

최소 거리값을 이용한 효율적인 공간 도약법

채유식⁰ 신병석

인하대학교 전자계산공학과

g2011143@inhavision.inha.ac.kr⁰, bsshin@inha.ac.kr

An Efficient Space Leaping Using Minimum Distance Value

Yu-Sik Chae⁰ Byeong-Seok Shin

Dept. of Computer Science & Engineering, Inha University

요 약

효과적인 볼륨렌더링을 위해서는 빠른 속도와 고품질이라는 두 가지 조건을 모두 충족시켜야 한다. 기존에 사용된 광선추적법은 화질이 좋은 반면에 실시간 렌더링을 하기에는 한계가 있었다. 공간 도약법은 빈 공간에 대한 샘플링을 생략함으로써 렌더링 속도를 향상시킬 수 있으나 관측조건에 따라 성능이 저하되는 문제가 있다. 본 논문에서는 기존의 공간 도약법을 개선하여 관측조건과 무관하게 고품질을 유지하면서 속도를 향상시키는 방법을 제안한다. 여기서는 뷰 평면을 균일한 격자로 분할하고 각 격자내에서 가장 가까운 객체의 표면까지의 거리를 구한 후 그 위치부터 공간 도약함으로써 속도를 향상시킨다.

1. 서 론

볼륨렌더링(Volume Rendering)은 컴퓨터 그래픽스 분야에서 3차원 렌더링 기법의 하나로 보통 의료영상과 같이 대용량 데이터를 처리하는 분야에서 주로 사용된다. 볼륨 렌더링 방법은 화상순서(image-order) 알고리즘과 객체순서(object-order) 알고리즘으로 분류할 수 있다. 대표적인 화상순서 알고리즘인 광선추적법은 고품질의 영상을 얻을 수 있으나 대용량의 데이터를 처리할 때 시간이 많이 걸린다는 단점을 가지고 있다. 이를 보완하기 위하여 많은 가속화 기법들이 나왔는데 템플릿 기반 알고리즘[1], 광선 조기 종료법[2], 쉬어-웍 (shear-warp) 렌더링[3], 거리 변환(distance transform) 방법[4] 등이 있다. 템플릿 기반 알고리즘은 렌더링 속도를 크게 가속화시켰으나 원근 투영에는 사용할 수 없다는 단점을 가지고 있다. 쉬어-웍 방법은 삼선형 보간법 대신에 쌍선형 보간법을 사용함으로써 처리속도 면에서 큰 효과를 거둘 수 있었으나 쌍선형 보간을 한다는 점에서 화질이 다소 떨어지는 것을 볼 수 있으며 가상내시경과 같은 원근 투영 시에는 영상에 심한 왜곡이 생길 수 있다는 단점이 있다. 거리 변환 방법은 광선추적법처럼 고품질을 유지하면서 속도를 향상시킬 수 있다는 장점을 가지고 있다. 이 방법은 전처리 과정에서 각각의 복셀들과 가시화하려는 경계 복셀(boundary voxel)까지의 최단거리를 계산하여 거리맵이라는 공간 자료구조에 저장하고 실제 렌더링 중에는 이 정보를 이용하여 투명한 복셀들을 비약하도록 함으로써 렌더링 속도를 향상시킨다. 그러나 좁은 공간에서 렌더링하는 경우나 객체 표면이 관측 방향과 평행한 경우처럼 관측조건에 따라서 성능이 저하되는 문제점을 가지고 있다.

본 논문에서는 화상순서 알고리즘을 바탕으로 광선 조기 종료 기법과 거리 변환 기법을 사용하면서 렌더링 속

도를 가속화 시키기 위한 공간 도약법을 제안한다.

여기서 사용하는 최소 거리는 광선의 시작점에서 객체 표면과 만나지 않고 한번에 도약 할 수 있는 최대의 거리이다. 화면을 일정 간격의 격자로 나누어 각각의 격자 내에 네 개의 모서리에서 광선을 쏘아 객체의 표면과 만나는 점까지의 거리를 구하고 그 값들을 비교하여 최소 값을 구한후 격자별로 저장한다. 각 격자 내에서 네 개의 모서리의 픽셀을 제외한 나머지 픽셀에서 발사되는 광선들은 이 최소 거리값 만큼 도약 후 그 점에서부터 광선 추적을 시작한다.

