J Biomed. Eng Res.
Vol. 21, No 4, 433-439, 2000

쉬어-왑 분해를 이용한 블록 기반의 볼륨 렌더링 기법

권성민 · 김진국 · 박현욱 · 나종범

한국과학기술원 전자전산학과 전기 및 전자공학 전공

(2000년 2월 9일 접수, 2000년 6월 14일 채택)

A Block-Based Volume Rendering Algorithm Using Shear-Warp Factorization

S.M. Kwon, J.K. Kim, H.W. Park, and J.B. Ra

Div. of Electrical Engineering, Dept. of Electrical Engineering & Computer Science
Korea Advanced Institute of Science and Technology (KAIST)

(Received February 9, 2000, Accepted June 14, 2000)

요 약 볼륨 렌더링은 기하학적인 기본 도형으로 모델링하지 않고, 3차원 데이터를 직접 가시화 하는 방법이다. 이런 볼륨 렌더링의 특성으로 말미암아 3차원 영상을 도시할 때에, 복잡한 물체의 경우에도 물체의 표면을 상세하게 표현하는 데 유리하여 의료 영상을 가시화 하는 쪽으로의 적용이 많이 이루어져 왔다. 일반적으로 볼륨 데이터의 크기가 커서 실시간으로 처리하기 쉽지 않기 때문에, 근래에는 이 렌더링 시간을 줄이기 위해서 많은 여러 가지 렌더링 알고리즘이 제안되었다. 하지만, 제안되었던 대부분의 알고리즘들은 부호화되어 있지 않은 볼륨 데이터를 빠르게 렌더링 하기에 적합하지 않다. 본 논문에서는 부호화 되어 있지 않은 볼륨 데이터를 빠르게 렌더링 하기 위해서, 쉬어-왑 분해를 이용하는 블록 기반의 볼륨 렌더링 기법을 제안한다. 이 방법에서는 블록 기반의 데이터와 함께 장기의 영역 분할 데이터를 동시에 이용하여 볼륨 렌더링을 수행하므로써, 부호화되어 있지 않은 데이터에 대해 렌더링 속도를 증가시킨다. 본 논문에서는 3차원 X-ray CT 흉부 영상과 MR 3차원 두부 영상을 렌더링 함으로써 제안한 방법의 성능을 검증하였다.

Abstract : Volume rendering is a powerful tool for visualizing sampled scalar values from 3D data without modeling geometric primitives to the data. The volume rendering can describe the surface-detail of a complex object. Owing to this characteristic, volume rendering has been used to visualize medical data. The size of volume data is usually too big to handle in real time. Recently, various volume rendering algorithms have been proposed in order to reduce the rendering time. However, most of the proposed algorithms are not proper for fast rendering of large non-coded volume data. In this paper we propose a block-based fast volume rendering algorithm using a shear-warp factorization for non-coded volume data. The algorithm performs volume rendering by using the organ segmentation data as well as the block-based 3D volume data, and increases the rendering speed for large non-coded volume data. The proposed algorithm is evaluated by rendering 3D X-ray CT body images and MR head images.

Key words : Volume rendering, Shear-Warp factorization, Shearing, Warping, Block, Segmentation

서 론

전자공학의 발전과 함께 20세기 의료기기의 혁명이라 할 수 있는 CT(computerized tomography), MR(magnetic resonance), 초음파 촬영장치 등이 등장하였다. 이러한 기기 들을 이용하면 인체 내부를 직접 열어보지 않고도, 단면 영상들로 이루어진 3차원 볼륨 데이터 정보를 얻을 수 있는 장점 때문에 그 활용 범위가 넓어지고 있다. 더불어 3차원 볼륨 데이터를 3차원으로 도시화하기 위한 알고리즘들이 활발히 연구되어 왔다[1-2].

