

변형된 마칭큐브 테이블을 이용한 볼륨 광선 투사법 가속화

(Volume Ray Casting Acceleration Method using Modified Marching Cubes Tables)

임 석 현 [†] 김 주 환 ^{**} 신 병 석 ^{***}
 (Sukhyun Lim) (Ju-Hwan Kim) (Byeong-Seok Shin)

요 약 CPU기반의 볼륨 광선 투사법을 위하여 빈 공간을 도약하는 많은 방법들이 소개되었다. 하지만 광선의 샘플점이 투명한 공간과 비투명한 공간 사이에 있는 반투명-셀에 놓일 경우 기존의 방법들은 투명도 값을 매번 계산해야 하는 문제점이 있다. 이 문제를 해결하기 위하여 기존 마칭큐브를 이용하는 방법이 제안되었다. 광선이 반투명-셀에 도착하면 마칭큐브 방법을 이용하여 생성된 삼각형들을 둘러싸고 있는 육면체를 구하고, 현재 샘플점이 육면체 내/외부에 있는지 계산한다. 샘플점이 육면체 외부에 위치하면 투명하다고 판단되어 재샘플링 연산없이 다음 샘플점으로 이동한다. 하지만 기존의 마칭큐브 테이블 구조를 그대로 이용하면 인접한 복셀의 테이블 값을 여러 번 참조해야 하므로 비효율적이다. 본 논문에서는 볼륨 광선 투사법에 적합한 마칭큐브 테이블을 제안함으로써 렌더링 속도를 높인다.

키워드 : 볼륨가시화, 볼륨 광선 투사법, 마칭큐브, 반투명셀, 공간도약

Abstract Several empty-space leaping methods have been proposed for CPU-based volume ray casting. When sample points are located in semi-transparent cells, however, previous leaping methods perform unnecessary resamplings even if the scalar values on those points are confined within transparent range. A semi-transparent cells leaping method for volume ray casting using the Marching Cubes algorithm is proposed to solve this problem in our previous work. When a ray reaches a semi-transparent cell, our method performs in-out test between current sample point and the bounding box enclosing the triangles generated by the Marching Cubes. If the sample point lies on outside of the bounding box, we estimate the point is regarded as transparent. In this case, the ray advances to the next sample point without performing a resampling operation. We can frequently refer the tables for neighboring voxels, however, when we exploit conventional data structures of the Marching Cubes. We propose modified Marching Cubes tables for solving this problem.

Key words : Volume visualization, volume ray casting, Marching Cubes, semi-transparent cell, space leaping

· 본 연구는 인천정보산업진흥원이 주관하는 "실감형3D영상원천기술개발 사업"의 지원으로 수행되었음
 · 이 논문은 제35회 추계학술대회에서 '변형된 마칭큐브 테이블을 이용한 볼륨 광선 투사법 가속화'의 제목으로 발표된 논문을 확장한 것임

[†] 학생회원 : 인하대학교 컴퓨터정보공학과
s.lim@inha.ac.kr

^{**} 비 회 원 : 인하대학교 컴퓨터정보공학과
jujusid@hotmail.com

^{***} 종신회원 : 인하대학교 컴퓨터정보공학과 교수
bsshin@inha.ac.kr
(Corresponding author임)

논문접수 : 2008년 12월 22일
심사완료 : 2009년 3월 16일

Copyright©2009 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 시스템 및 이론 제36권 제3호(2009.6)

1. 서론

볼륨 렌더링(volume rendering)은 볼륨 데이터로부터 의미 있는 정보를 추출하여 가시화하는 기법이다 [1]. 볼륨 데이터는 복셀(voxel)로 표현할 수 있으며, 여덟 개의 복셀들로 구성된 육면체 구조를 “셀(cell)”이라 한다. 모든 복셀이 투명하면 “투명-셀 (transparent cell)”이라 하고, 모두 비투명하면 “비투명-셀(nontransparent cell)”이라 한다. 또한 투명한 복셀과 비투명한 복셀이 모두 존재하는 셀을 “반투명-셀(semi-transparent cell)”이라 한다.