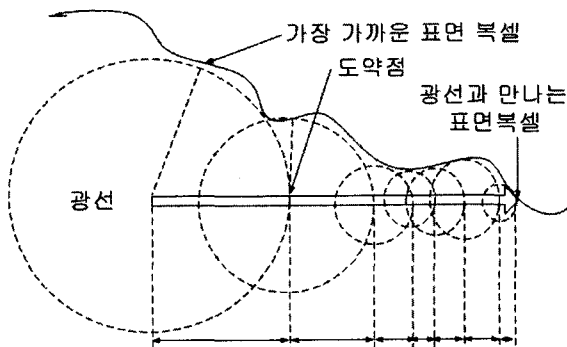
2절에서는 최소 거리를 구하는 방법과 이를 이용하여 렌더링 속도를 가속화시키는 방법에 대해서 살펴보고, 3절에서는 격자의 간격을 조절해 가면서 렌더링 속도를 측정하여 기존의 광선추적법에 광선 조기 종료 기법과 거리 변환 기법만을 적용한 경우와 비교 분석한 결과를 보여준다.

2. 최소 거리를 이용한 볼륨렌더링 가속화 기법

본 논문에서는 기존의 거리맵을 이용한 공간 도약법에서 속도 저하의 원인을 살펴보고 그 문제점을 보완함으로써 렌더링 속도를 향상시킬 수 있는 방법을 제시한다.

2.1 거리맵을 사용한 광선추적법의 문제점

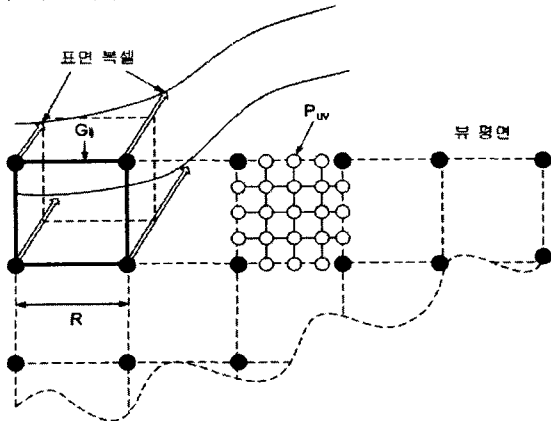
광선추적법에 거리맵을 적용하면 빈 복셀들을 도약함으로써 상당한 렌더링 시간 단축이 가능하다. 그러나 아래의 그림1에서 보는 바와 같이 관측점이 객체 표면에 근접하고 관측방향이 객체 표면의 경사와 평행한 경우 한번에 객체의 표면까지 도약할 수 없고 여러번에 걸쳐서 도약해야 한다. 이것은 거리맵 상의 각 거리값이 가장 가까운 표면 복셀로부터의 거리를 나타내기 때문이다.



[그림 1] 거리맵을 이용한 공간 도약법의 속도저하 문제

2.2 격자별 최소거리 계산

이 절에서는 이 논문에서 제안하고 있는 격자 별 최소거리 계산 방법을 설명한다.



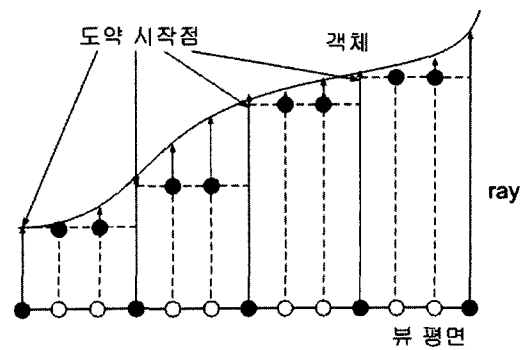
[그림 2] 격자 내 최소 거리 계산

광선추적법은 화상순서 알고리즘으로 뷰 평면의 시작 픽셀로부터 차례로 광선을 발사한다. 본 논문에서 제안하는 방법은 광선 추적을 시작하기 전에 화면을 일정 간격의 격자로 나누고 각각의 격자에서 네 개의 모서리에서만 광선을 쏘아 객체까지의 거리를 구하도록 한다. 구해진 네 개의 거리 값을 비교하여 최소 값을 구하게 되고 각 구간별로 구한 최소값을 식(1)과 같이 격자 개수만큼 할당된 자료구조 $dmin_{ij}$ 에 저장한다. 화상의 해상도가 N^2 , 격자의 크기가 R^2 , 그리고 픽셀 P_{uv} 에서의 거리가 d_{uv} 인 경우 격자 G_{ij} 에서의 최소거리 $dmin_{ij}$ 는 다음과 같다.

$$dmin_{ij} = \min\{d_{i^*R, j^*R}, d_{(i+1)^*R, j^*R}, d_{i^*R, (j+1)^*R}, d_{(i+1)^*R, (j+1)^*R}\} \quad (1)$$

2.3 최소거리를 이용한 공간 도약

각 격자 별로 최소거리를 계산한 후에는 광선추적을 수행하게 된다.

