

블리킹을 이용한 대용량 초음파 볼륨 데이터 렌더링

김주환*, 권구주**, 신병석***

Large-Scale Ultrasound Volume Rendering using Bricking

Ju Hwan Kim*, Koojoo Kwon**, Byeong-Seok Shin***

요약

최근 높은 해상도의 볼륨 데이터를 획득할 수 있게 되면서 제한된 용량의 메모리를 가진 그래픽 하드웨어에서 대용량 볼륨 데이터를 렌더링 하는 방법이 필요하게 되었다. 대용량 볼륨 데이터의 렌더링 방법 중 데이터를 적절히 분할하여 순차적으로 처리하는 블리킹(bricking) 방법이 많이 사용된다. 그러나 일반적인 블리킹 방법은 직교 좌표계를 사용하는 CT와 MR 데이터를 위해 고안된 것으로, 원환체(torus) 좌표계를 사용하는 부채꼴 형태의 초음파 볼륨 데이터에 적용하면, 관측광선이 블릭(brick)의 곡면경계로 진입한 후 다시 빠져 나갈 때 동일한 블릭이 GPU메모리에 두 번 적재되는 경우가 발생한다. 본 논문에서는 초음파 볼륨을 렌더링 할 때 반복적인 텍스처 스위칭이 발생하지 않도록 블릭의 크기를 결정하는 방법을 제안한다. 블릭의 경계는 곡면으로 되어 있으므로 이들의 곡률을 계산하여, 관측광선이 동일한 블릭을 두 번 참조하는 영역을 찾는다. 이 영역에 해당하는 복셀들을 인접한 두 블릭들이 공유하도록 크기를 정하면 둘 중의 한 블릭에서만 재샘플링하게 함으로써 블릭이 중복 적재되는 것을 피할 수 있다.

Abstract

Recent advances in medical imaging technologies have enabled the high-resolution data acquisition. Therefore visualization of such large data set on standard graphics hardware became a popular research theme. Among many visualization techniques, we focused on bricking method which divided the entire volume into smaller bricks and rendered them in order. Since it switches between bricks on main memory and bricks on GPU memory on the fly, to achieve better performance, the number of these memory swapping conditions has to be minimized. And, because the original bricking algorithm was designed for regular volume data such as CT and MR, when applying the algorithm to ultrasound volume data which is based on the toroidal coordinate space, it revealed some performance degradation. In some areas near bricks' boundaries, an orthogonal viewing ray intersects the same brick twice, and it consequently makes a single brick memory to be uploaded onto GPU twice in a single frame. To avoid this redundancy, we divided the volume into bricks allowing overlapping between the bricks. In this paper, we suggest the formula to determine an appropriate size of these shared area between the bricks. Using our formula, we could minimize the memory bandwidth, and, at the same time, we could achieve better rendering performance.

▶ Keyword : 대용량 볼륨 데이터(Large-Scale Volume Data), 블리킹(Bricking), 볼륨 광선 투사법(Volume Ray Casting), 초음파 볼륨 데이터(Ultrasound Volume Data)

• 제1저자 : 김주환 교신저자 : 신병석

• 접수일 : 2008. 10. 11, 심사일 : 2008. 10. 25, 심사완료일 : 2008. 12. 24.

* 인하대학교 정보공학과 컴퓨터정보공학부 ** 인하대학교 정보공학과 컴퓨터정보공학부 강의전임 교수

*** 인하대학교 정보공학과 컴퓨터정보공학부 부교수

1. 서론

컴퓨터 단층촬영장치(CT), 자기공명장치(MR), 초음파 진단장치 등 의료기기를 이용하여 획득한 다양한 의학영상들은 의사들의 진단을 돕는 중요한 도구이다. 특히, 초음파 데이터는 CT, MR데이터와 달리 획득비용이 저렴하며, 빠른 속도로 임의의 방향에서 스캔하여 결과를 확인 할 수 있기 때문에 많이 사용된다[1][2][3].

일반적으로 의료영상 데이터를 가시화하기 위해 볼륨 렌더링 방법을 사용한다. CT나 MR과 같이 직교 좌표상에서 획득된 볼륨을 사용하는 볼륨 렌더링 방법들은 직교 좌표계에서 각 복셀을 참조하는 반면에 부채꼴 형태로 초음파 빔을 방사하여 획득된 초음파 볼륨 데이터는 원환체(torus) 좌표계를 사용하기 때문에 특별한 좌표변환 과정이 필요하다 [4]. 이러한 변환 과정은 렌더링 속도를 저하시키는 원인이 되는데, 최근에는 GPU에서 좌표변환 연산을 하도록 함으로써 빠른 시간에 렌더링 할 수 있게 되었다 [5][6].

좀 더 정확한 진단을 위해서는 정밀한 데이터가 필요하며, 이 때문에 영상 데이터의 크기가 증가하고 있다. 대용량 의료 영상 데이터를 제한된 메모리를 가지는 그래픽 하드웨어에서 렌더링 하기 위한 다양한 방법들이 제안되었다[7][8][9][10][11][12][13]. 이 중에서 볼륨데이터를 여러 개의 작은 블록들로 나누어서 GPU에서 차례로 렌더링한 후 조합하는 방법을 블리킹이라고 하며, 하나의 작은 블록을 블럭이라고 한다 [13]. 블럭은 GPU 메모리 크기보다 작은 3차원 텍스처로서 누적 순서에 맞게 순차적으로 메인 메모리에서 GPU 메모리로 전송된 후 렌더링된다. 이때 블럭의 크기에 따라, 제어 비용과 GPU의 캐쉬 효율이 달라지므로 적절한 블럭의 크기를 정하는 것이 중요하다 [14].