3차원 볼륨 데이터를 도시화하기 위한 방법으로는 크게 표면 렌더링(surface rendering)과 볼륨 렌더링(volume rendering)으로 나눌 수 있다. 표면 렌더링 방법은 동일 표면 추출

본 연구는 정보 통신부 국제 공동 연구 지원 사업(과제번호 98-10)의 지원 하에 수행되었음

통신저자 : 나종범, (305-701) 대전광역시 유성구 구성동 373-1
한국과학기술원 전자전산학과
Tel. (042)869-5434, Fax (042)869-8360
E-mail. jbra@ee.kaist.ac.kr

(iso-surface extraction) 과정을 거쳐 삼각형 등의 기하학적인 기본도형으로 모델링 한 후, 컴퓨터 하드웨어의 도움을 받아 도시화하는 방법이다[3]. 이 방법은 하드웨어의 도움으로 인해 실시간 도시화는 가능하지만, 삼각형으로 모델링하기 위한 전처리 시간이 많이 걸린다는 단점이 있다. 그리고, 복잡하거나 부피가 작은 장기의 경우에는 표면 모델링(surface modeling)이 부적합하여 원래 형태를 상실하거나 왜곡이 되는 단점이 있다

볼륨 렌더링은 3차원 볼륨 데이터를 직접 도시화하는 방법이다[4-7]. 이 방법은 표면 모델로의 변환 과정이 없기 때문에 복잡하거나 작은 장기에 대해서도 효과적으로 3차원으로 도시할 수 있고, 전처리 과정이 비교적 단순하다는 장점이 있다 하지만, 일반적으로 볼륨 데이터는 그 양이 방대하기 때문에 렌더링 시간이 많이 걸린다 이를 해결하기 위해 3차원 데이터의 전처리 과정에서 볼륨 데이터를 알맞게 부호화(encoding)함으로써, 렌더링에 참여하는 데이터의 수를 줄여 렌더링 속도를 향상시키는 방법들이 연구되어 왔다 이중 한가지 방법으로 런 길이 부호화(the run-length encoded volume)와 쉬어-왑 분해(shear-warp factorization)를 이용한 방법이 있다[7-8]. 이 방법에서는 전처리 과정에서 불투명도를 이용하여, 렌더링 될 데이터를 미리 결정해서 런-길이 부호화를 해 두기 때문에, 특정 하드웨어의 도움 없이 빠른 3차원 가시화가 가능하다. 반면에, 렌더링 될 장기의 불투명도의 변경과 같은 렌더링 변수들을 쉽게 변화시킬 수 없다는 단점이 있다. 최근에 활발히 연구되고 있는 사전 수술 계획 시스템이나 모의 실험기 등의 개발에는 이러한 렌더링 변수들을 대화식 수준에서 변경 하는 기능이 필수적이다

본 논문에서는 렌더링 변수를 대화식 수준으로 변경하기 위해 볼륨데이터를 부호화(encoding)하지 않은 상태로 렌더링 하면서도, 기존 방법과 비슷한 렌더링 속도를 유지하는 볼륨 렌더링 기법을 제시하려고 한다 이를 위해 쉬어-왑 분해를 이용한 볼륨 렌더링 방법에 블록의 개념을 도입하였다. II 장에서는 쉬어-왑 분해 알고리즘을 간단히 살펴본다 III 장에서는 제안한 개념을 구체화 하고, 제안한 개념을 효과적으로 운용하기 위한 방법 중의 하나로, 영역 분할 데이터(segmentation data)를 이용하여 여러 개의 장기들을 동시에 렌더링 하는 방법을 제시한다 IV 장에서는 흉부의 CT 데이터와 두부의 MR 데이터를 이용하여 제안한 알고리즘을 검증한다.

쉬어-왑 분해를 이용한 볼륨 렌더링 방법

볼륨 데이터의 단위 원소를 복셀(voxel), 볼륨 데이터가 투영되는 평면을 투영 평면(projected image plane)이라고 하고, 투영 평면의 한 픽셀(pixel)로부터 볼륨 데이터가 존재하는 3차원 공간으로 나가는 직선들을 시선이라고 하자. 그림 1은 일반적인 레이 캐스팅(ray-casting) 방법을 2차원으로 단순화한 그림이다. 그림 1에서 볼 수 있듯이 렌더링 된 영상의 픽셀 값을 얻기 위해서는 시선을 따라가며 일정한 간격으로 샘플링하고, 얻어진 샘플 값을 합성해야 한다 이때 복셀들의 위치와 투영 평면으로부터 나오는 시선의 샘플링 점들이 서로 일치하지 않기 때문에, 보간(interpolation)을 통해 샘플링 점들의 값을 구해야 한다.