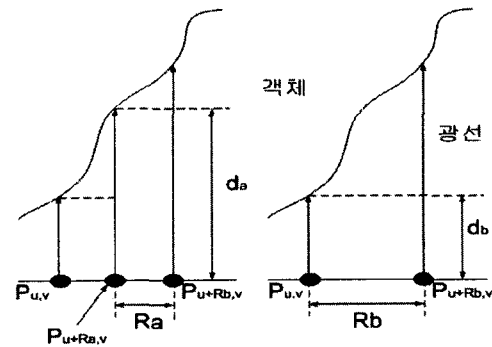


[그림 3] 최소거리를 이용한 공간 도약법

그림 3은 최소거리를 적용한 공간도약법을 보여주고 있다. 화면상에서 검은색 점은 미리 객체 표면까지의 거리가 계산된 픽셀을 나타내고 흰색 점은 미리 계산된 최소거리를 적용하는 픽셀을 나타낸다. 흰색 점에서 발사된 광선은 뷰 평면부터 공간 도약을 시작하지 않고, 최소거리 만큼 도약 후에 광선추적을 시작한다. 즉, 회색점 위치부터 객체 표면까지 기존의 거리맵을 참조하여 추가 도약을 하는 것이다. 따라서 최소거리만큼 가는 데 필요한 공간 도약 회수를 줄임으로써 렌더링 시간을 단축시킬 수 있다.

2.4 격자 크기와 성능과의 관계

본 논문에서 제안한 방법에서는 격자의 크기가 출력 영상의 화질과 렌더링 속도를 좌우한다. 격자의 크기가 클수록 최소거리 계산 시간이 단축되지만 데이터의 공간 주파수가 높을 경우 영상이 왜곡될 수 있다. 또한 객체 표면의 경사도와 광선의 방향이 평행에 가까울 경우 격자 내의 모서리 픽셀에서의 거리값 사이에 차이가 커져서 최소거리 계산에 의한 이득이 줄어들어 오히려 속도저하를 가져올 수 있다.



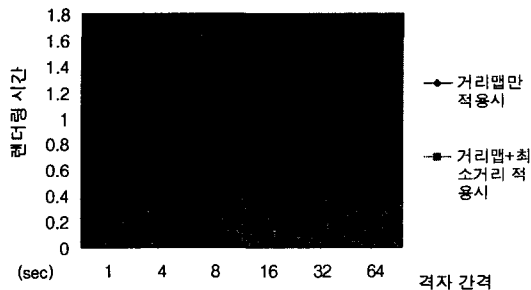
[그림 4] 격자 간격에 따른 최소거리의 차이

그림 4에서 볼 수 있듯이 격자의 크기가 R_b 일 경우 $P_{u,v}$ 와 $P_{u+R_b,v}$ 사이의 픽셀로부터 발사된 광선들은 d_b 만큼 도약후 추적을 시작하게 된다. 격자의 크기가 R_a 일 경우 $P_{u,v}$ 와 $P_{u+R_a,v}$ 사이의 픽셀에서는 d_b 만큼 도약후 추적을 시작하지만 $P_{u+R_a,v}$ 와 $P_{u+R_b,v}$ 사이의 픽셀에서는 d_a

만큼 도약후 추적을 시작한다. 따라서 어떤 경우가 더 유리한지 최소거리 값의 산출 시간과 광선의 추적회수에 따른 계산 시간 사이에서 최적점을 찾는 것이 중요하다.