본 논문에서는 대용량의 초음파 볼륨데이터를 GPU에서 렌더링할 때 블럭의 크기를 결정하는 방법을 제안한다. 초음파 볼륨 데이터는 직교 좌표계상에서 샘플링된 후 원환체 좌표로 인덱싱되는 GPU 메모리에 저장되므로 렌더링 과정에서 직선으로 진행되는 관측광선이 물리적 메모리 공간에서는 곡선으로 진행된다. 따라서 직교 좌표계에서 샘플링된 볼륨 데이터를 렌더링하는데 사용하는 블리킹 방법을 초음파 볼륨 데이터에 적용할 경우, 관측광선이 두 블럭의 곡면경계로 진입한 후 다시 빠져 나갈 때 이미 사용했던 블럭을 다시 적재해야 하는 경우가 발생한다. 본 논문에서는 인접한 두 개의 블럭을 부분적으로 중첩시켜서 동일한 블럭을 반복해서 적재하지 않도록 하였다.

본 논문의 2장에서는 기존의 블리킹 방법과 초음파 볼륨 데이터의 특성을 설명하고, 3장에서는 본 논문에서 제안하는 초음파 볼륨 데이터의 블리킹하는 과정을 상세하게 기술한다. 4장에서는 제안하는 방법의 실험 결과를 보이고, 마지막으로 5장에서는 결론 및 향후 연구 방향을 기술한다.

II. 관련 연구

초음파 볼륨 데이터는 그림 1에서 보는 것과 같이 프로브(probe)에서 원환체 형태로 초음파 빔을 순차적으로 방사한 후 (그림 1(a)), 프로브에서 얻은 연속된 2차원 영상들을 모아 (그림 1(b)) 초음파 볼륨 데이터를 만들고 (그림 1(c)) 이것을 메모리에 순차적으로 저장한다 (그림 1(d)). 이 데이터를 렌더링하기 위해서는 그림 2에서 보는 것과 같이 관측광선이 진행되는 논리적 좌표 공간과 (그림 2(a)) 물리적 메모리에 데이터가 저장된 원환체 좌표 공간 (그림(2(b)) 사이의 좌표변환과정이 필요하다. 최근의 초음파 볼륨 데이터 획득 장치는 심장의 움직임, 태아의 모습 등을 실시간으로 획득할

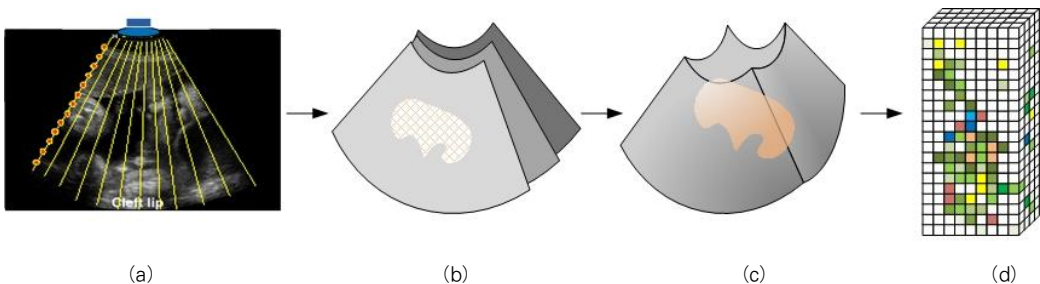


그림 1. 초음파 볼륨 데이터 획득 과정
Fig. 1. Acquisition Procedure of Ultrasound Volume Data

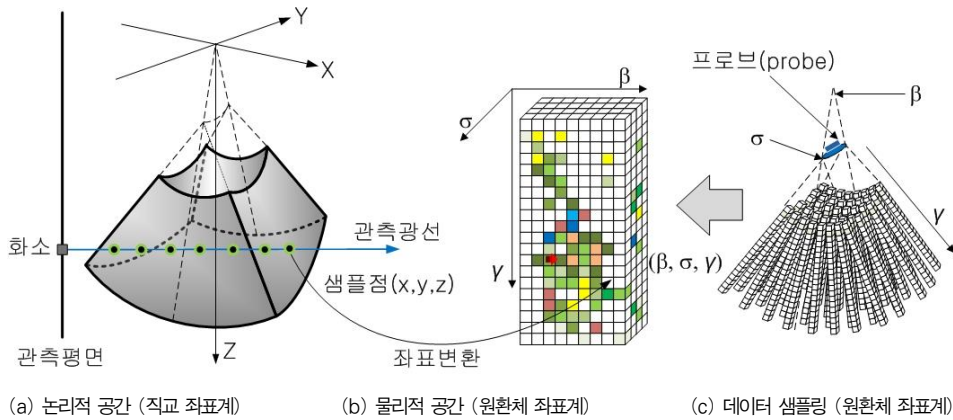


그림 2. 초음파 볼륨 렌더링에서 좌표변환과정
 Fig. 2. Coordinate Conversion in Ultrasound Volume Rendering

수 있을 정도로 빠르기 때문에 렌더링도 실시간에 되어야 하며 이를 위해 GPU를 이용한 고속 초음파 렌더링 방법이 제안되었다 [5][6].

초음파 볼륨렌더링을 위해 일반적으로 좋은 화질의 영상을 만들어 내는 것으로 알려진 광선 투사법을 사용한다 [15][16][17]. 이 방법은 관측평면의 각 화소값을 결정하기 위해 발사된 관측광선을 따라 일정간격으로 샘플링된 값들을 누적한다. 논리적 공간에 위치한 각 샘플점의 값을 알기 위하여 직교 좌표상의 샘플점 위치를 원환체 좌표상의 위치로 변환하고 물리적 메모리에서 값을 읽어온다. 직교 좌표의 위치 (x, y, z) 를 원환체 좌표의 위치 (β, σ, γ) 로 변환하는 수식은 다음과 같다 [4].

