연속 함수를 이용한 볼륨 데이터의 렌더링

노현아,^{O} 김재정
전남대학교 전산학과
hyuna^{O}@dal.chonnam.ac.kr, jaykim@chonnam.ac.kr

Volume Rendering Based On a Continuous Function

Hyuna Noh,^{O} Jayjeong Kim
Dept. of Computer Science, Chonnam National University

요 약

MRI 나 CT 스캔에 의해 생성된 볼륨 데이터는 일반적으로 여러 샘플 지점에서의 이산적인 수치 데이터 일 뿐 데이터 상호간의 함수적 연속성은 제공되지 않고 있다. 이러한 데이터로부터 우리가 원하는 임계값(threshold)에 의한 등가면(isosurface)을 렌더링하는 방법은 보통 Marching Cube에서처럼 많은 다각형을 생성해서 렌더링 하는 방법에 의존해 왔다. 그러나 원하는 등가면을 직접 표현할 수 있는 함수가 존재할 경우 많은 양의 다각형을 추출하고 보관해야 하는 시공간적 부담이 없게 된다. 본 논문에서는 각 Cube별로 정의되는 Tri-linear Interpolation 함수를 기반으로 하여 Interval Method 에 의한 등가면 렌더링 알고리즘을 제안한다.

1. 서 론

MRI 나 CT 스캔에 의해 생성된 단층 촬영 볼륨 데이터는 일반적으로 3D Regular Grid에서의 이산적인 수치 데이터 만을 기억하고 있을 뿐 데이터 상호간의 함수적인 연속성에 대한 정보는 전혀 제공되지 않고 있다. 이러한 데이터로부터 우리가 원하는 어떤 임계값(threshold)에 의해 결정 되어지는 등가면을 렌더링 하는 방법은 보통 Marching Cube[7] 알고리즘이나 그에 유사한 다른 변형된 알고리즘, 또는 볼륨 렌더링 등의 알고리즘에 의존해 왔다.

Marching Cube알고리즘은 보통 수많은 삼각형을 추출하여 그것을 렌더링 하는 방법으로서, 주어진 Threshold에 따라 정의된 등가면을 포함하는 큐브에서 추출된 수많은 삼각형들을 별도로 기억하고 처리하기 위한 시간적 공간적 부담이 따르게 된다. 반면에 만약 등가면을 직접 표현할 수 있는 함수가 있다면 그러한 시공간적 부담은 줄어드는 대신 그 등가면을 직접 렌더링 해야 하는 작업이 필요하게 되는데 이와 같은 직접 렌더링은 함수의 종류에 따라 난이도가 크게 달라지게 되며 Bezier surface와 같은 비교적 쉬운 함수로 표현된 곡면일지라도 광선 추적법으로 그를 렌더링 하는 알고리즘은 대단히 복잡하다. 그러나 이러한 함수 일지라도 함수 자체가 매우 특별한 연산이 아닌 일반적인 산술 연산만으로 구성된 단순한 함수일 경우엔 Interval Method [4] 라 하는 매우 탁월한 기법을 Ray Casting 에 적용하여 이미지 렌더링이 가능하게 되는데 이러한 Interval Method는 1차 수렴 속도를 가지기 때문에 Newton과 같은 2차 이상의 수렴 속도를 가지는 알고리즘에 비해 다소 느리게 느껴지지만 보통 함수의 종류나 유형에 민감하지 않고 어떤 경우에도 잘 적용되는 방법으로서 Bezier함수에도 잘 적용된다.