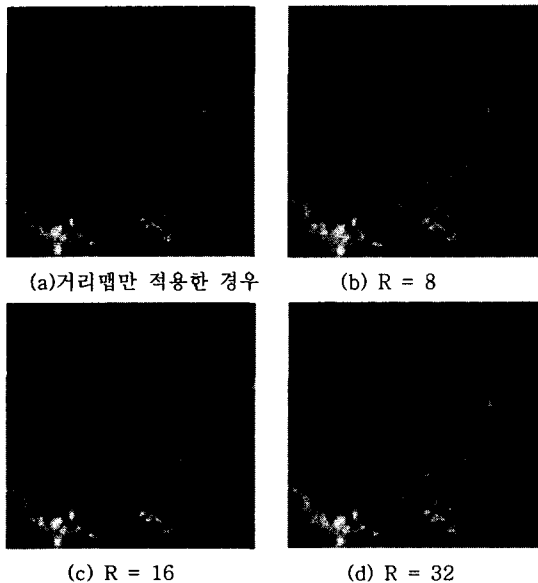
3. 실험결과

여기서는 거리맵을 이용하여 공간 도약했을 경우의 렌더링 시간과 본 논문에서 제안하는 알고리즘을 적용하였을 경우의 렌더링 시간을 측정하여 비교 분석한다. 실험은 펜티엄IV 2.2GHz CPU와 1GB의 메모리를 장착한 PC에서 이루어졌고 개발툴로는 Visual C++을 사용하였다. 실험 데이터는 대장 내시경을 위해 환자의 복부를 512×512×541 해상도로 촬영한 데이터를 사용하였다.



[그림 5] 격자 간격에 따른 렌더링 속도

동일한 조건에서 시간을 측정하기 위하여 동일한 관측점과 관측방향을 사용하여 렌더링하였다. 그림 5에서 볼 수 있듯이 최소거리를 이용하였을 경우 거리맵만을 이용하였을 경우보다 렌더링 시간이 단축되었음을 볼 수 있다. 격자 간격과 속도와의 관계를 보면



[그림 6] 격자간격에 따른 렌더링 영상

격자 간격이 8일 경우 렌더링 시간이 53% 단축되어 가속화가 가장 많이 이루어지는 것을 볼 수 있다. 반면에 격자의 간격이 8보다 커지거나 작아질 경우 가속량이 감소하는 것을 볼 수 있다.

그림 6 (a)는 동일한 위치에서 거리맵만 적용했을 경우의 영상이고 그림 6 (b)~(d)는 격자 간격을 8~32 사이에서 바꿔 가면서 최소거리를 적용하여 렌더링 한 영상을 보여주고 있다. 그림에서 볼 수 있듯이 격자 간격이 8, 16인 경우(b),(c) (a)의 영상과 큰 차이가 없는 것을 볼 수 있다. 그러나 격자의 간격이 32인 (d)의 경우 극소적으로 영상의 왜곡 현상이 발생하는 것을 볼 수 있다.

4. 결론

일반적으로 화질과 렌더링 속도는 서로 상충관계에 있기 때문에 이 두가지를 모두 만족시키기는 쉽지 않다. 볼륨 렌더링이 주로 사용되는 의료영상과 같은 분야에서는 정확한 영상과 빠른 처리 속도 모두를 필요로 한다. 본 논문에서는 볼륨렌더링에서 공간 도약법의 하나인 거리변환 방법의 문제점을 보완하여 우수한 화질을 유지하면서도 렌더링 속도를 향상시킨 방법을 제시하였다. 실험 결과를 통해서 알 수 있듯이 격자의 간격에 따라서 거리맵만을 사용하였을 경우와 비교하여 대등한 화질을 유지하면서도 렌더링 속도는 크게 향상된 것을 볼 수 있다.

참고문헌

- [1] Yagel, R and Kaufman, A., "Template-based volume viewing," Computer Graphics Forum (Eurographics 92 Proceedings), pp. 153-167, Cambridge, UK, 1992.
- [2] Levoy, M., Efficient ray tracing of volume data. ACM Transactions on Graphics, pages 245-261, July 1990.
- [3] Lacroute P. and Levoy, M., "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation," Computer Graphics (SIGGRAPH 94 Proceedings), Orlando, Florida, pp. 451-458, 1994.
- [4] Zuiderveld, K., Koning, A. H. J., Viergever, M. A., "Acceleration of Ray Casting Using 3D Distance Transforms", Proceedings of Visualization in Biomedical Computing 1992, 1808, 324-335, October 1992.
- [5] Levoy, M., "Display of Surfaces from Volume Data," IEEE Computer Graphics and Applications, Vol. 8, No. 3, pp.29-37, 1988.
- [6] Danksin, J., and Hanrahan, P., "Fast algorithm for volume raycasting," Workshop on Volume Visualization, pp. 91-98, 1992.
- [7] Yage, R. and Shi, Z., "Accelerating volume animation by space-leaping," Proceedings of IEEE Visualization '93, pp. 62-69. 1993.