

의료영상의 체적가시화를 위한 가속 알고리즘에 관한 연구

임현우*, 이동혁*, 정용규**

* 마로테크 의료영상연구소

** 서울보건대학, 현재 경기대학교 박사과정

A Study on Accelerative Algorithm for Medical Images Volume Rendering

Hyun Woo Lim*, Dong Hyuk Lee*, Yong Gyu Jung**

* Medical image research center, Marotech Co.

** Seoul Health College, Doctral Course of Kyonggi University

요 약

체적가시화(Volume Rendering)는 단면촬영기나 표면인식기 등을 이용해 얻어 들인 Data를 원래의 형태로 화면상에 보여 주는 것으로 일반적인 방법이 Surface Rendering과 Volume Rendering이 있다. Volume Rendering은 Data 처리속도 문제와 한정적인 메모리 양으로 인해 기존의 알고리즘을 그대로 적용하는 경우 실시간 가시화가 힘들 뿐만 아니라 3차원 영상의 질이 저하되는 문제가 있었다. 따라서, 본 연구는 3차원 영상의 질 저하 없이 실시간으로 MR Angio의 3차원 Volume 가시화를 구현한다. 본 연구에서 사용되는 속도 개선 알고리즘은 Marc Levoy가 제안한 8진Tree(Octree) 자료구조를 이용하며, 또한 Volume Data 내에 존재하는 공기와 같이 가시화될 필요가 없는 부분에 대해 불필요한 계산을 피하고 가시화하고자 하는 부분만을 계산함으로써 Rendering에 소요되는 시간을 줄이는 방법을 사용한다.

Key words : 의료영상, 체적가시화, Rendering, Octree, Raycasting, MR Angio

서 론

본 연구는 의료영상을 3차원으로 Rendering하는 보다 정확하고 효과적인 방법을 제시한다. 단면영상으로 구성된 의료영상 데이터는 관심있는 영역과 주변의 인체 장기에 의한 Artifact와 주변 Noise인 비관심 영역으로 임의적으로 나눌 수 있다. 보통은 관심영역과 비관심 영역을 분리하기 위해 Density level과 Gradient 크기를 이용한다. 그러나 이 방법은 정확히 주위 Artifact와 Noise를 제거하지 못한다. 또한 Segmentation 과정을 거쳐 정확히 영상을 구하기 어렵다. 비관심 영역을 분리하는 방법은 가속 알고리즘에 따라 달라진다. 즉, 관심 영역을 미리 전처리 단계에서 추출하고 가속을 위해 Volume 데이터는 Run-length code로 변환하거나, Octree (8진트리)와 같은 계층적인 데이터 구조로 변환하여 관심 없는 영역을 빠르게 지나가 메모리를 참조하는 비율을 줄인 것이다. 위의 경우 미리 정확하게 segmentation이 되어야 한다. 참고적으로 나머지 가속 알고리즘은 각 Voxel(복셀)당 연산을 줄이는 것이 문제이다. 이를 위해 Gradient가 미리 계산되고 차지하는 메모리 영역을 줄이기 위해 정량화하여 부호화를 한다.

본 연구에서는 간단한 Segmentation 과정을 통해 비관심 영역을 계산하지 않음으로써 3차원적인 형태 Rendering을 가속화하는 방법을 제시한다. 이를 위해 3차원 projection 에 많이 사용되는 Ray Casting(광선추적법)과 Octree를 사용하였다. Ray Casting은 투영되는 방향으로 데이터를 샘플링하는 방법으로 관찰자 기준의 좌표에서 Z 방향으로 진행해 나가며, 이 좌표를 역좌표 변환의 과정을 거쳐 데이터 기준의 좌표로 변환하여 보간법 (Interpolation)을 하는 과정으로 구성된다. Volume Rendering의 경우 각 Voxel들은 조명모델에 의해 정해진 밝기를 샘플링한 후 이 값들을 반투명한 물체로

인식, 합성 (Blending)을 하게 된다.

조명모델을 사용할 경우 각 Voxel에서 Gradient를 구해야 한다. 이 과정은 시간이 많이 소요되고 또한 미리 계산을 할 경우 많은 메모리를 차지하므로 비효율적이다.