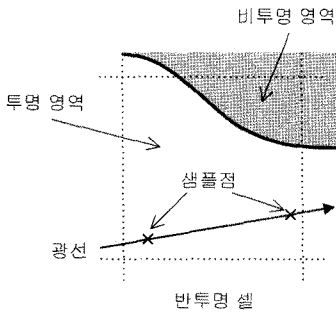


그림 1 반투명-셀의 예

일반적으로 볼륨 렌더링 기법에는 표면 렌더링 기법(surface rendering)과 직접 볼륨 렌더링 기법(direct volume rendering)이 있다. 마칭큐브(Marching Cubes) 방법은 표면 렌더링 기법 중 대표적인 방법으로 볼륨 데이터를 메쉬(mesh) 형태로 표현하는 기법이다[2]. 직접 볼륨 렌더링 기법은 볼륨 데이터를 메쉬 형태로 재구성하지 않고 직접 렌더링 하는 기법으로, 가장 대표적인 방법으로는 볼륨 광선 투사법(volume ray casting)이 있다[3].

볼륨 광선 투사법은 화소(pixel)로부터 가상의 광선이 발사되면 투명-셀들을 도약한 후, 반투명-셀이나 비투명-셀에 놓이면 현재 샘플점이 실제로 사용자가 원하는 물체의 표면이나 내부에 있는지 정확히 판단하기 위해 재샘플링(resampling) 필터를 이용하여 밀도값을 추정한다. 재샘플링 필터 중 볼륨 광선 투사법에서 많이 사용되는 것은 삼선형 보간법(trilinear interpolation)이다. 이는 일곱 번의 선형 보간으로 구성되어 있기 때문에 정확한 밀도값을 계산할 수 있지만 계산량이 많아서 속도가 느린 단점이 있다[4].

이런 문제를 해결하기 위하여 마칭큐브 알고리즘[2]을 직접 볼륨 렌더링 기법 중 하나인 볼륨 광선 투사법[3]에 적용하여 재샘플링 단계 이전에 재샘플링 필터의 적용 여부를 판단하는 방법이 제안되었다[5]. 하지만 기

존의 마칭큐브 방법은 기하정보를 얻는 데 최적화되어 있기 때문에 볼륨 광선 투사법에 적용 시 육면체를 생성하기 위하여 작은 복셀의 스칼라값(scalar value)을 참조해야 한다. 본 방법에서는, 기하정보가 아닌 육면체의 교점 정보만 필요하기 때문에 기존의 마칭큐브 구조를 변형하는 방법을 제안한다.

논문의 구성은 다음과 같다. 2.1절에서 본 논문의 기본이 되는 마칭큐브에 기반한 가속기법을 설명하고, 2.2절에서 본 논문에서 제안하는 변형된 마칭큐브 테이블 구조를 이용하여 렌더링 성능을 향상시키는 기법을 설명한다. 다음으로 실험결과를 보이고 결론을 맺는다.

2. 관련 연구

Yagel은 광선의 진행 경로에 따른 샘플의 위치와 재샘플링 가중치를 미리 계산해 놓은 원형(template)을 이용하여 샘플링 연산을 단축시켰다[6]. 하지만 이 방법은 직교투영(orthogonal projection)에만 가능하며 투시투영(perspective projection)이 필요한 분야에는 이용하기 힘들다. 쉬어-왁(shear warp) 볼륨 렌더링은 메모리 참조 효율성을 높여서 색상 계산 단계를 가속하는 방법이다[7]. 하지만 대화적 분류(interactive classification) 환경에서 캐시 미스(cache miss)를 유발하며, 메모리 참조 효율성을 높이기 위해 재샘플링 필터로 쌍선형 보간(bilinear interpolation)을 이용하기 때문에 영상의 질이 떨어진다. 인텔(IntelTM)사의 MMXTM(MultiMedia eXtension) 기술을 이용하여 캐시 연산을 가속하는 볼륨 광선 투사법 시스템이 발표되었다[8]. 이는 실시간에 가까운 렌더링 속도를 보이지만 영상의 크기가 256×256으로 제한되어 있어, 그 이상의 해상도를 원할 경우 256×256의 영상을 확대해야 한다. 이 경우, 영상이 흐릿(blurring)하게 보이는 문제점이 발생한다. Mora 등은 볼륨 광선 투사법에 스플래팅(splating) 개념을 도입하여 셀들을 동일한 큐브(cube) 모양으로 투영하는 객체-기반 볼륨 광선 투사법(object order volume ray casting)을 제안하였다[9]. 이 방법은, 렌더링 속도는 빠르지만 투시투영에는 이용할 수 없다는 문제점을 가지고 있다. 최근에는 CPU의 캐시(cache) 공간에 저장되어 있는 기울기 벡터(gradient vector)를 효율적으로 활용하여 메모리를 임의 참조하는 횟수를 줄이는 방법이 소개되었다[10]. 이 방법을 사용하면 범용의 노트북 PC에서 초당 10프레임 이상의 속도를 보이지만, 이 방법 역시 직교투영만 가능하다.