본 논문에서는 한 큐브 내에서 Tri-linear Interpolation을 위한 3차 근사식을 그 큐브 내에서의 물체를 나타내는 연속 함수로 정의하고 여기에 Interval Method를 적용함으로써 이미지를 렌더링하는 알고리즘을 제안하였다. Ray Casting은 이미지를 가시화 하기 위한 알고리즘이지만 Interval Method를 적용하기 위해서는 약간의 좌표계 변환이 필요하다. 즉 Interval Method를 적용할 때는 xyz 좌표계의 3개의 Interval을 반복 분할하는 과정을 통해서 원하는 근에 접근해 가게 되는데, 광선과 등가면과의 교점을 구하고자 할 때는 보통 광선이 하나의 매개 변수 t 에 의해 나타내어 지는 점을 감안하여 변수 치환 과정을 통해서 xyz대신 광선의 매개변수 하나에 대해서만 분할을 시행하여 빠르고 간편하게 렌더링을 할 수 있게 된다. 그러나 그러한 변수의 치환은 각각이 정의된 좌표 공간이 동일할 때에만 가능하다. 즉 광선이 정의된 좌표계와 Tri-linear Interpolation이 정의되는 각 큐브 내에서의 좌표계가 다르기 때문에 전자와 같은 치환은 좌표계의 변환이 선행된 후에야 가능하게 된다. Tri-linear Interpolation은 특정 threshold에 의한 등가면을 나타내기 위해 각 큐브 내에서 정의되는 연속 함수 인데 이를 이용하면 Marching Cube처럼 삼각형을 따로 추출할 필요가 없게 된다.

2. Marching Cube 알고리즘

평면 Regular Grid데이터를 여러 개 쌓아서 만들어진 3D 볼륨 데이터는 주어진 임계값 T 에 의해 정의되어 지는 등가면을 렌더링 하는데 쓰이게 되는데 이와 같은 형상을 찾는데 보통 사용되는 알고리즘이 Marching Cube 이다. 이러한 알고리즘에서는 한 쌍의 인접한 슬라이스 내에서 여러 개의 인접한 가상의 큐브들을 만든 후 각각의 큐브 내에서 T 에 맞는 등가면을 나타낼 삼각형을 추출하게 된다. 그렇게 해서 생성된 대량의 삼각형들은 삼각형을 따로 처리하는 렌더링 엔진으로 보내지고 거기서 이미지가 그려지게 된다. 이 알고리즘은 처음 발표됐을 때 알려지지 않았던 몇 가지 오류가 있었으나 그 후 큐브 내에서 바로 삼각형을 추출해 내는 대신에 하나의 큐브를 5개의 삼각뿔(tetrahedron)로 쪼개어 삼각형을 추출하는 등의 수정된 알고리즘이 선보인 후 비로소 안정적으로 사용되기 시작했다. 그러나 어떤 경우에도 항상 삼각형들을 미리 추출해야 하는 불편함이 있고 또 T 값이 달라지면 또다시 새로운 삼각형들을 추출해야만 하는 번거로움이 있다. 따라서 implicit surface와 같이 물체를 직접 나타낼 수 있는 함수가 있다면 이렇게 삼각형을 미리 따로 추출해야 하는 등의 문제는 사라지게 된다.

3. 연속 함수의 정의: Tri-Linear Interpolation

이산적인 3D볼륨 데이터에서는 일반적으로 수치 데이터만 존재할 뿐 그 안에서 어떤 등가면을 나타내거나 정의할 수 있는 함수는 존재하지 않는다. 그러나 이 데이터는 marching cube에서 보는 바와 같이 수많은 작은 큐브들로 구성되어 있는 것으로 볼 수 있다. 이때 한 큐브 내에서의 임의의 한 점의 값을 보간하기 위해 8개의 꼭지점들을 이용하는 tri-linear interpolation을 고려한다면 그 3차 보간식은 그 큐브 내의 모든 임의의 점들에 대한 데이터의 연속성을 제공하는 함수로 사용될 수 있으며, 또 그러한 큐브들 간에는 piecewise continuity가 성립되는 것을 알 수 있다. 따라서 한 큐브 내에서 특정 threshold에 의해 정의되는 등가면을 찾기 위해서는 이와 같은 tri-linear interpolation 식을 등가면을 나타내는 함수로서 사용하게 되면 별다른 어려움 없이 처리되어 질 수 있게 되며, 그 식을 자세히 정의하면 다음과 같다.