Octree 구조를 이용하여 Rendering 속도를 빠르게 한다. Octree 구조는 Volume 데이터를 X, Y, Z 방향으로 반복적으로 이등분함으로써 8개의 노드로 구성된 레벨들의 Tree로 구성한다. 각각의 노드는 최대, 최소값을 저장하고 있어 Ray Casting시 Volume 데이터의 많은 부분을 차지하는 공기와 같이 빈 공간을 뛰어 넘을 수 있어 속도를 빠르게 할 수 있다.

관 련 연 구

MR Anigo로부터 산출되는 의료영상 데이터에 대하여 보다 현실적이고 사실적인 영상을 제공하기 위해서 Volume Rendering기법을 통해 3차원 영상으로 Rendering 한다. Volume Rendering 기법은 Volume 데이터를 구성하고 있는 Voxel위에 정의된 스칼라(Scalar)와 벡터(Vector) 데이터로부터 영상을 얻는 기술을 말한다.

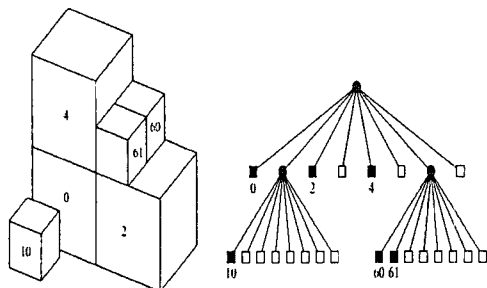
Volume Rendering 알고리즘(Algorithm)은 영상공간(Image Space) Rendering 방법과 물체공간(Object Space) Rendering 방법으로 나뉘어져 있다. 영상공간 Rendering 방법은 영상 평면(Plane)의 각 픽셀(Pixel)에서 광선을 쏘아서 일정한 간격으로 샘플링하여 영상을 생성해내는 알고리즘이다. 대표적인 영상공간 Rendering 방법에는 Ray Casting 방법이 있다. 물체공간 Rendering 방법은 Volume 데이터를 순차적으로 순회하면서 Voxel을 영상평면으로 투영하여 영상을 생성하는 알고리즘으로 Splatting 방법이 이에 속한다. 이 방법은 광선 샘플링의 조기 종결을 구현하기 힘들다.

일반적으로, 우수한 영상의 질과 효과적인 Rendering를 위해서 Ray Casting 방법에 기반을

두고 있는 영상 공간 Rendering 기법을 많이 사용한다. 이 방법은 좋은 질의 의료 영상을 나타 내기는 하지만, Rendering 속도가 비교적 느려 하나의 영상을 만드는데 많은 시간이 요구된다. 따라서, 이러한 Ray Casting 방법을 효과적으로 가속화하는 알고리즘이 요구된다.

Rendering 알고리즘에 많은 시간이 필요한 것은 처리해야 할 데이터 양이 많기 때문이다. 이에 Volume 데이터 내에 존재하는 공기와 같이 Rendering될 필요가 없는 부분에 대한 불필요한 계산을 피하기 위해 Octree, 피라미드, k-D Tree (k진 트리)등과 같은 계층 구조를 갖는 자료 구조들을 사용하여 가속화하여 왔다. 물론, 이런 가속화를 위한 계층 자료 구조는 그 자료 구조를 순회하는 작업으로 인하여 추가적인 비용이 필요하다.

Octree를 사용하는 Ray Casting 방법에서 각 광선이 계층구조를 방문하는데 드는 중복된 계산은 [그림 1]에서 보는 바와 같이 Octree를 한번만 방문하면서 각 Volume 데이터의 공간에서 광선을 찾는 방법으로 대신하는 것으로 없앨 수 있다. Octree의 노드에 해당하는 공간에서 광선이 지나는 지점을 효과적으로 찾고, 색상과 투명도를 계산할 수 있어야 한다. 색상과 투명도의 계산은 일반적으로 3차원 샘플링에 의해 이루어진다. 3차원 샘플링은 한 지점에서의 값을 알아내기 위해 많은 보간 계산이 요구된다. 비슷한 영상의 질을 유지하면서 보간 계산을 줄이는 것은 Rendering의 속도향상에 도움이 된다.