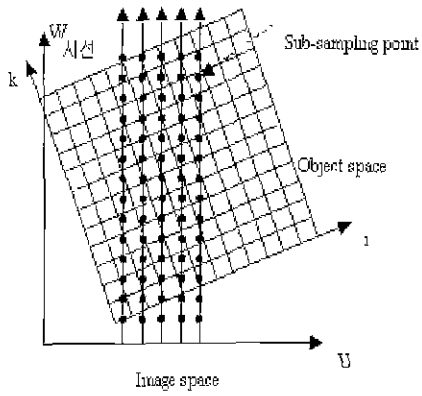


그림 1. 일반적인 레이 캐스팅 방법의 2차원 개략도
Fig. 1. Simplified 2D diagram of the conventional ray-casting method

그림 1에서 U축은 투영 평면을 1차원으로 나타낸 것이고, 정사각형 격자들은 볼륨 데이터를 2차원으로 나타낸 것이다. 또한, 정사각형 격자의 교점들은 복셀들을 의미하고, U축에 수직으로 볼륨데이터로 뻗어 나가는 직선들이 시선을 나타낸다. 그림 1에서는 시선 위의 샘플링 점들이 복셀들이 놓인 격자에 불규칙하게 놓이는 것을 확인 할 수 있다. 이 경우 샘플 값의 보간에 필요한 주변 복셀들의 가중치를 계산해야 하므로, 계산량이 많아진다.

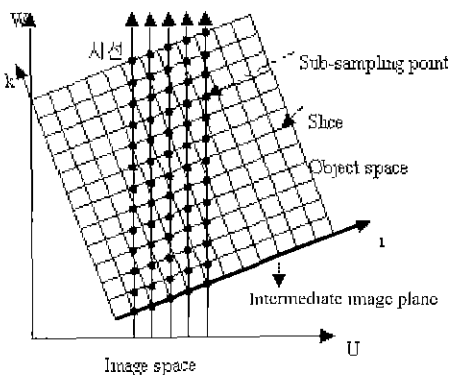


그림 2. 일반적인 쉬어-왑 분해 방법의 2차원 개략도
Fig. 2. Simplified 2D diagram of the conventional shear-warp factorization

반면에 쉬어-왑 분해 방법은 그림 2와 같이 중간 투영 평면(intermediate image plane)을 두고, 중간 투영 평면에 볼륨 데이터를 투영하게 된다 이 경우 중간 투영 영상으로부터 같은 거리만큼 떨어진 샘플링 점들은 격자에 규칙적으로 놓이므

로, 보간에 필요한 가중치를 한번만 구하면 된다. 이 과정을 쉬어(shear)라고 하고, 보간에 필요한 가중치 계산을 줄임으로써 렌더링 속도를 향상시킨다. 이때, 중간 투영 평면에 얻어진 영상은 실제 투영 평면의 영상과 달리 왜곡되어 있으므로, 최종 결과 영상을 얻기 위해서 와핑(warping)을 해 주어야 한다. 이와 같이 쉬어-왑 분해 방법은 쉬어와 와핑의 두 과정으로 나누어 처리함으로써 계산량을 줄이는 방법이다.

이 쉬어-왑 분해 방법을 이용하여 3차원 볼륨 데이터를 렌더링 하는 대표적인 방법은 Lacroute가 제안한 방법으로, 이 방법에서는 볼륨 데이터의 전처리 과정에서 미리 도시 될 복셀들을 결정하여 런 길이 부호화(the run-length encoded volume)를 하고, 그 다음 렌더링 단계에서는 부호화(encoding) 된 볼륨 데이터를 쉬어-왑 분해 방법을 이용하여 도시하게 된다[7]. 이 방법은 전처리 과정에서 렌더링 되는 볼륨 데이터만을 부호화(encoding) 함으로써 데이터 양을 줄이고, 계산량을 줄이는 장점이 있다. 하지만, 보고자 하는 장기를 바꾸거나, 장기의 불투명도 등의 렌더링 변수를 변경하는 경우에는 다시 런 길이 부호화를 해야 하므로, 사전 수술 계획 시스템이나 모의 실험기처럼 렌더링 변수를 대화식 수준으로 변경하는 기능이 필요한 경우에는 부적합 하다는 단점이 있다