$$R = \sqrt{y^2 + z^2} - a \dots\dots\dots (2.1)$$

$$\beta = \frac{\pi}{2} + \tan^{-1}\left(\frac{y}{z}\right) \quad (0 \leq \beta \leq \pi) \dots\dots\dots (2.2)$$

$$\sigma = \frac{\pi}{2} + \tan^{-1}\left(\frac{x}{R}\right) \quad (0 \leq \sigma \leq \pi) \dots\dots\dots (2.3)$$

$$\tau = \sqrt{x^2 + R^2} - b \dots\dots\dots (2.4)$$

여기서, β 는 프로브 내부의 스텝핑 모터의 스윙(swing) 각도 범위인 스캔 관측 각도이고, σ 는 프로브로부터 스캐닝 되는 2차원 영상의 넓이 각도 범위인 프로브 관측 각도이다. γ 는 초음파 영상이 획득 되는 부분부터 초음파 영상내 임의의 3차원 데이터까지의 거리이며, R 은 초음파 영상내 임의의 3차원 데이터의 위치를 YZ 평면에 투사한 위치에서 프로브까지

의 거리이다. a 는 스캔 관측 각도의 꼭지점에서 프로브 관측 각도의 꼭지점까지의 거리이며, b 는 프로브 관측 각도의 꼭지점에서 3차원 초음파 영상이 획득되는 부분까지의 거리이다. 그리고 x, y, z 는 직교 좌표의 위치이다 [5].

그림 3은 초음파 볼륨 데이터의 좌표계를 보여준다. (a)는 임의의 3차원 초음파 데이터를 3차원 직교 좌표계로 표현되는 화면에서 나타낸 것이고, (b)는 YZ 평면 그리고 (c)는 XR^* 평면에 투사된 좌표계이다. (d)는 XR^* 평면의 정의와 R^* 축의 범위를 보여준다. X 축, Y 축, Z 축은 직교 좌표계를 구성하는 축이며, R^* 축은 3차원 초음파 영상이 획득되는 부분부터 X 축에 직교되는 축이다.

식 2.1은 그림 3(b)의 Z 축을 한 변으로 하는 삼각형에서 피타고라스 정리를 이용하여 $(R+a)^2 = y^2 + z^2$ 을 R 에 관한 식으로 정리한 것이고, 식 2.4는 그림 3(c)의 R^* 축을 한 변으로 하는 삼각형에서 피타고라스 정리를 이용하여 $(\gamma+b)^2 = x^2 + R^2$ 을 γ 에 관한 식으로 정리한 것이다. 식 2.2는 그림3(b)에서 보듯이 β 에서 $\pi/2$ 를 뺀 값은 Z 축을 한 변으로 하는 삼각형에서 R 과 z 가 이루는 사이각이고, $\tan(\beta - \pi/2) = y/2$ 가 성립하며 이를 β 에 관한 식으로 정리한 것이다. 식 2.3은 그림3(c)에서 보듯이 σ 에서 $\pi/2$ 를 뺀 값은 R^* 축을 한 변으로 하는 삼각형에서 γ 와 R 이 이루는 사이각이고, $\tan(\sigma - \pi/2) = x/R$ 이 성립하며 이를 σ 에 관한 식으로 정리한 것이다.

볼륨 렌더링 분야에서 블리킹은 대용량 볼륨을 가시화하기 위한 대표적인 방법이다. 먼저 볼륨 데이터를 GPU메모리에

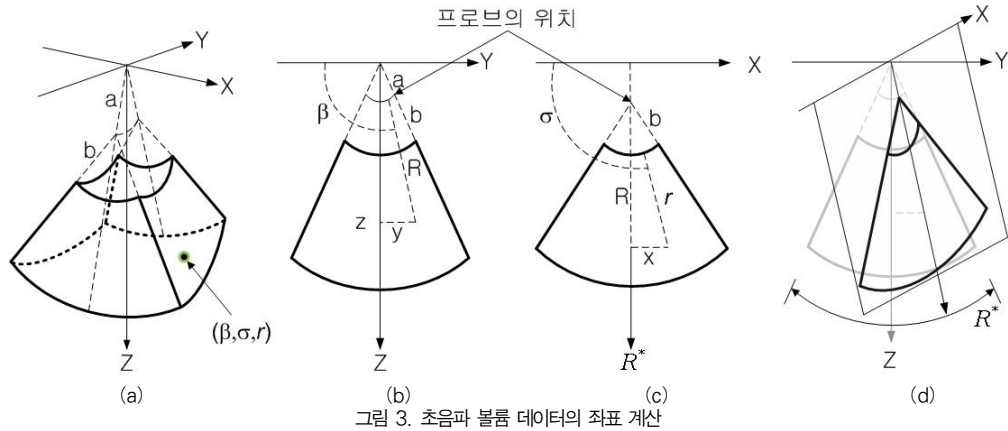


그림 3. 초음파 볼륨 데이터의 좌표 계산
Fig. 3. Coordinate Calculation of Ultrasound Volume Data

적재 될 수 있는 크기의 블럭들로 나눈다. 블럭들은 3차원 텍스처로 메인 메모리상에 저장되어 있다가, 누적 순서대로 GPU 메모리에 전송되어 렌더링된다 [18][19][20].

재샘플링 필터로 삼선형 보간법을 사용할 경우, 8개의 복셀이 필요하지만 블럭의 경계에서는 참조될 일부 복셀들이 다른 블럭에 위치하기 때문에 틀린 값이 나온다. GPU 메모리에는 한번에 하나의 블럭만 적재되기 때문에, 이웃 블럭에 있는 복셀값은 참조할 수 없다. 따라서 블럭들의 경계에서 적어도 1개 이상의 복셀이 중복 되어야 한다 [13].

블럭을 크게 하면 한 프레임에서 처리하는 블럭의 수가 적으므로 제어비용이 적지만 캐쉬 효율은 저하된다. 블럭이 작으면 빈 공간 도약 등 가속화 기법을 적용하기 용이하지만 블럭의 수가 많아져서 제어비용이 커진다. 따라서 적절한 블럭 크기를 구하는 것이 중요하다.

