 할 때 경계면의 복셀이 화면에 출력되어야 할지, 그렇지 않을 지에 대해 결정하는데 사용한다. 즉, D_{near} 보다 가까운 복셀과 D_{far} 보다 먼 복셀은 렌더링 할 때 모두 버리고 그 사이의 복셀만을 렌더링하여 표현한다.

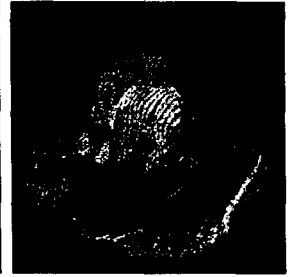
3. 구현 및 결과

본 실험은 Pentium II 333MHz, RAM 64MB의 하드웨어 환경과 Windows 98에서 Visual C++ 6.0을 사용하여 구현하였다. 본 실험에서 사용한 데이터는 [표 1]에 정리되어 있다.

구의 데이터는 임의로 256× 256× 256(16MB) 크기의 지구 내부를 표현한 것과 같이 만들어서 사용하였다. 전체가 모두 3개의 서로 다른 물질로 이루어져 있고 이를 절단한 후 렌더링한 결과를 [그림 4]에 나타내었다. 그리고 머리부분의 데이터는 CT로 촬영된 인간의 머리부분 영상으로 머리 바깥 부분인 공기와 머리의 표피 부분, 그리고 뼈 부분, 뇌 부분의 4가지로 분류하였고 이를 절단한 후 렌더링한 결과를 [그림 5]에 나타내었다.



[그림 4] 구의 절단



[그림 5] 머리의 절단

4. 결론 및 향후 연구

본 논문에서는 3차원 데이터를 실시간으로 절단하여 그 결과를 나타낼 수 있도록 전처리 단계에서 각 경계부분에 대한 정보를 미리 추출하여 저장할 수 있는 경계버퍼를 사용하고, 시점 방향의 변화에 따라 시점에서 출발한 광선이 볼륨 데이터를 통과하면서 교차하는 교차점을 저장하는 교차버퍼를 사용한다.

이 알고리즘은 서피스 렌더링의 빠른 수행속도와 볼륨 렌더링의 내부를 표현할 수 있는 두 장점 모두를 나타낼 수 있어 매우 효과적으로 3차원 데이터를 표현할 수 있다는 것을 증명한다. 또한 자료의 저장공간을 매우 효과적으로 줄일 수 있어 의료 데이터를 실시간으로 절단, 렌더링 할 수 있게 되었다. 그러나, 경계버퍼를 이용하여 절단하는데 뒷부분의 경계면들이 나타나지 않는 문제점이 발생했는데 향후 이를 해결하도록 할 계획이다.

참고문헌

- [1] 송주환, 김무영, 권오봉, 김용성, 장옥배, "경계 정보 테이블을 이용한 볼륨렌더링", 한국정보과학회 '96 봄 학술발표논문집 Vol.23, No.1 pp.727-730, April 1996
- [2] John Amanides, and Andrew Woo, "A Fast Voxel Traversal Algorithm for Ray Tracing," EUROGRAPHICS '87, pp.3-10, 1987.
- [3] Mara .W. Jones and Min. Chen, "Fast cutting operations on three dimensional volume datasets." Visualization in Scientific Computing (ed. M. Goebel, Hueller and B. Urban), Springer-Verlag, Berlin (1995) 1-8