제안한 볼륨 렌더링 알고리즘

1. 필요성

3차원 렌더링 기법을 사용하는 의공학 쪽의 애플리케이션(application)에는 사전 수술 계획 시스템이나 모의 실험기 등이 있다. 이러한 시스템에서는 장기 상호간의 위치정보나 장기끼리의 포함관계 등의 3차원적인 공간 개념을 사용자에게 보다 직관적으로 제공하는데 그 목표가 있다. 이런 3차원 공간 개념을 제공하는 방법 중에 한가지는, 각각의 장기의 불투명도를 다르게 적용함으로써 사용자에게 공간 개념을 전달하는 방법이다. 앞에 있는 장기의 불투명도를 작게 하여 뒤에 있는 물체와의 상호 관계를 살펴 볼 수 있고, 겉을 둘러 싸고 있는 장기의 불투명도를 작게 하여 안에 있는 장기의 정보를 얻을 수 있다. 따라서 애플리케이션의 개발에는 불투명도를 대화식 수준으로 변경 하는 기능이 필요하다.

대화식 수준으로 장기의 불투명도를 조절하는 기능을 위해서는 볼륨 데이터를 런 길이 부호화(the run-length encoded volume)를 하지 않은 상태에서 직접 렌더링 해야 한다. 이 경우에, 전체 볼륨에 대해서 영상 분할 정보를 이용하면 장기의 불투명도를 임의로 조절 할 수 있지만, 도시 되지 않아도 되는 복셀을 결정하는 작업이 전처리 과정이 아닌 렌더링 과정에서 이루어지므로 렌더링 시간이 증가한다는 단점이 있다[4]. 또, 도시 되지 않는 복셀들의 위치 정보를 옥트리(octree) 구조를 이용하여 미리 저장하는 방법의 경우에는 렌더링 변수가 바뀔 때 다시 옥트리를 갱신해야 하므로, 렌더링 변수 변경에 시간이 많이 걸린다는 단점이 있다[9]. 또한, 위 두 방법 모두 전체 볼륨 데이터를 직접 렌더링에 사용하므로 메모리의 일관성이 깨지기 때문에 보는 방향에 따라 렌더링 속도가 크게 달라진다는 단점이 있다[8] 이 문제를 해결하기 위해, 본 논문에서는 쉬어-왑 분해를 이용하는 블록 기반의 볼륨 렌더링 기법을 제안한다.

2. 문제 해결 방법

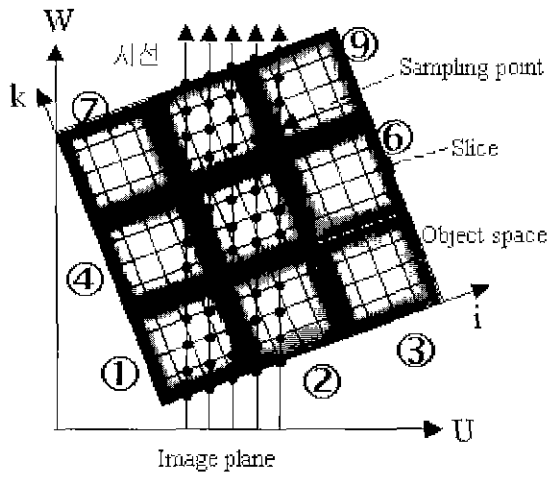


그림 3. 제안한 방법의 2차원 개략도
Fig. 3. Simplified 2D diagram of the proposed algorithm

위의 메모리 일관성 문제를 해결하기 위해, 본 논문에서는 볼륨 데이터를 블록 단위로 나누어 저장하고, 렌더링 하는 방법을 제안한다. 그림 3은 제안하는 볼륨 렌더링 방법을 2차원으로 단순화한 그림이다. 그림 2와 같이 쉬어-왑 분해 방법을 사용하였지만, 볼륨 데이터를 블록 단위로 나누고, 각 블록 별로 중간 투영 평면에 투영시킨다는 점이 이 알고리즘의 특징이다. 이 경우 볼륨을 바라보는 방향에 따라 각 블록들의 렌더링 순서를 결정해야 한다. 각 블록의 처리 순서는 투영 평면에 가까운 순서대로 정해진다. 그림 3에서의 원 안의 숫자는 각 블록이 렌더링 되는 순서를 예를 들어 표시하였다. 투영 후에 만들어 지는 한 장의 중간 투영 평면을 기존과 같은 방법으로 와핑(warping)하여, 최후 렌더링 영상을 얻는다