볼륨 데이터상의 한 위치 또는 그 위치를 기점으로 하고 그 부근의 8개의 정점들을 꼭지점으로 하는 큐브를 나타내는 표기로서 (I,J,K)를 사용하기로 하자. 이때 그 8개의 꼭지점 $P_0, P_1, \ldots, P_7$ 의 위치를 정하기 위해 P_n을 위치 (I+i, J+j, K+k) 에서의 꼭지점(단 n = 4i + 2j + k 이고 i,j,k=0,1) 이라 하면 큐브 (I,J,K)는 P_0를 정점으로 하는 큐브가 된다. (그림 1)

또한 큐브 내에서의 한 점 p 의 위치를 나타내기 위해 (x,y,z) 를 p 의 큐브 내에서의 Barycentric coordinate 라 하자. 즉

$$\overrightarrow{P_0P} = x \cdot \overrightarrow{P_0P_4} + y \cdot \overrightarrow{P_0P_2} + z \cdot \overrightarrow{P_0P_1}$$

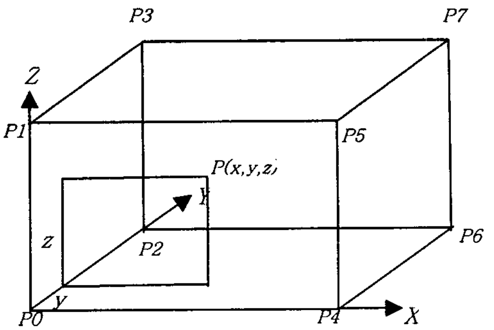

그림1: 큐브의 꼭지점들과 그 안에서의 한 점 P의 좌표

그러면 이는 p 가 그 큐브 내에서 사실상 다음과 같은 3 차 보간식으로 나타내어 지는 것과 동일해 진다.

$$\begin{aligned} P = \sum_{n=0}^{7} w_n P_n = P_0 \cdot (1-x)(1-y)(1-z) \\ + P_1 \cdot (1-x)(1-y)z \\ + P_2 \cdot (1-x)y(1-z) \\ + P_3 \cdot (1-x)yz \\ + P_4 \cdot x(1-y)(1-z) \\ + P_5 \cdot x(1-y)z \\ + P_6 \cdot xy(1-z) \\ + P_7 \cdot xyz \end{aligned}$$

물론 이 보간식은 변수 *x, y, z* 에 대한 3 차식(*cubic*)이 되며 그 변수들의 영역은 큐브 내에 국한 되는 영역, 즉

$$(x,y,z) \in [0,1] \times [0,1] \times [0,1]$$

이 된다.

4. Interval Method 에 의한 Ray Casting

삼각형들을 미리 추출하여 ray casting 알고리즘 등으로 등가면을 렌더링하는 Marching Cube 와는 달리 본 논문에서는 등가면을 직접 나타내는 함수를 이용하여 렌더링하고 있다.

광선 R 의 시점과 방향 벡터를 각각 O 와 V 라 하면 R 은 매개변수 t 에 의해 R = O + tV 로 나타내 진다. 그러나 이 광선을 정의하는 좌표계는 일반적으로 tri-linear interpolation 을 적용하는 각각의 큐브 내에서의 좌표계와 다르기 때문에 사전에 이를 일치시켜야 하지만 이 작업은 간단한 좌표계 변환에 의해 시행될 수 있으므로 큰 문제는 아니다. interval method 는 모든 변수에 대해서 사전에 주어진 영역이 초기 interval 이 되며 여기서 광선 R 이 큐브 내에서의 등가면과 만나는 점을 찾기 위해선 광선의 각 성분, 즉 x = Ox + tVx, y = Oy + tVy, z = Oz + tVz 를 3 차 보간식에 대입하여 t 에 관한 식으로 바꾼 후 t 의 큐브 내에서의 범위를 미리 구하여 초기 interval 로 사용한다.

다음 그림 2와 그림 3은 큐브와 삼각뿔 형태로 배치된 exponential blob 으로부터 추출된 3 차원 이산 데이터로부터 원래의 이미지를 복원시킨 결과이다. 그림 4는 CT 데이터로부터 서로 다른 등가면을 보인 결과로서 이러한 결과들은 본 논문의 알고리즘을 이용하여 서로 다른 등가면을 렌더링하는데 있어서도 우수한 성능을 보임을 알 수 있다.