3. 본론

본 논문에서는 현재 조사하려는 샘플점으로부터 마칭큐브 알고리즘을 이용하여 생성되는 모든 삼각형들을

둘러싸는 육면체 구조를 이하 “육면체”라 한다.

3.1 마칭큐브 알고리즘을 이용한 반투명-셀 도약

초기의 볼륨 광선 투사법은 투명도 값을 계산하기 위하여 재샘플링 필터를 적용했을 뿐만 아니라 매번 색상값도 계산했다. 불투명한 공간에서는 색상값을 계산하지 않아도 되므로, 현재는 투명도값을 먼저 계산한 후 투명하지 않은 경우에만 색상값을 계산한다. 하지만 이 방법 또한 비효율적이다. 반투명-셀은 투명한 부분과 비투명한 부분이 동시에 존재하므로 삼선형 보간으로 계산된 밀도값은 투명할 수도 있다. 그림 1에서 두 개의 샘플점은 반투명-셀에 놓여있지만 투명한 값을 가지고 있기 때문에 최종 영상에 영향을 주지 않는다. 기존에 발표되었던 빈 공간 도약방법들은 오로지 투명-셀들만 도약할 수 있기 때문에 반투명-셀들에서는 가속이 불가능하다[7-16].

이를 해결하기 위하여, 마칭큐브 알고리즘을 이용하여 반투명-셀들에서 재샘플링 필터를 적용하지 않는 방법이 제안되었다[5]. 마칭큐브 알고리즘으로 생성된 육면체 영역과 현재의 광선에서 샘플점간의 인-아웃 검사(in-out test)를 시행하여, 샘플점이 생성된 육면체 안에 들어있지 않다면 이를 투명하다고 판단하여 재샘플링 필터를 적용하지 않는다(그림 2(a) 참고).

그림 3은 반투명-셀을 도약하는 예이다. s_1 샘플점은 투명-셀에 놓여 있기 때문에 기존의 투명 공간 도약법[11-16]으로 진행할 수 있다. 광선이 s_2, s_3, s_4, s_5 와 같은 반투명-셀에 놓이게 되면 육면체 영역과의 인-아웃 검사를 통하여 재샘플링 필터의 적용여부를 판단하게 된다. s_2, s_3, s_4 의 경우 샘플점이 육면체 밖에 놓이기 때문에 이들은 재샘플링 과정 없이 다음 번 샘플점으로

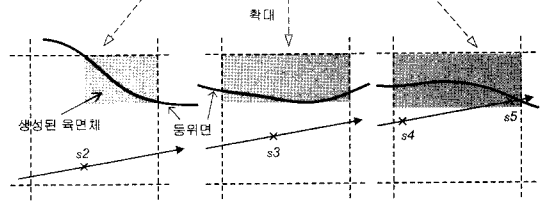
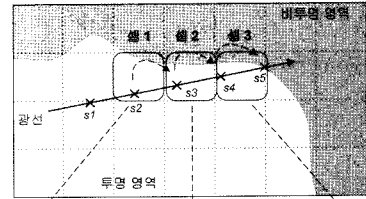


그림 3 마칭큐브 알고리즘을 이용한 재샘플링 단계의 유무 판별. s_2, s_3, s_4 샘플점들은 육면체 밖에 존재하므로 재샘플링 필터를 적용하지 않는다. s_5 와 같이 육면체 내부에 놓일 경우에만 재샘플링 필터를 적용한다.