3차원 배열은 일반적으로 1차원 배열의 메모리에 저장되므로, 3차원에서는 인접한 복셀이지만 방향에 따라 실제 메모리 상에서는 멀리 떨어져 있을 수 있다. 그리고, 블록의 크기가 커질수록 인접한 복셀 간의 간격이 실제 메모리 상에서는 더 커지게 된다. 이런 이유로, 블록의 크기가 큰 경우에는 보는 방향에 따라 캐시 미스(cache miss) 발생 빈도가 크게 달라지게 된다[8]. 컴퓨터의 구조상 메모리에서 데이터를 읽어오는 경우에는 외부 메모리의 속도를 극복하기 위해서, 그 데이터만 CPU로 읽어오지 않고 데이터 주변의 일정한 양의 데이터를 한번에 캐시 메모리로 읽어오게 된다. 이때, 읽고자 하는 데이터가 캐시 메모리에 없는 경우, 외부 메모리에서 다시 캐시 메모리로 정해진 양의 데이터를 읽어오게 된다. 이런 경우를 캐

시 미스라 하는데, 캐시 미스가 발생하면 속도가 느린 외부 메모리에서 계속 데이터를 많이 읽어와야 하므로, 렌더링 시간이 많이 걸리게 된다. 따라서, 캐시 미스를 모든 방향에 대해 줄이기 위해서는 블록 크기를 작게 해야 한다.

반면에 블록의 크기가 작을 때는 같은 크기의 볼륨을 나타내기 위해서는 볼륨을 이루는 블록의 개수가 많아지게 되는데, 이때 렌더링 될 블록들을 보는 각도에 따라 순서를 결정하고 순서대로 어드레싱 할 때에 걸리는 시간과 블록 헤더를 조사하는 데 걸리는 시간이 증가하게 된다. 또한 쉬어-왑의 가속화 기법으로 사용되는 중간 투영 평면 위에서의 런 길이 부호화(the run-length encoded image)는 중간 투영 평면의 불투명도를 이용하여 투영되지 않아도 되는 복셀들을 건너뛸 수 있는 구조를 제안하는데, 블록의 크기가 작아짐에 따라 건너뛰는 복셀의 개수도 작아져서 이 가속화 기법의 이점을 줄인다 [8]. 이 두 가지 상반된 조건을 고려하여 블록의 크기를 정해야 렌더링 시간을 줄일 수 있기 때문에, 블록의 크기를 실험을 통해 정하였다.

블록을 분할할 때는 블록 데이터에서 샘플링하기 쉽도록, 블록 주위 한 면의 복셀 슬라이스를 더 가지고 있어야 한다. 따라서, 블록의 한 변이 (2n+1)의 형태를 지닌다. 이 실험에서는 n은 2, 4, 8, 16, 32, 64의 6가지 경우에 대해 렌더링에 걸리는 CPU 클락 수를 측정하였다. 실험에는 Silicon Graphics사의 R10000 250MHz dual CPU, primary cache 32KByte, secondary cache 1MByte 와 256MByte RAM을 갖춘 Onyx2 컴퓨터를 이용하였다. 사용된 데이터는 플라스틱 두개골의 CT 영상이다.

표 1. 블록 크기에 따른 렌더링 클락 수 (전체 256×256×256 복셀)
Table 1. Block size versus the number of clocks for rendering total voxels of 256×256×256.

블록 크기	렌더링 클락 수
5×5×5	4402708
9×9×9	3709829
17×17×17	3551682
33×33×33	3565304
65×65×65	3589609
129×129×129	6516379

표 1은 256×256×256의 볼륨을 건너 뛰 없이 모든 복셀을 렌더링 했을 때, 렌더링에 소요되는 클락 수를 측정한 것이다. x축 방향으로 10°씩 36번 회전하고 y축 방향으로 10°씩 36번 회전하여 나온 클락 수를 평균한 것이다. 이 표에서는 블록의 크기가 클 때의 캐시 미스의 시간적인 영향과 블록의 크기가 작을 때의 부가적인 연산의 시간적인 영향을 CPU 클락 수를 통하여 살펴본다. 블록의 크기가 아주 크거나 작은 경우에 클락이 많이 소요됨을 관찰할 수 있다.