[그림 1] Octree Algorithm

위와 같은 방법으로 Octree (Octree)를 구성하고 나서 실제 Rendering시에는 Tree 순회를 통해 각 노드에서 최대/최소 값을 비교하여 Rendering에 사용되는 블록과 계산을 하지 않아도 되는 블록을 결정하게 된다.

Ray Casting 방법에서는 각 광선마다 공기와 같은 영역에 대한 불필요한 샘플링을 피하기 위해 매번 Octree를 계층적으로 순회해야 하는 중복된 계산을 포함하게 된다. Cohen은 계층구조의 순회에 드는 추가비용을 줄이기 위해 플랫폼 피라미드라고 하는 자료구조를 사용한다.

본 연구는 의료 데이터의 효과적인 Rendering를 위한 Ray Casting Rendering 모듈을 개발하고, 이를 가속하기 위한 Ray Casting 가속화 Algorithm을 개발하여 Rendering 속도를 향상시키는 것을 목표로 한다. 이를 위해 Rendering 모듈을 사용하기 위한 인터페이스를 정의하고, Rendering 모듈을 설계하며, 기존의 잘 알려진 방법 중의 하나인 Octree에 기반을 둔 Ray Casting 방법을 구현하고, 2차원 보간법에 기반을 둔 가속 Ray Casting 방법 모듈을 개발하였다

Ray Casting Rendering 방법은 Volume 데이터의 3차원 Rendering에서 사용하는 가장 대표적인 방법으로 좋은 영상을 제공한다. Ray Casting 방법은 비교적 Rendering에 많은 시간을 소모하는데 이를 효율적으로 단축시키기 위해 Levoy에 의해 Octree에 기초한 Ray Casting 방법이 제안되었다.

본 연구에서는 이를 확장하여, 동적인 Rendering 물질변화에 맞도록 최대/최소 Octree 구조로 변형하여 효과적인 Ray Casting 방법이 수행되도록 하였다. 또한 영상영역에서의 범위 Tree를 추가적으로 사용하여 그 속도를 더욱 향상시킬 수 있었다. 2차원 보간법에 기반을 둔 가속 Ray Casting 방법은 여러 종류의 데이터에 대해 실험한 결과를 통해 효율적인 속도 향상을 가져왔음을 확인할 수 있었다.

Ray Casting 모듈의 설계

Ray Casting에 소요되는 시간을 단축하기 위하여, Levoy가 제안한 Octree 자료구조를 사용하였다. 이 방법은 Rendering 하고자 하는 부분의 데이터 응집성을 이용하여, 보다 효율적인 Ray Casting 방법을 구현한다. 가령, 머리에 대한 Volume 데이터를 가지고 뇌를 Rendering하고자 할 경우, 실제 Rendering 연산이 필요한 부분은 전체 데이터의 일부분에 불과하다. 따라서, 뇌의 위치를 미리 파악하고 있다면, 광대한 Volume 데이터를 처리하는데 드는 불필요한 계산 비용을 제거할 수 있을 것이다. 우리는 Volume 데이터 블록의 최대/최소 데이터 값을 포함하는 Octree를 구성함으로써 Rendering 하는 부분의 위치를 빠르게 파악하도록 하였다.

우선, 원시 Volume 데이터 즉 Tree의 루트노드를 X, Y, Z 방향에 대해서 각각 이등분하여 8개의 블록을 얻는다. 이때, 각 블록들은 Octree의 레벨 1의 8개의 노드를 의미한다. 이렇게 얻은 각 블록들에 대해서 이 과정을 반복하여 수행하면, 레벨 i 은 8ⁱ개의 노드를 갖게 된다. Levoy의 논문에서는 Rendering을 하려는 물체의 데이터 값의 범위를 고정하여 사용자가 Rendering을 원하는 Volume 데이터의 값을 변화시킬 수 없었다. 따라서, 각각의 노드에는 그 노드의 8개의 자식들이 모두 불투명도가 0이면 0값이, 아니면 1값이 저장되는 1비트의 노드로 구성된 Octree를 사용하게 된다.