이동하게 된다. s_5 와 같이 샘플점이 육면체 내부에 놓여 비투명할 경우에만 재샘플링 필터를 적용한다. 육면체는 모든 삼각형들을 포함하고 있기 때문에 화질저하 없이 렌더링 시간을 단축시킬 수 있다.

하지만 마칭큐브의 15가지 경우 모두를 볼륨 광선 투사법에 이용할 수는 없다. 삼선형 보간은 일곱 번의 선형 보간으로 구성된다[4]. 하지만 마칭큐브로 생성된 삼각형들을 둘러싸는 육면체를 생성할 때, 선형 보간 횟수가 일곱 번 이상이면 이 방법을 적용해도 속도 향상을 기대할 수 없으므로 제외한다. 마칭큐브의 15가지 경우 중에서 경우 7, 경우 10, 경우 13이 각각 9, 8, 12번의 선형 보간이 필요하다. 두 번째로, 마칭큐브로 생성된 삼각형들을 둘러싸는 육면체가 전체 셀을 덮고 있다면 역시 고려할 필요가 없다. 샘플점과의 인-아웃 검사 결과가 항상 육면체 영역 안에 샘플점이 들어가 있는 것으로 나오기 때문에 속도가 향상되지 않는다. 마칭큐브의 15가지 경우 중에서 경우 9, 경우 11, 경우 14가 이에 해당한다. 마지막으로 경우 0번은, 기존의 빈 공간 도약법으로 이미 도약한 후이기 때문에 제외한다.

세 가지 예외사항을 적용한 볼륨 광선 투사법에 최종적으로 응용 가능한 마칭큐브의 경우는 그림 4와 같다. 그림에서 회색으로 표시한 부분은 생성된 육면체를 나타낸다.

3.2 볼륨 광선 투사법을 위한 마칭큐브 테이블의 변형

기존의 마칭큐브는 색인-테이블(index table)과 변-테이블(edge table)을 이용한다. 색인-테이블은 8비트로 구성되며 셀의 여덟 개 복셀의 밀도값에 대한 투명도가 불투명한 경우 해당 비트가 1로 설정되며, 그 값에 따라

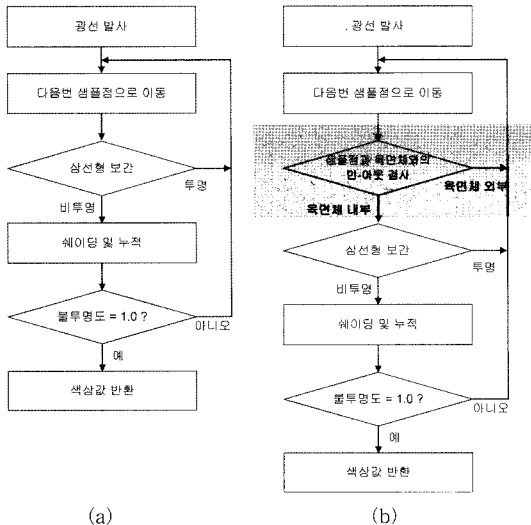


그림 2 기존의 볼륨 광선 투사법 파이프라인(a)과 마칭큐브를 이용한 볼륨 광선 투사법 파이프라인(b)

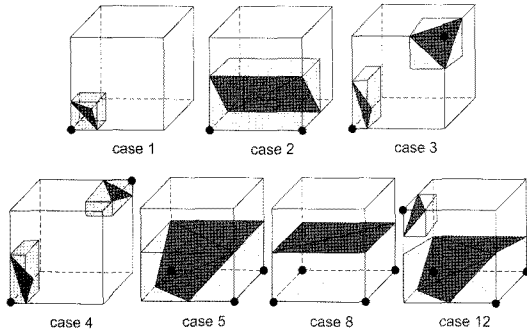


그림 4 볼륨 광선 투사법의 가속을 위해 선택된 7개의 경우들. 음영으로 된 부분은 생성된 삼각형들과 육면체이다.