표 2. 블록 크기에 따른 렌더링 클락 수 (가속화 기법 적용)
Table 2. Block size versus the number of clocks for rendering based on several accelerating methods

블록크기	렌더링 클락 수	
	블록 건너뛰기	블록 건너뛰기 + 이른 시선 계산 끝내기 - 투영 평면상의 런 길이 부호화
5×5×5	285440	114337
9×9×9	267785	94930
17×17×17	315889	122593
33×33×33	438222	221376
65×65×65	679213	-
129×129×129	1745354	-

표 2는 제안한 방법에 렌더링 되지 않는 블록의 건너뛰기와 기존에 제안되었던 이른 시선 계산 끝내기(early-ray termination), 투영 평면상의 런 길이 부호화 등의 가속화 기법들을 적용하여 실험한 결과이다. 블록의 건너뛰기는 다음 3절에서 설명한다. 이 표에서는 블록의 크기에 따라, 앞에서 제안한 방법과 표 1에서 보인 영향 그리고 기존의 가속화 기법들이 함께, 렌더링 속도에 어떤 영향을 미치는지를 CPU 클락 수를 통하여 살펴본다. 256×256×256의 플라스틱 두개골의 CT 볼륨 데이터의 경우 9×9×9 크기의 블록을 사용하면 속도가 가장 빨라짐을 알 수 있었다. 플라스틱 두개골의 CT 볼륨 데이터 이외에, 512×512×338의 인체 흉부의 CT 볼륨 데이터를 이용하여 실험했을 경우에도 같은 결과가 나타났다.

3. 영역 분할 데이터를 이용한 블록 운용 방법

앞서 설명한 블록 기반의 렌더링 기법을 통하여 렌더링 할 대상을 미리 런 길이 부호화 (the run-length encoded volume) 하지 않고 물체의 불투명도를 대화식으로 조절하며, 3차원 영상을 생성 할 수 있다. 이 알고리즘을 이용해서 특정 장기를 선택하고 선택된 장기의 불투명도를 변환 시키기 위해서는 영역 분할 데이터와의 효과적인 융합 방법이 필요하다. 이 논문에서는 볼륨데이터를 블록으로 나누고, 블록의 볼륨 데이터와 함께 영상 분할 데이터도 함께 저장하는 방법을 사용한다. 각 블록은 헤더와 복셀 정보를 가지는데, 헤더는 장기의 개수에 따라 1~2 바이트이고, 헤더의 각 비트는 각 장기를 나타낸다. 따라서 한 블록 안에 포함되어 있는 장기의 종류에 따라 헤더의 비트들이 결정된다. 블록들은 헤더 이외에도 복셀 정보를 저장하는데, 이 각각의 복셀과 함께 해당 복셀의 영상 분할 데이터도 저장한다.

앞서 언급한 알고리즘을 통하여 구현한 렌더링 구조는 크게 세 단계로 구성되어 있다.

처음 단계는 전처리 과정이다. 이 과정에서는 장기를 포함하고 있는 블록들 만을 선택하여, 블록 안의 복셀 정보와 블록 헤더를 파일에 저장한다. 이렇게 만들어진 파일은 렌더링 될

가능성이 있는 모든 복셀들을 포함하고 있기 때문에 어떠한 경우에도 다시 전처리 과정을 수행 할 필요는 없다. 두 번째 단계는 각 블록을 차례대로 중간 투영 평면으로 투영하는 블록 투영 단계이다 렌더링 시 사용자가 선택하는 장기가 선택하는 불투명도로 투영이 된다. 이 과정에서 블록의 헤더를 이용하여 각 블록 안의 영상 분할 데이터를 전부 조사하지 않고도, 블록의 렌더링 여부를 결정할 수 있다 2절에서 설명한 블록 건너 뛰기는 이러한 블록 헤더를 이용한 방법을 명명한 것인데 이 방법을 통하여 렌더링 시간과 불투명도가 변환될 블록을 선택하는 시간을 단축 할 수 있다. 세 번째 단계는 중간 투영 평면을 이용하여 최종 결과 영상을 얻는 와핑 단계이다. 블록을 투영하여 얻은 중간 투영 평면은 블록기반이 아닌 방법으로 만들어진 중간 투영 평면과 동일하기 때문에, 기존의 와핑 방법 그대로 사용이 가능하다.

실험 결과

본 실험에서는 제안한 방법을 검증하기 위하여, 런-길이 부호화한 데이터를 쉬어-왑 분해를 이용하여 렌더링 하는 방법과 제안한 방법을 함께 비교하였다. 기존 방법과의 비교는 렌더링 시간과 렌더링 변수의 변경에 필요한 시간을 측정함으로써, 제안한 방법이 렌더링 시간에서 큰 손실 없이 렌더링 변수를 대화식 수준으로 변경할 수 있음을 보였다 사용된 컴퓨터는 앞 장에서 사용한 Onyx2를 사용하였고, 실험에 사용된 데이터는 256×256×256 크기의 플라스틱 두개골 CT 데이터를 이용하였다.