Grimm은 CT와 MR 볼륨 데이터를 대상으로 블럭의 형태를 정육면체로 가정하고 실험하여 32×32×32 크기 (복셀당 16 bit)가 적합하다고 주장하였다 [14].

Gobbetti는 대용량 볼륨 데이터를 단일 패스로 렌더링하기 위해서 메인 메모리에 볼륨 데이터를 블럭들로 나누어 저장하고, 렌더링에 필요한 블럭들을 GPU메모리에 비동기적으로 전송하는 방법을 제안하였다 [21]. 블럭들을 팔진트리(octree)로 구성한 후 가시성 선별기법과 전이함수(transfer function)를 이용하여 렌더링에 필요한 블럭들을 작업 세트(working set)로 지정하고 이들의 계층구조를 인코딩한 참조 텍스처를 생성한다. 작업 세트와 참조 텍스처를 매 프레임마다 GPU메모리로 전송하여 렌더링한다.

III. 블리킹을 이용한 초음파 볼륨 데이터 렌더링

초음파 볼륨 데이터를 획득하기 위해 프로브로부터 초음파 빔을 순차적으로 송신하고 수신하여 영상을 얻는 순차 영상법을 사용한다 [22]. 그림 1은 1차원 프로브 배열을 이용하여 2차원 초음파 영상들을 만들고, 프로브를 직각방향으로 회전하여 얻은 각 단면 영상을 3차원적으로 나열하여 초음파 볼륨을 만드는 과정을 보여준다.

초음파 볼륨 데이터를 메인 메모리에 저장할 때는 용량 문제를 크게 고려할 필요가 없지만, GPU 메모리에 저장할 경우 3차원 텍스처의 최대 크기가 제한되어 있기 때문에 저장할 수 없는 경우가 발생한다. CT와 MR등의 볼륨 데이터와 비교하여 초음파 볼륨 데이터는 상대적으로 크기가 작지만, 의료 영상 획득장치의 발달에 따라 초음파 볼륨 데이터의 크기도 점차 커지고 있어 예외가 되지 않는다.

기존의 블리킹 연구는 CT와 MR 볼륨 데이터와 같이 직교 좌표공간에서 샘플링된 데이터만을 사용했다. 그러나 초음파 장치는 원환체 좌표계를 사용하기 때문에 볼륨데이터를 블럭들로 분할할 때와 광선 투사법을 적용하여 광선순회(ray traversal) 할 때 그림 3처럼 논리적 좌표공간과 물리적 좌표공간 사이의 매핑을 고려해야 한다. 효과적인 메모리 사용을 위해 블럭의 크기는 2ⁿ이 좋다고 알려져 있었으나 [14] 최근 그래픽 하드웨어의 발달로 데이터의 크기가 2ⁿ이 아닌 경우에도 처리가 가능하다.

이 논문에서 제안하는 방법에서는 실시간 렌더링을 위해 전처리 과정이 필요한 복잡한 자료구조를 사용하지 않았다 [21]. 프로브에서 데이터를 획득하면, 이 초음파 데이터는 해당하는 블릭들의 배열로 복사된다. 이때 각 블릭들은 고유한 ID를 가진다.

그림 2에 나와 있듯이 초음파 볼륨 데이터는 원환체 좌표계로 획득되어 3차원 텍스처로 메인 메모리상에 저장되지만, 관측광선이 진행하는 논리적 공간에서는 직교 좌표계를 사용한다. 따라서 물리적 공간에서 직육면체 형태로 나누어진 블릭들은 논리적 공간에서 6개의 경계면 중 Z축 방향(물리적 공간에서 γ 축 방향 경계면)의 2개의 경계면은 곡면이 된다. 하나의 관측광선이 이 곡면으로 진입한 후 다시 빠져 나갈 때 블릭1이 두 번 적재되는 문제가 발생한다.

그림 4는 동일한 블릭이 두 번 적재되는 경우를 보여준다. 초음파 볼륨 데이터가 물리적 메모리 공간에서 γ 축 방향으로 두 개의 블릭들로 나누어 졌다고 가정할 때, 관측광선이 두 블릭의 경계를 따라 진행하면서 세그먼트(a)는 블릭1, 세그먼트(b)는 블릭2, 세그먼트(c)는 다시 블릭1의 복셀값을 참조한다. GPU메모리에는 한번에 한 개의 블릭만 적재될 수 있기 때문에, 이 경우 블릭1을 두 번 적재해야 한다. 이로 인해 메인 메모리와 GPU 메모리간의 불필요한 데이터이동이 발생하여 렌더링 속도가 저하된다. 이러한 문제를 해결하기 위하여 관측광선이 곡면경계를 지날 경우 한 개의 블릭만 참조하도록 크기를 조정해야 한다.

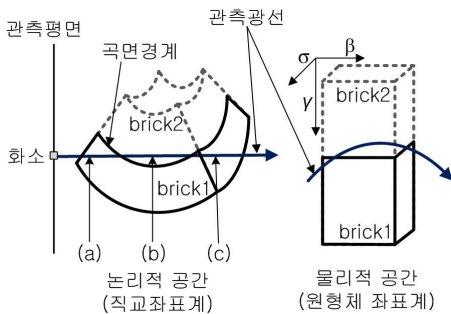


그림 4. 논리적 공간과 물리적 메모리 공간에서 관측광선의 진행경로 비교
 Fig. 4. Comparison of Ray Progress with Logical Space and Physical Space

블릭의 크기를 정하기 위하여 값 그림 5의 K_{yz} 와 K_{xz} 를 이용한다. K_{yz} 는 YZ평면에서 초음파 볼륨 데이터의 원점과 기존 곡면경계의 거리(l)와 곡면경계의 끝점과 Y축의 거리(h)의 차이이다. K_{xz} 는 XZ평면에서 초음파 볼륨 데이터의