변-테이블에서의 색인을 결정한다(그림 5 참조). 이와 같은 자료구조는 생성되는 삼각형들의 기하정보를 저장하고 있기 때문에 기존의 마칭큐브에서는 매우 최적화되어 있는 구조이다. 하지만 본 방법은 육면체의 교차점만을 빠르게 계산하는 것이 필요하기 때문에 기존의 테이블을 변형한다.

그림 6은 제한한 마칭큐브 변-테이블의 구조이다. 각각은 13개의 필드로 구성되어 있다. 첫 번째 필드는 마칭큐브의 몇 번째 경우에 해당하는지를 저장한다. 만약

-1이 저장되어 있으면 본 방법에서 사용하지 않는다는 것을 의미한다. 이 경우 삼선형 보간을 수행한다. 나머지 12개의 필드는 셀에서의 12개의 면에 대한 정보를 저장하고 있으며, 각각의 필드는 3개의 서브필드(sub-field)로 구성되어 있다.

각 서브필드의 구성 요소 중, 첫 번째와 두 번째 서브 필드는 육면체의 교차점의 위치를 알기 위한 복셀 위치이다. 본 방법에서는 삼각형들의 기하정보가 아닌 교차점의 위치가 필요하기 때문에 효율을 높이기 위하여 교차점이 존재하는 복셀의 위치를 저장한다. 세 번째 서브 필드에는 육면체 영역을 저장한다. 비록 첫 번째와 두 번째 서브필드를 보고 양 복셀 사이에 교차점이 존재한다는 것을 알 수 있지만, 첫 번째 복셀로부터 교차점까지가 육면체인지, 교차점으로부터 두 번째 복셀까지가 육면체인지는 알 수 없다. 따라서, 세 번째 서브필드에는 육면체의 방향 정보를 저장하여 이 문제를 해결한다.

만약 각각의 축에 위치하는 교차점이 두 개 이상이고, 세 번째 서브필드의 값이 같을 경우, 최대값과 최소값을 고려해야 한다. 생성되는 모든 삼각형들을 둘러싸는 육면체를 생성해야 하기 때문에 방향 성분이 같다면 최대값과 최소값을 선택해야 한다. 세 번째 서브필드가 0이면, 우리는 동일 축의 두 개 이상의 교차점 중 최대값을 선택한다. 반대로, 세 번째 성분이 1이라면, 최소값을 선