그러나 본 연구에서는 사용자가 인터페이스를 통하여 Rendering할 데이터 값의 범위를 원하는 대로 지정할 수 있게 하기 위해서 한 노드마다 2개의 바이트를 사용하여 노드의 자식들에 대한 최대/최소 데이터값을 저장하게 하는 자료 구조를 사용하였다.

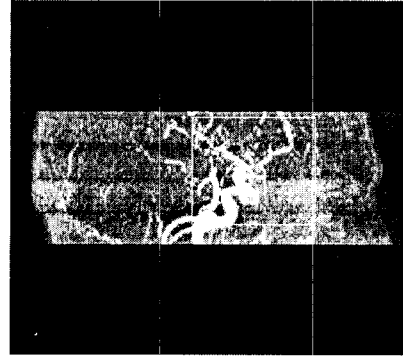
앞에서 말한 것과 같이 Octree를 구성한 다음에, 각 광선에 대하여 제일 상위

레벨에서의 광선과 블록과의 교차점을 계산한다 Octree 노드에 대해 해당되는 각 레벨의 데이터 블록을 하나의 셀로 정의할 때, 광선이 그 셀과 만나면 셀에 대한 Octree 노드의 최대/최소 값을 비교해 보고, 그 값이 Rendering해야 하는 범위에 있다면, 레벨을 하나 증가시켜서 다음 레벨에서의 셀에서 다시 최대/최소 값을 비교하게 된다. 만약에, 최대/최소 값의 범위가 Rendering 해야 하는 데이터 값의 범위를 벗어난다면, 그 셀에 해당하는 부분의 Rendering은 수행할 필요가 없고, 광선을 추적해서 그 레벨에서의 다음 셀로 건너뛰게 된다. 이 때 다음 셀의 부모 노드가 현재 셀의 부모 노드와 같지 않다면, 레벨을 하나 감소시킨 후에 위의 과정을 반복한다. 이렇게 레벨을 감소시키게 되면, 셀에 해당하는 데이터 블록의 크기는 2 배만큼 커지게 되고, 이 때 새로이 감소된 레벨의 셀에서의 최대/최소 범위가 Rendering하고자 하는 범위를 벗어날 경우, 보다 큰 범위의 셀 블록을 한 번에 건너 뛸 수 있게 되어 보다 많은 시간 감소를 가져올 수 있다. 반면, 셀의 밀도값의 최대/최소 값이 Rendering하고자 하는 범위를 만족시키게 되면 계속해서 레벨을 증가시키게 되는데, 레벨이 정의한 최대 레벨까지 내려가면 Rendering을 하게 된다. 이것에 대한 Algorithm을 간략하게 나타내면 다음과 같다.

```
TraceRay(u)
{
  Initialize color and opacity to 0;
  Initialize level to 0;
  // 데이터가 최대/최소값 이내이며
  // 불투명도가 경계값 이내일때
  while ( InBound(x) and (u) <= 1 - )
  {
    i = index(m,x);
    // 만약, 셀이 범위 안에 있으면, 레벨을 증가
    if ( Bound(i) and level < max_level )
      level ++;
    else {
      if (Bound(i))
```

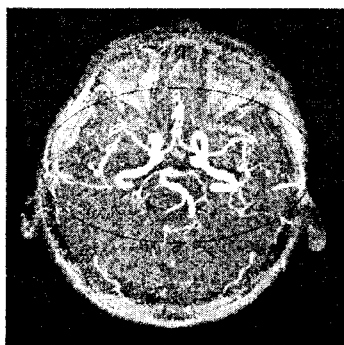
```

RenderCell(u, x);
// 다음 셀로 진행,
//상위 레벨로 레벨감소.
while ( Parent(level, Index(level,
Next(level, X, u))) != Parent(level, i)
and level > 0 )
{
i = Parent(level, i);
level --;
}
// 다음 셀로 진행
x = Next(level, X, u);
}
}
} // end TraceRay
// u : Image Space Coordinates
// x : Object space Coordinates
    
```



[그림 2] Octree를 이용하여 Ray Casting한 후의 MR Angio영상

이 방법을 사용하면 Octree를 저장하기 위한 여분의 메모리 공간을 사용하게 되는데, 이 공간은 Volume 데이터의 양에 비하면 적은 양이다. 따라서 이 방법은 Volume 데이터의 크기에 비해 여분의 메모리를 많이 사용하지 않고 시간을 효율적으로 줄일 수 있는 방법이다. 이 방법으로 사용하여 Ray Casting한 영상은 [그림 2]와 같다.