원점과 기존 곡면경계의 거리(l)와 곡면경계의 꼭지점과 X축 사이의 거리(h)의 차이이다. 이 값들 중 큰 값만큼 Z축 방향으로 블릭1을 연장하면 곡면영역(빗금친 부분)을 포함하도록 할 수 있다. 논리적 공간의 블릭 형태를 K만큼 확장하기 위해 필요한 데이터는 잘린 구 형태로 그림 5의 빗금친 부분이지만, 물리적 메모리에 3차원 텍스처는 곡면으로 정할 수 없으므로 실제로는 그림 5의 회색 부분까지 포함된다. 빗금친 부분은 관측광선이 지나면 동일한 블릭이 두 번 적재될 수 있는 영역이며, 기존 곡면경계는 두 블릭을 물리적 공간에서 균등하게 나눈 경계면이고, 필요한 경계는 위 문제를 해결하기 위해 빗금친 부분을 인접한 두 개의 블릭들이 공유하도록 하는 경계면이며, 실제 확장된 경계는 물리적 메모리에 3차원 텍스처를 곡면으로 정할 수 없으므로 확장된 경계다.

l 과 h 가 이루는 사이각은 $\theta/2$ 이므로, K에 대해서 다음의 식이 성립한다.

$$\begin{aligned} h &= l \cos(\theta/2) \\ K &= l - h \\ h/l &= \cos(\theta/2) \dots \dots \dots (3.1) \\ K &= l - l \cos(\theta/2) \end{aligned}$$

이 영역은 논리적 공간에서 관측광선이 두 블릭의 곡면경계를 두 번 통과해서 진행될 가능성이 있는 부분이다. Z축 상에 두 점 $P_1(0, 0, l)$ 과 $P_2(0, 0, h)$ 를 메모리 공간으로 수식 2를 이용하여 좌표변환(transform)한다. 이 변환된 두 좌표의 거리(K^*)를 이용해 블릭 크기를 결정한다. 그림 5의 실제 확장된 경계까지 γ 축으로 데이터를 확장하여 블릭을 생성한다.

$$K^* = \left\| \begin{matrix} transform(Max(P1_{xz}, P1_{yz})) \\ -transform(Min(P2_{xz}, P2_{yz})) \end{matrix} \right\| \dots (3.2)$$

빈 공간 도약을 하고 관측광선의 시작점과 진행방향을 구하기 위해서, 관측광선이 진행하는 공간인 직교 좌표계에서 초음파 볼륨 데이터의 바운딩 박스(bounding box) 전면과 후면을 렌더링한다[23]. 전면을 렌더링하면 각 픽셀에서 광선의 시작점을 계산할 수 있으며 이것을 텍스처에 저장한다. 다음으로 후면을 렌더링하면서 계산된 진행방향을 텍스처에 저장한다. 앞서 계산한 텍스처와 깊이 차이를 계산하면 각 픽셀에서의 광선의 진행방향을 계산할 수 있다.

본 연구에서는 front-to-back 방식의 볼륨 광선 투사법을 사용한다. 즉 관측평면과 가까운 블릭들부터 차례대로 GPU 메모리로 전송하여 렌더링하고 각각의 중간 결과영상을 프레

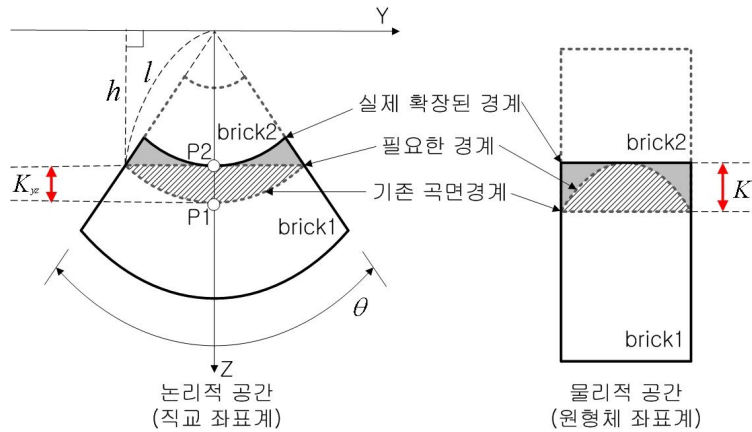


그림 5. 논리적 공간(직교 좌표계)의 YZ평면(왼쪽)과 대응하는 물리적 메모리 공간(오른쪽)
 Fig. 5. YZ Plane of Logical Space(left) correspond with Physical Space(right)

입 버퍼로 보낼 최종영상에 누적한다. 블릭들의 순서를 정하기 위하여 각 블릭마다 한 개의 점(무게중심)을 선택하여 비교한다. 메모리 공간에서 블릭들의 무게중심점을 논리적 공간으로 좌표변환하고 이점들의 깊이값을 비교한다. 깊이값이 작은 순서대로 블릭의 ID를 정렬한다. 그림 6의 경우 렌더링 순서는 b4, b5, b1, b2, b6, b3이 된다.

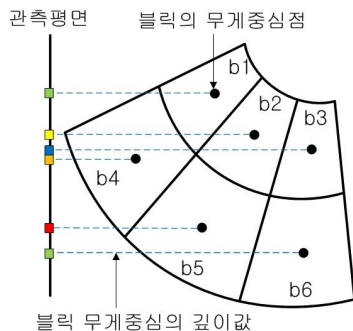


그림 6. 블릭들의 렌더링 순서를 정하기 위해 각 블릭 무게중심점의 깊이값을 비교
 Fig. 6. Depth Value Calculation of Center of Gravity Points of the Bricks to Decide Rendering Order

렌더링 순서가 정해지면, 차례대로 GPU 메모리로 전송한다. 한 블릭이 렌더링된 후 다음 블릭이 렌더링될 때 렌더링 시작점을 다시 계산하지 않도록 하기 위해 관측평면과 같은 크기의 광선 시작위치 텍스처가 필요하다 [20]. 여기에 각 블릭들이 렌더링될 때마다 광선이 진행한 거리를 기록한다. 한 개의 텍스처를 동시에 읽고 쓸 수 없기 때문에 광선 시작위치 텍스처와 광선 종료위치 텍스처를 번갈아 가며 사용하는 이중

버퍼링(double buffering) 기법을 이용한다 [23].