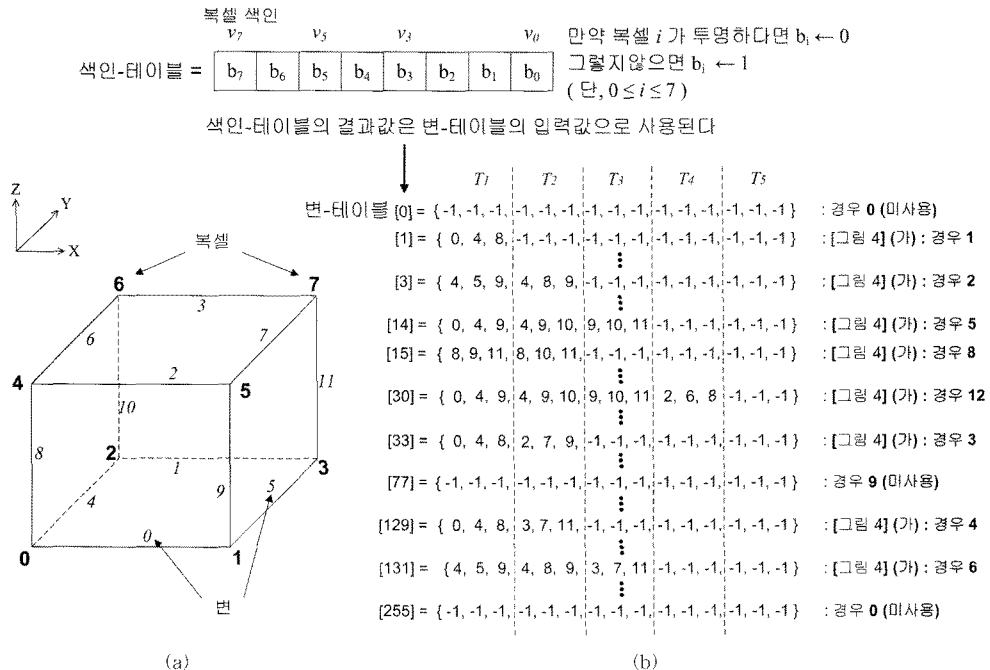


그림 5 기존 마칭큐브 테이블의 구조. 8비트로 구성된 색인-테이블을 이용하여 15비트로 구성된 변-테이블을 참조하여 기하정보를 구한다.

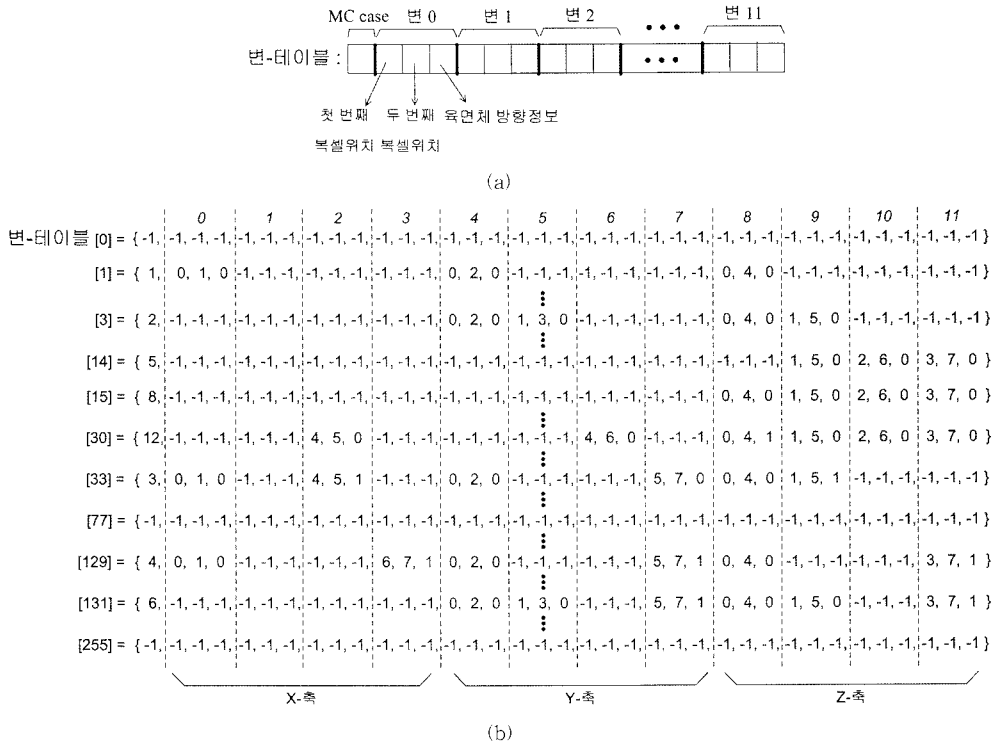


그림 6 (a) 제안하는 변-테이블 구조. 총 13개의 필드로 구성되어 있으며, 첫 번째 필드를 제외한 나머지 필드는 3개의 서브필드로 구성되어 있다. (b) 제안한 변-테이블. 각각은 그림 5와 연결된다.

백한다. 그러나 육면체 영역이 셀의 특정 축을 모두 덮고 있다면 이는 고려할 필요가 없다. 이 경우, 해당 서브필드에는 -1을 배정하여 고려하지 않도록 한다.

결국 본 논문에서 제안하는 방법은 기존의 테이블 구조에 비해 최소한의 복셀의 스칼라값 참조를 위하여 인-아웃 테스트에 필요한 정보를 모두 담고 있으므로 렌더링 성능이 향상된다.