실험 결과는 표 3과 같다. 렌더링 시간은 보는 방향을 x축을 중심으로 10°씩 36번 회전시키고, y축을 중심으로 다시 10°씩 36번 회전시켰을 때 걸린 시간을 평균한 시간이다 표 3에서 알 수 있듯이 렌더링 시간은 기존 방법이 제안한 방법에 비해서 다소 빠르다. 하지만, 불투명도와 같이 렌더링 변수를 변화시키는 경우, 제안한 방법이 기존 방법에 비해 훨씬 빠르기 때문에 사전 수술 계획 시스템이나 모의 실험기 등에 적합하다는 것을 알 수 있다.

흉부 CT 영상을 렌더링 할 때 사용한 영상 분할 결과는 그림 4(b)와 같고, 자체적으로 개발한 영상 분할 프로그램을 이용하여 영상을 분할하였다[10] 그림 5는 영상 분할 정보를 이용하여, 흉부 CT 영상에 대해서 허파의 불투명도만 변화시키며 렌더링 한 영상들이다 그림에서 볼 수 있듯이 허파의 불투명하게 렌더링 한 경우에는 허파 내부의 구조물들을 자세하게 표현 할 수 있고, 허파를 투명하게 하면 허파와 주변 뼈 사이의 상호 위치 관계를 파악하기가 쉽다 그림 6은 256×256×256 두부 MR 볼륨 데이터를 렌더링 한 결과이다 그림 6(a)에서는 피부를 불투명하게 렌더링 하여 피부의 질감을 느낄 수 있고, 그림 6(b)에서는 피부를 약간 투명하게 렌더링 함으로써 피부와 뇌와의 위치관계를 쉽게 파악 할 수 있다

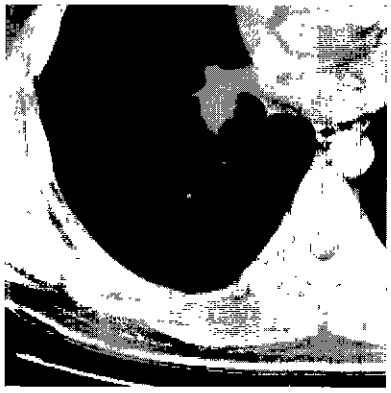
표 4는 렌더링 되는 장기들을 변화 시킨 경우에 대해 Onyx 2와 PC에서의 렌더링 속도를 측정하여 제시하였다. 사용된 데이터는 512×512×335 크기의 흉부 CT 볼륨 데이터 중 256×

표 3. 기존 방법과 제안한 알고리즘의 비교
Table 3 Comparison between the conventional and proposed algorithms

걸린 시간	기존 방법	제안한 방법
렌더링 변수 변경 (불투명도 변화의 경우)	51.84 sec	0.92 sec
렌더링 시간	96 msec	116 msec

표 4. 서로 다른 렌더링 변수에 대한 렌더링 속도들
Table 4. Rendering speeds for different rendering-parameters

선택된 장기	렌더링 속도 (Onyx2)	렌더링 속도 (PC)
뼈	5.0 f/s	3.9 f/s
뼈 + 양측 허파	2.3 f/s	1.8 f/s
뼈 + 양측 허파 + 피부	1.9 f/s	1.2 f/s



(a)

(b)

그림 4 (a) 흉부 CT 영상의 예 (b) 영역 분할 데이터의 예
Fig 4. (a) A CT image of the chest (b) It's segmentation result

(a) (b)

그림 5. CT 데이터의 렌더링 결과 (a) 불투명한 허파의 경우 (b) 약간 투명한 허파의 경우
Fig. 5 Rendering results of a CT data set with (a) opaque lung, and (b) translucent lung

(a) (b)

그림 6. MR 데이터의 렌더링 결과 (a) 불투명한 피부의 경우 (b) 약간 투명한 피부의 경우
Fig. 6. Rendering results of a MR data set with (a) opaque skin, and (b) translucent skin

(a) (a)

그림 7. (a) 전체 흉부 CT 데이터의 렌더링 결과 (b) 허파를 없앤 렌더링 결과
Fig. 7 Rendering results for (a) the total CT chest data set and (b) without lungs.