한 블릭이 GPU메모리로 전송되면, 각 광선들이 논리적 공간에서 진행한다. 광선의 매 샘플점에서 좌표변환을 통해 GPU메모리에 저장된 데이터를 인덱싱한다. 광선이 광선 시작위치 텍스처에 저장된 거리값을 이용하여 재샘플링을 시작한 후 샘플위치를 좌표변환하여 인덱싱한 물리적 공간의 위치가 현재 적재된 블릭의 영역을 벗어나기 전까지 스칼라값과 불투명도를 누적하고 광선이 진행한 거리를 광선 종료위치 텍스처에 저장한다. 각 블릭이 렌더링될 때마다 만들어진 영상들을 프레임 버퍼에 누적한다. 다음 블릭이 적재되면 그림 7처럼 위치 텍스처는 교환되고 광선시작 위치버퍼에 저장된 거리값을 이용하여 광선을 진행하는 과정을 반복한다.

블릭들 사이에 중첩되는 부분이 있지만, 각 관측광선이 진행한 거리를 버퍼에 저장하고 있기 때문에 중첩된 데이터 영역은 한 번만 샘플링 되며, 블릭은 각각 한 번씩만 적재된다. 각 광선의 진행이 두 버퍼를 이용해 선형적으로 이루어지기 때문에 영상의 누적순서가 적절하게 유지된다.

IV. 실험결과

실험을 위해 INTEL사의 듀얼 코어 프로세서 6400(2.13 GHz)과 2GB의 주 메모리를 가진 PC를 이용했다. 그래픽 가속기는 NVIDIA사의 8800GTX (768MByte 텍스처 메모리)를 사용하였고, DirectX 9.0c와 셰이더 버전 3.0 HLSL을 사용하였다.

실험에서 사용한 데이터는 815.36MByte(32 bit크기의 복셀, X축 1280개, Y축490개, Z축325개)의 초음파 불투명 데이터이고 결과영상의 크기는 375×375 이다. 이 데이터는 초음파

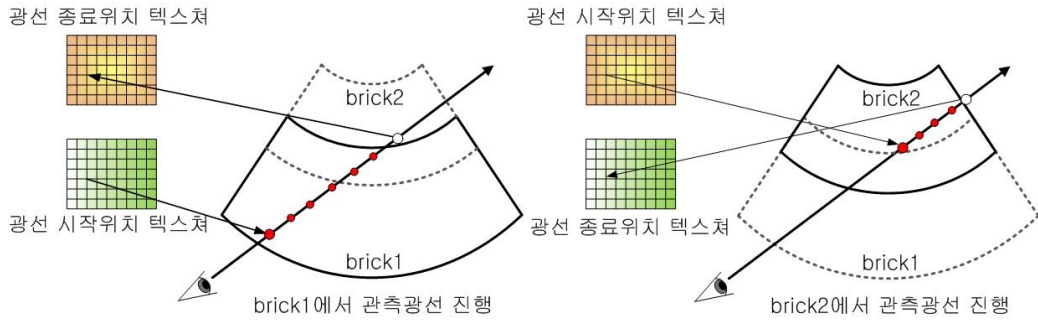


그림 7. 광선 시작(종료)위치 텍스처를 이용한 블릭 렌더링
 Fig. 7. Brick Rendering using Position Texture of Ray Start(End)

볼륨 렌더링의 결과를 검증할 수 있도록 만들어진 표준 태아 모델이며 실제 태아와 비슷한 모형을 초음파로 촬영한 후, 볼륨 데이터로 재구성한 것이다. 스캔 관측 각도의 꼭지점에서 프로브 관측 각도의 꼭지점까지의 거리(그림 3의 a)의 값은 70.04mm이고 프로브 관측 각도의 꼭지점에서 3차원 초음파 영상이 획득되는 부분까지의 거리(그림 3의 b)의 값은 338.11mm이다. 스캔 관측 각도(그림 3의 β)의 범위는 57.29° 이고 프로브 관측 각도(그림 3의 σ)의 범위는 103.64° 이다.

그림 8의 왼쪽은 중첩 영역을 설정하지 않고 블릭들을 한 번씩 적재 하는 일반적인 볼륨 렌더링을 할 때 꼭면경계를 지나는 관측광선들이 잘못된 순서로 샘플링한 결과영상을 보여준다. 관측광선이 꼭면경계를 진행할 때 먼저 적재된 블릭2에서 샘플값들이 누적된 후 블릭1이 적재되어 샘플점들의 값이 잘못된 순서로 누적되었다. 기존 방법으로 이 문제를 해결하기 위해서는 일부 블릭들을 두 번 적재해야 하며 이 때문에 속도가 저하된다.

그림 9는 블릭의 크기를 늘려서 한 번씩만 적재하도록 하는 방법과 기존 방법을 팬텀 데이터에 적용하여 실험한 결과이다. 제안한 방법은 기존 방법보다 약 10%의 속도향상이 있었다. 이것은 기존방법에 비해 제안한 방법은 적재되는 블릭

의 수가 적으므로 누적순서 계산, 텍스처 패치, 관측광선 위치 계산 등 제어비용이 적기 때문이다. 블릭의 크기가 줄어들수록 속도가 증가하다가 일정지점 이후로는 감소하였다. 이것은 블릭의 크기가 작으면 데이터의 지역성이 높아지지만 제어비용은 증가하기 때문이다. 따라서 크기, 형태와 중첩 비율을 고려하여 적당한 블릭의 개수를 찾는 것이 중요하다. 우리의 실험에서 블릭의 크기 162 × 122 × 640(개수 2 × 4 × 2, 중첩 비율 0.63%)일 때 최적의 결과를 보였다.