4. 실험 결과

실험은 AMD사의 애슬론(Athlon)64x2TM 4200+ CPU 와 2GB의 주 메모리를 탑재한 PC에서 수행하였다. 실험 데이터는 모두 여섯 가지를 사용하였으며, 크기는 다음 표 1과 같다.

본 논문에서 제안하는 방법은 반투명-셀들을 가속화하기 위한 방법이므로 빈공간을 효과적으로 도약하기

위한 빈 공간 도약법이 필요하다. 이를 위해서 거리맵 자료구조(distancemap)[11,12]와 계층적 최대-최소 옥트리(hierarchical min-max octree) 자료구조[13-16]를 이용하였다. 계층적 최대-최소 옥트리를 사용한 경우 투명하다고 판단된 옥탄트(octant)의 효율적인 도약을 위하여 거리원형(distance template) 자료구조를 이용하였다[16].

표 2는 시점을 고정한 후, 본 논문에서 제안한 방법을 사용했을 때 렌더링 속도 향상 폭을 보여준다. 실험결과 약 20% 정도의 속도 향상이 있었음을 확인하였다. 기존의 방법으로는 13% 정도가 향상되었었다[5]. 기존의 방법과 본 논문에서 제안한 방법 모두 투시투영으로 렌더링하였으며, 영상의 크기는 512x512이다.

표 2 기존 방법과 렌더링 효율 비교(단위: %). 빈공간 도약법으로는 거리맵 자료구조(경우 1)와 계층적 최대-최소 옥트리(경우 2)를 사용하였다.

표 1 실험 데이터

데이터	크기	데이터	크기
머리	512x512x450	엔진	512x512x512
뇌동맥	512x512x512	주전자	512x512x512
치아	512x512x322	분개	512x512x512

구분	머리	뇌동맥	치아
경우1	18.8	23.0	21.5
경우2	16.8	19.6	18.1
구분	엔진	주전자	분개
경우1	20.5	22.0	24.1
경우2	18.9	20.1	21.7

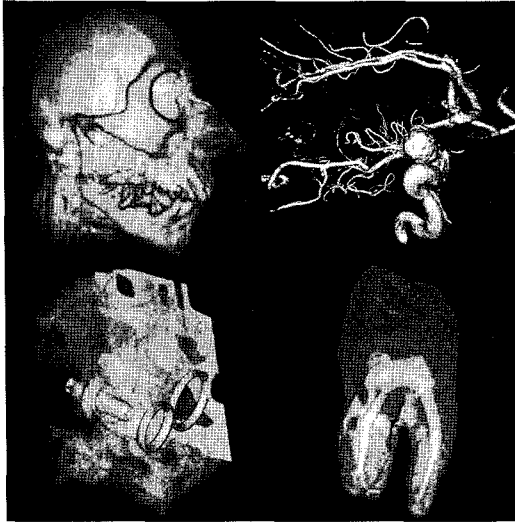


그림 7 렌더링 결과

또한 본 논문에서 제안한 방법으로 생성한 결과 영상을 검증하기 위하여 적용 전후의 영상 각각의 화소별로 비교해본 결과 차이가 없음을 확인하였다. 이는 본 논문의 방법을 적용했을 때 영상의 화질저하 없이 렌더링 속도가 향상됨을 의미한다.

5. 결론

볼륨 렌더링에서는 전처리 과정이나 추가적인 자료구조 없이 렌더링 속도를 향상시키는 것이 중요하다. 이를 위해, 마칭큐브 알고리즘을 볼륨 광선 투사법에 적용하여 반투명-셀들을 도약하는 방법이 제안되었다. 하지만 이 방법은 기하구조를 나타내기 위한 마칭큐브 테이블 구조를 그대로 사용함으로써 인접 복셀의 스칼라값을 빈번하게 참조해야 했다. 본 논문에서는 이 방법의 성능 향상을 위해 기존 방법의 테이블 구조를 변형하여 렌더링 속도를 줄이는 방법을 제안하였다. 이 결과 20% 정도의 렌더링 속도 향상을 얻었다. 제안한 방법은 대부분의 볼륨 광선 투사법의 가속기법에 적용 가능하며, 속도 저하나 추가적인 자료구조 없이 투시투영에도 적용이 가능하다.