256×256 크기의 데이터만을 잘라내 이용하였고, 선택된 장기를 변경하는데 걸리는 시간은 표 3의 렌더링 변수 변경 시간

과 동일하다. 실험에는 PII Xeon 450MHz dual CPU, 256 MByte RAM을 갖는 PC를 사용하였다. 그림 7은 흉부 CT 볼륨 데이터 전체를 제안한 알고리즘을 이용하여 렌더링 한 결과이다. 그림 8(a)는 허파가 있을 때를 도시한 결과이고, 그림 8(b)는 허파를 없애고 도시한 결과이다

결 론

본 논문에서는 쉬어-왑 분해를 이용한 블록 기반의 볼륨 렌더링 방법을 제안하였다 사전 수술 계획 시스템이나 모의 실험기 등의 경우에는 도시하고자 하는 장기의 선택이나 불투명도의 변경 등, 렌더링 변수를 대화식 수준으로 바꿀 수 있어야 한다. 기존의 방법은 전처리 과정에서 볼륨 데이터를 부호화함으로써 빠른 렌더링 속도를 가졌지만, 렌더링 변수를 변경하는 경우 다시 전처리 과정을 수행해야 하는 단점이 있었다. 이런 단점을 극복하기 위해 제안한 알고리즘에서는 볼륨 데이터를 부호화하는 대신, 블록으로 나누어 저장하고 블록 단위로 쉬어-왑 분해 방법을 적용하였다 전처리 과정에서는 볼륨데이터를 블록으로 나누어 저장하고, 영역 분할 데이터를 이용하여 각 블록이 포함한 장기에 대한 정보를 함께 저장한다. 렌더링 과정에서는 사용자에 의해 선택된 블록을 효과적으로 찾아내어 블록 별로 렌더링을 수행함으로써 불필요한 계산을 막았다 이와 같은 방법을 통해 기존 방법에 비하여 렌더링 속도의 큰 손실 없이 대화식 수준으로 렌더링 변수를 변경할 수 있게 되었다. 흉부 CT 영상 데이터, 두부 MR 영상 데이터를 이용하여 제안한 알고리즘을 검증하였고, 대화식 수준의 렌더링 변수 변경이 가능함을 확인하였다. 본 논문에서는 블록의 크기가 일정한 경우만을 다뤘지만, 볼륨 데이터의 특성에 따라서 적응적으로 블록의 크기를 가변 하는 방법에 대한 연구가 필요하다.

참 고 문 헌

1 M Magnusson, R. Lenz, and P.E. Danielsson, "*Evaluation of methods for shaded surface display of CT-volumes*", 9th International Conference on Pattern Recognition, vol. 2, pp. 1287 1294, 1988

2. M Levoy, "*Display of surfaces from volume data,*" IEEE Computer Graphics and Applications, pp. 29-37, May 1988

3 W E Loresen, "*Marching cubes- A high resolution 3D surface construction algorithm,*" Computer Graphics, vol. 21, no 4, pp. 163-169, July 1987

4 R A Drebin, L. Carpenter, and P. Hanrahan, "*volume rendering,*" Computer Graphics, vol. 22, no 4, pp. 65-74, August 1988

5 M. Levov, "*Efficient ray tracing of volume data,*" ACM Transactions on Graphics, vol. 9, no. 3, pp. 245-261, July 1990

6 L.A Westover, "*Footprint Evaluation for Volume Rendering*", Computer Graphics, vol. 24, no 4, pp. 367-376, August 1990

7 P. Lacroute and M. Levov, "*Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation*", Proceedings of SIGGRAPH 94, pp. 451-458, Orlando, Florida, July 1994

8. P. Lacroute, "*Fast volume rendering using a shear-warp factorization of the viewing transformation,*" PhD dissertation, Stanford Univ., 1995

9 D. Meagher, "*Geometric modeling using octree encoding,*" Comput. Graph Image Process 19, pp. 129-147, 1982

10. J.B Ra, J. Yi, and S. Ko, "*A Semiautomatic Segmentation Tool for XCT Body Images Using Watershed Algorithm,*" Proc. KOSOMBE Conference, vol. 21, no. 1, pp. 68-69, KyungSan, May 1999