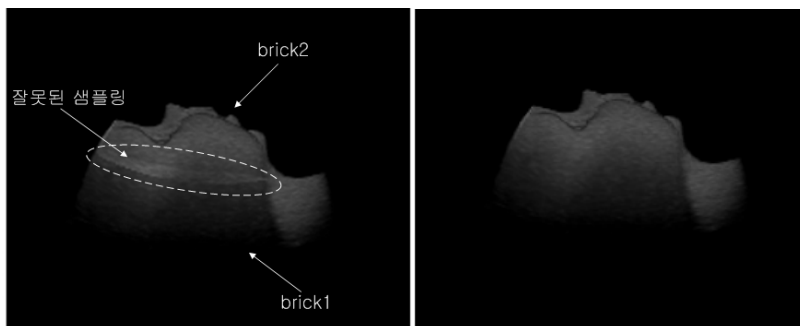


그림 8. 잘못된 순서로 샘플링 된 결과(왼쪽)와 본 논문에서 제안한 방법을 이용한 결과(오른쪽)
 Fig. 8. Result of Wrong Sampling Order(left) and Correct Result by our Method(right)

표 1. 블릭의 분할 방식에 따른 렌더링 속도
Table 1. Result of Rendering Speed by Brick Division Method

블릭 개수			블릭 크기	기존 방법	제안 방법	
β	σ	γ	(Voxel)	FPS	중첩 비율	FPS
2	2	1	162 × 245 × 1280	13.99	0%	13.97
2	1	2	162 × 490 × 640	11.57	7.58%	1.30
1	2	2	325 × 245 × 640	11.64	2.03%	13.27
1	1	4	325 × 490 × 320	8.35	30.23%	0.63
2	2	2	162 × 245 × 640	15.49	2.03%	16.61
2	2	4	162 × 245 × 320	17.66	7.97%	0.99
2	4	2	162 × 122 × 640	20.88	0.63%	22.87
4	2	2	81 × 245 × 640	1.07	2.03%	1.05
2	2	8	162 × 245 × 160	17.62	20.78%	1.10
2	8	2	162 × 61 × 640	19.92	0.63%	21.46
8	2	2	40 × 245 × 640	1.57	2.03%	1.50
4	4	4	81 × 61 × 160	3.19	2.11%	2.73
2	8	4	162 × 61 × 320	15.43	2.58%	17.93
2	32	2	162 × 15 × 640	10.49	0.63%	11.77
4	4	8	81 × 122 × 160	2.94	5.47%	3.02
4	4	16	81 × 122 × 80	4.22	12.89%	1.98
8	8	4	40 × 61 × 320	4.97	0.70%	3.92

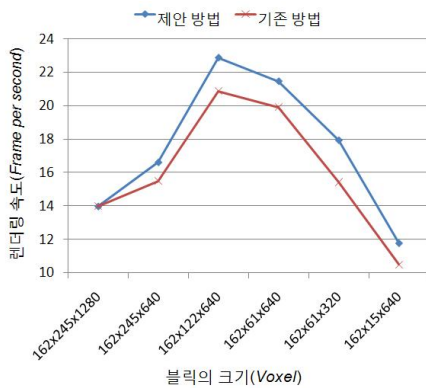


그림 9. 블릭의 크기에 따른 렌더링 속도
Fig. 9. Rendering Speed vs. Brick Size

블릭의 크기가 같더라도 분할하는 방법에 따라 속도의 변화가 있었다. 대체로 β 와 γ 방향보다는 σ 방향으로 분할할 때 속도가 향상되었다. 표 1의 회색부분들은 10 fps 이상의 속도 구간이다. 이들의 중첩 비율은 3%미만이며 블릭의 분할 개수를 정할 때 중첩비율이 3%가 넘지 않도록 해야 한다. γ 방향을 기반으로 분할하면 중첩 비율이 증가하지만 β , σ 방향

으로 분할하면 중첩 비율을 감소시킬 수 있다.

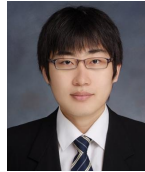
V. 결론

실시간에 대용량 볼륨 데이터를 렌더링하기 위해서 그래픽 하드웨어를 이용한다. 그래픽 하드웨어는 제한된 용량의 메모리를 가지며 이러한 문제로 볼륨 데이터를 작은 블릭들로 나누어 렌더링하는 블리킹 방법을 사용한다. 본 논문에서는 대용량 초음파 볼륨 데이터를 렌더링하기 위한 효율적인 블릭 생성 방법을 제안하였다. 초음파 볼륨 데이터는 원환체 좌표계를 사용하기 때문에 직교 좌표계에서 얻은 볼륨 데이터를 처리하기 위한 블리킹방식을 적용할 수 없다. 따라서 블릭을 생성할 때 두 블릭의 곡면경계를 포함하는 크기의 중첩 영역을 설정하여 각 블릭을 한 번만 적재하였다. 제안한 방법은 기존 방법과 비교하여 약 10%정도의 속도향상을 보였다. 향후에는 대용량 초음파 볼륨 가시화의 속도를 더욱 향상시키기 위하여 최소, 최대 블릭을 이용한 빈 공간 도약 등의 가속화 기법을 적용해야 할 것이다. 본 논문의 방법을 응용하면 다양한 비구조적 대용량 볼륨의 가시화에도 적용할 수 있을 것이다.