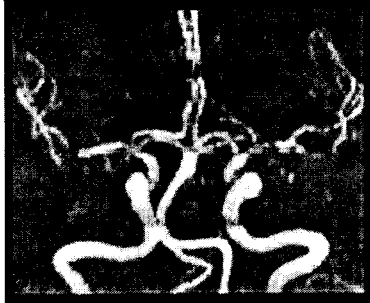
구현 및 실험

본 연구에서는 MR Angio영상을 사용하여 관심있는 영역을 중심으로 가시화하여 제안한 Algorithm으로 타당성을 실험하였다. 제안한 Algorithm의 몇가지 파라미터들을 수정함으로써 속도를 향상시킬 수 있었다. 기존의 Algorithm과 제안한 Algorithm의 실행속도는 [표 1]과 같다.

Data	Resolution	Ray Casting		Speed Up(배)
		Ray Casting (초)	with oct-ee(초)	
Brain	256X256X167	6.92	1.65	4.20
Head	256X256X225	14.49	3.28	4.41
Head	256X256X225	21.82	5.19	4.21

[표 1] Rendering 실험결과

실험결과, 처리시간이 평균 4배 이상 빨라졌다. 더 정확한 Rendering을 위해서는 단순히 Threshold에 의한 Segmentation이 아닌 혈관을 따라가면서 뼈대를 생성하는 알고리즘 같은 것이 적용될 수 있을 것이다. 또한, 속도가 크게 문제되지 않는다면 Gradient를 이용해 외부광원에 의한 효과를 주는 것도 좋을 것이다.



[그림 3] Clipping한 Octree에 의한 결과 이미지

결 론

본 연구는 PC에서 구현 가능하고 Rendering시간도 줄일 수 있는 알고리즘을 구현하였다.

Ray Casting과 Octree 알고리즘을 기본 알고리즘으로 Surface를 검출하여 실제 필요한 부분만을 Rendering함으로써 보다 효과적인 Rendering을 수행하도록 하였다.

앞으로 Volume Rendering Algorithm의 가속화를 위해 연구되어야 할 과제는 Software적으로 Algorithm을 가속화하는 한편, Hardware를 이용하여 가속화하는 방법을 도입하면 속도 향상은 많을 것이다. 이는 의료영상 전반에 걸쳐 적용할 수 있는 여러 Algorithm을 개발하는 것과 함께 연구되어야 할 것이다. 실제 임상에 효과적으로 이용되기 위해서는 실시간 Rendering이 필요하다. 따라서, 앞으로의 과제는 병렬 프로세싱 Rendering과 하드웨어를 이용하여 일반 PC에서도 구현 가능한 Rendering 알고리즘을 개발하여야 할 것이다.

참고문헌

11. Levoy, M., "Display of surfaces from volume 데이터," IEEE Computer Graphics and Applications, Vol.8, No.5, pp.29-37, May 1988.
2. Levoy, M., "Volume rendering by adaptive refinement," The Visual Computer, Vol.6, No.1, pp.2-7, May 1990.
3. Levoy, M., "Efficient ray tracing of volume 데이터," ACM Transactions on Graphics, Vol.9, No.3, pp.245-261, May 1990.
4. 이래경, 임인성. 2차원 샘플링에 기반을 둔 Ray Casting 방법의 속도향상 기법, 한국 컴퓨터 그래픽스 학회 학술대회, 1999년 8월.
55. A. Van Gelder and K. Kim, "Direct Volume Rendering with Shading via Three-Dimensional Textures", In Symposium on Volume Visualization, pages 23-30, San Francisco, CA, October 1996, ACM.