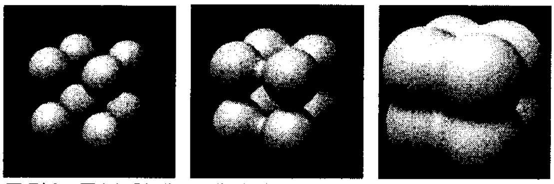

그림2: 큐브 형태로 배치된8개의exponential blob의 이산 데이터로부터 서로 다른 등가면들에 대해 복원된 이미지

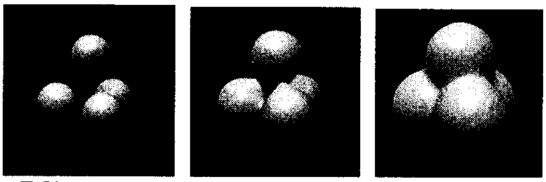

그림3: 삼각뿔 형태로 배치된4개의exponential blob의 이산 데이터로부터 서로 다른 등가면들에 대해 복원된 이미지.

그림4: 113x256x256 CT 데이터로부터 서로 다른 등가면에 대해 복원된 이미지.

5. Non-regular Grid 로의 일반화

대부분의 볼륨 데이터는 regular grid 데이터이지만 만약 임의의 샘플링에 의해 추출된 random grid 데이터라면 단순히 한 쌍의 슬라이스에 의한 큐브들의 집합체만으로 처리하기엔 복잡한 문제가 있다. 따라서 이러한 경우에는 먼저 전체 grid를 삼각뿔화(tetrahedralization)하는 알고리즘을 적용하여, cube대신 tetrahedron내에서 같은 아이디어를 적용하면 된다.

삼각뿔화하는 알고리즘은 전처리 과정에서 하게 되므로 기존의 어떤 알고리즘을 적용해도 상관이 없다.

하나의 삼각뿔의 네 꼭지점을 p_1, p_2, p_3 와 p_0 라 하고 (x,y,z) 를 내부의 한 점p 의 Barycentric coordinate라 하면, 즉

$$\overline{p_0 p} = x\cdot\overline{p_0 p_1} + y\cdot\overline{p_0 p_2} + z\cdot\overline{p_0 p_3}$$

그러면 같은 논리로

$$(x,y,z)\in[0,1]\times[0,1]\times[0,1]$$

와

$$x+y+z\le 1$$

가 되고, 광선과의 교점도 t에 관한 interval method를 적용하여 이미지 렌더링이 가능하게 된다.

6. 결론 및 향후 연구 방향

지금까지 우리는 한 큐브 내에서 tri-linear (cubic) interpolation 식이 주어진 볼륨 데이터로부터 임의의 등가면을 직접 렌더링할 수 있는 연속함수로서 이용될 수 있음을 보았다. 그리고 이와 같은 렌더링은 interval method라는 매우 효율적인 알고리즘을 ray casting에 적용해서 이루어 졌다. 그러나 이와 같은 연구는 향후 등가면의 위상의 변화를 예측할 수 있는 critical point의 계산에도 적용될 수 있도록 연구가 되어져야 한다. regular grid가 아닌 일반적인 random grid에 대해서도 적용할 수 있도록 알고리즘을 확장 할 수 있다. 물론 그러기 위해선 데이터를 먼저 trahedralization을 먼저 시행해야 함은 물론이다.

7. 참고문헌

[1] B. Wyvill, Metamorphosis of Implicit Surfaces, *SIGGRAPH '93 Course Note* 25, 1993.

[2] B. Stander, J. Hart, Guaranteeing the Topology of an Implicit Surface Polygonization for Interactive Modeling, *SIGGRAPH '97 Annual Conference*, Aug. 1997.

[3] D. Mitchell, Three applications of interval analysis in computer graphics, Frontiers of Rendering. *SIGGRAPH '91 Course Note*, 1991.

[4] R.E. Moore, *Interval Analysis*, Prentice Hall, 1966.

[5] C.L. Bajaj, D.R. Schikore, Topology Preserving Data Simplification with Error Bounds, *Computer & Graphics*, Vol.22, No.1, pp.3-12, Pergamon, 1998.

[6] C.L. Bajaj, V. Pascucci, D.R. Schikore, Fast Isocontouring for Improved Interactivity, *Purdue University*.

[7] W.E. Lorensen, H.E. Cline, Marching Cubes: A High Resolution 3D Surface Construction Algorithm, *SIGGRAPH '87 Annual Conference*, Vol.21 No.4, 163-169, August 1987.