참고문헌

- [1] S. W. Smith, G. E. Trahey, O. T. von Ramm, "High-speed ultrasound volumetric imaging system—Part I: Transducer design and beam steering," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 38, no. 2, pp. 100 - 108, 1991.
- [2] S. W. Smith, G. E. Trahey, and O. T. von Ramm, "Twodimensional arrays for medical ultrasound," in *Proc. IEEE Ultrason. Symp.*, pp. 625 - 628, 1991.
- [3] O. T. von Ramm, S. W. Smith, and H. G. Pavy, Jr., "Highspeed ultrasound volumetric imaging system—Part II: Parallel processing and image display," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 38, no. 2, pp. 109 - 115, 1991.
- [4] 권의철, "진단용 3차원 초음파 영상 장치 및 임상적 유용성," *정보과학회지*, 제23권, 제10호, pp. 38-48, 2005.
- [5] K. Kwon, "A Study on Acceleration of Rendering using Ultrasound Volume Data," PhD thesis, Inha University, 2007.
- [6] J. Kuo, G. R. Bredthauer, J. B. Castellucci, O. T. Von Ramm, "Interactive volume rendering of real-time three-dimensional ultrasound images," *Ultrasonics, Ferroelectrics and Frequency Control*, *IEEE Transactions on*, vol. 54, pp. 313-318, 2007.
- [7] K. Engel, M. Hadwiger, J. M. Kniss, C. Rezk-Salama, D. Weiskopf, "Real-Time Volume Graphics," pp. 441-459, A.K. Peters, 2006.
- [8] W. Li, A. Kaufman, "Texture partitioning and packing for accelerating texture-based volume rendering," *Graphics Interface*, pp. 81-88, 2003.
- [9] J. Schneider, R. Westermann, "Compression Domain Volume Rendering," In *Proceedings of IEEE Visualization*, pp. 168-175, 2003.
- [10] M. Weiler, R. Westermann, Charles Hansen, K. Zimmerman, and Thomas Ertl, "Level-Of-Detail Volume Rendering via 3D Textures," In *Proceedings of IEEE Symposium on Visualization*, pp. 7-13, 2000.
- [11] M. Strengert, M. Magallon, D. Weiskopf, S. Guthe, T. Ertl, "Hierarchical Visualization and Compression of Large Volume Datasets Using GPU Clusters," In *Eurographics Symposium on Parallel Graphics and Visualization*, pp. 41-48, 2004.
- [12] N. Fout, K. Ma, "Transform Coding for Hardware-accelerated Volume Rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, 2007.
- [13] S. Grimm, "Real-Time Mono- and Multi-Volume Rendering of Large Medical Datasets on Standard PC Hardware," PhD thesis, Vienna University of Technology, 2005.
- [14] S. Grimm, S. Bruckner, A. Kanitsar, E. Gr öller, "A refined data addressing and processing scheme to accelerate volume raycasting," *Computers and Graphics*, vol. 28, no. 5, pp. 719-729, 2004.
- [15] M. Levoy, "Efficient ray tracing of volume data," *ACM Transactions on Graphics*, vol. 9, no. 3, pp. 245-261, 1990.
- [16] T. Elvins, "A survey of algorithms for volume visualization," *Computer Graphics*, vol. 26, no. 3, pp. 194-201, 1992.
- [17] S. Rottger, S. Guthe, D. Weiskopf, T. Ertl, "Smart Hardware-Accelerated Volume Rendering," In *Proceedings of EG/IEEE TCVG Symposium on visualization VisSym*, pp. 231-238, 2003.
- [18] S. Guthe, M. Wand, J. Gonser, W. Stra ßer, "Interactive rendering of large volume data sets," In *Proceedings of IEEE Visualization*, pp. 53 - 60, 2002.
- [19] D. Weiskopf, M. Weiler, T. Ertl, "Maintaining constant frame rates in 3D texture-based volume rendering," In *Proceedings of IEEE Computer Graphics International*, pp. 604 - 607, 2004.
- [20] P. Kohlmann, S. Bruckner, A. Kanitsar, M. E.

- Groller, "Evaluation of a bricked Volume Layout for a Medical Workstation based on Java," Journal of WSCG, vol. 15, no. 1, pp. 89-96, 2007.
- [21] E. Gobbetti, F. Marton, J. A. I. Guitián, "A single-pass GPU ray casting framework for interactive out-of-core rendering of massive volumetric datasets," Visual Computer, vol. 24, pp. 797-806, 2008.
- [22] O. Ssamu, "3D 초음파," Japan Society of Ultrasonics in Medicine 2005.
- [23] J. Krueger, R. Westermann, "Acceleration techniques for GPU-based volume rendering," Proceedings of IEEE Visualization, pp. 38-46, 2003.
- [24] 이중연, "GPU 클러스터를 이용한 VKH 데이터의 빠른 볼륨 렌더링," 한국정보과학회 학술지, 제23권, 제2호, pp. 763-765, 2005.
- [25] 박혜진, 송수민, 김명희, "고속 초음파 태아영상 볼륨 가시화를 위한 공간도약 거리변환 비교," 한국시물레이션학회 논문지, 제16권, 제3호, pp. 57-63, 2007.



김 주 환

2007년 인하대학교 컴퓨터정보공학과
학사
2007년~현재 인하대학교 컴퓨터정보
공학과 석사과정
관심분야: 볼륨 그래픽스, 의료 영상



권 구 주

1999년 우석대학교 컴퓨터공학과 학사
2001년 인하대학교 컴퓨터정보공학과
석사
2007년 인하대학교 컴퓨터정보공학과
박사
2007년~현재 인하대학교 컴퓨터정보
공학과 강의전임교수
관심분야: 실시간 렌더링, 의료 영상



신 병 석

1990년 서울대학교 컴퓨터공학과 학사
1992년 서울대학교 컴퓨터공학과 석사
1997년 서울대학교 컴퓨터공학과 박사
2000년~현재 인하대학교 컴퓨터정보
공학과 부교수
관심분야: 실시간 렌더링, 볼륨 그래픽스, 의료 영상

저 자 소 개