참고 문헌

[1] Engel, K., Weiskopf, D., Rezk-salama, C., Kniss, J. and Hadwiger, M., Real-time Volume Graphics. AK Peters, 2006.
 [2] Lorensen, W.E. and Cline, H.E., "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," SIGGRAPH 1987, Vol.21, pp. 163-169, 1987.

[3] Levoy, M., "Display of Surfaces from Volume Data," IEEE Computer Graphics and Applications, Vol.5, No.4, pp. 29-37, 1988.
 [4] Hill, S., "Trilinear Interpolation," In P.S. Heckbert, Graphics Gems IV. Pages 521-525, Academic Press Professional, San Diego, CA, 1994.
 [5] Lim, S. and Shin, B.S., "On-the-fly Space-leaping for Semi-transparent Cells Using Marching Cubes," Pacific Graphics 2006, pp. 176-179, 2006.
 [6] Yagel, R. and Kaufman, A., "Template-based Volume Viewing." Proc. Eurographics 1992, Vol.11, No.3, pp. 153-167, 1992.
 [7] Lacroute, P. and Levoy, M., "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation," Proc. SIGGRAPH 1994, pp. 451-457, 1994.
 [8] Knittel, G., "The UltraVis System," Proc. IEEE Volume Visualization 2000, pp. 71-79, 2000.
 [9] Mora, B., Jessel, J.P. and Caubet, R., "A New Object Order Ray-casting Algorithm," Proc. IEEE Volume Visualization 2002, pp. 203-210, 2002.
 [10] Grimm, S., Bruckner, S., Kanitsar, A. and Gröller, E., "Memory Efficient Acceleration Structures and Techniques for CPU-based Volume Raycasting of Large Data," Proc. IEEE Volume Visualization 2004, pp. 1-8, 2004.
 [11] Zuiderveld, K., Koning, A. and Viergever, M., "Acceleration of Ray-casting Using 3D Distance Transforms," Proc. Visualization in Biomedical Computing 1992, pp. 324-335, 1992.
 [12] Cohen, D. and Shefer, Z., "Proximity Clouds - An Acceleration Technique for 3D Grid Traversal," The Visual Computer, Vol.10, No.11, pp. 27-38, 1994.
 [13] Hadwiger, M., Sigg, C., Scharsach, H., Bühler, K. and Gross, M., "Real-time Ray-casting and Advanced Shading of Discrete Isosurfaces," Graphics Forum, Vol.24, No.3, pp. 303-312, 2005.
 [14] Lim, S. and Shin, B.S., "Efficient Space-Leaping Using Optimal Block Sets," IEICE Trans. on Information and Systems, Vol. E88-D, No.12, pp. 2864-2870, 2005.
 [15] Lim, S. and Shin, B.S., "A Half-Skewed Octree for Volume Ray Casting," IEICE Trans. on Information and Systems, Vol. E90-D, No.7, pp. 1085-1091, 2007.
 [16] Lim, S. and Shin, B.S., "A Distance Template for Octree Traversal in CPU-based Volume Ray Casting," The Visual Computer, Vol.24, No.4, pp. 229-237, 2008.



임 석 현

1999년 인하대학교 전자계산공학과, 물리학과 학사. 2001년 인하대학교 전자계산공학과 석사. 2006년 인하대학교 컴퓨터정보공학과 박사. 2007년~현재 인하대학교 연구교수. 관심분야는 불륨그래픽스, 실시간렌더링, 차세대컴퓨터



김 주 환

2007년 인하대학교 컴퓨터정보공학과 2009년 인하대학교 정보공학과 석사. 관심분야는 불륨렌더링



신 병 석

1990년 서울대학교 공과대학 컴퓨터공학과. 1992년 서울대학교 대학원 컴퓨터공학과 석사. 1997년 서울대학교 대학원 컴퓨터공학과 박사. 2000년~현재 인하대학교 컴퓨터공학부 교수. 관심분야는 실시간렌더링, 불륨렌더링, 의료영상, 차

세대컴퓨터