

3차원 의학영상 처리에서 사용되는 비균등 볼륨 데이터를 위한 효율적인 렌더링 방법

김태영¹ · 신병석²

¹서경대학교 컴퓨터공학과, ²인하대학교 컴퓨터공학과
(2002년 8월 19일 접수, 2002년 12월 17일 채택)

An Efficient Rendering Method for Anisotropic Volume Data in 3D Medical Imaging

Tae-Young Kim¹ and Byeong-Seok Shin²

¹Department of Computer engineering, Seokyeong University, ²Department of Computer engineering, Inha University
(Received August 19, 2002. Accepted December 17, 2002)

요약 : 3차원 의학영상처리 분야에서는 단면 해상도가 종축 방향의 공간 해상도보다 높은 비균등한 볼륨 데이터가 많이 사용된다. 본 논문은 비균등한 볼륨 데이터를 효율적으로 렌더링하기 위하여 인접한 두 슬라이스상에 있는 복셀에서의 광도를 보간하여 중간 복셀의 광도를 직접 계산하는 방법을 제안한다. 기존 렌더링 방법과 달리 이 방법은 중간 슬라이스를 만들기 위해 전처리과정을 거치지 않으며, 생성된 중간 슬라이스를 저장하기 위한 기억공간도 필요 없다. 이 방법을 고속 볼륨 렌더링 알고리즘인 쉬어왓 분해 알고리즘에 적용한 사례를 통해 제안된 방법이 기존 방법보다 화질상의 손상이 없이 속도가 향상됨을 보인다.

Abstract : In medical visualization fields, anisotropic volume data are more common than isotropic ones. In this paper, we propose an efficient rendering method for anisotropic volume data, which directly computes the intensity of intermediate samples by interpolating the intensity of two corresponding voxels on consecutive slices. Unlike conventional rendering method, it does not require a preprocessing step for generating intermediate slices or additional memory for storing them. In order to evaluate the validity and performance of our method, we applied the method to shear-warp rendering algorithm. Experimental results show that this method improves rendering speed without significantly sacrificing the image quality.

Key words : 의학영상, 비균등 볼륨 데이터, 쉬어왓 분해 알고리즘, 볼륨 렌더링

서 론

볼륨 렌더링은 일련의 단면 영상 정보로 구성된 볼륨 데이터로부터 의미있는 정보를 추출하여 3차원 영상으로 가시화시켜주는 방법으로 3차원 데이터에 대한 효과적인 분석과 통찰력을 제공한다. 볼륨 렌더링을 수행할 때 사용되는 데이터는 크게 균등한(isotropic) 볼륨 데이터와 비균등한(anisotropic) 볼륨 데이터로 구분할 수 있다(그림1)[1]. 균등한 볼륨 데이터는 공간상에 세 직교축을 따라서 대상 물체를 동일한 간격으로 샘플링하여 만든 것이다. 실제 응용에서 샘플링 결과로 만들어

진 볼륨 데이터가 균등한 경우는 드물지만, 관측방향과 무관하게 동일한 방법으로 볼륨을 처리할 수 있기 때문에 지금까지 고안된 대부분의 볼륨 렌더링 방법들은 균등한 볼륨 데이터를 가시화하는데 집중되었다.[2-6] 반면, 비균등한 볼륨 데이터는 세 직교축에서의 공간 해상도가 동일하지 않은 것으로서 균등한 볼륨 데이터보다 더 일반적이고 실제적이다. 특히 3차원 영상 기반의 진료나 수술 계획 시스템과 같은 의학영상 분야에서는 단면 해상도가 종축방향(transaxial)의 공간 해상도보다 높은 비균등 볼륨 데이터가 일반적이다. (그림 1(c))

의학영상 분야에서 비균등 볼륨 데이터가 생성되는 이유는 인체에 조사할 수 있는 방사선량은 제한되어 있는데 반해 스캐닝 장비의 처리 속도가 느리고, 영상의 픽셀 간격이 일반적으로 0.5 mm 이하인데 반해 이제까지 사용된 단일 슬라이스 CT 기술(Single slice CT technology)로는 기껏해야 1 mm

슬라이스 간격까지만 스캐닝이 가능했기 때문이다. 최근에는 한번에 여러 개의 슬라이스 탐지기(detector)를 사용하여 고속으로 고화질의 슬라이스를 스캐닝할 수 있는 다중 슬라이스 CT 기술(Multislice CT technology)이 등장하였다 [7]. 이 기술은 최소 0.5 mm 슬라이스 간격으로 스캐닝이 가능하므로 거의 균등한 볼륨을 재구성(reconstruction)할 수 있게 되었다. 그러나 이 장비의 가격이 매우 비싸 대중적으로 보급되지 못하고 있으며, 사용한다고 해도 슬라이스의 FOV(Field Of View)가 작아지면 픽셀 간격이 0.5 mm 이하가 되어 결국 비균등한 볼륨 데이터가 생성되게 된다.

이와같은 비균등한 볼륨 데이터를 렌더링하기 위하여 제시된 기존 연구 방법은 결여된 슬라이스를 보간하여 균등한 볼륨 데이터로 구성한 후, 렌더링을 수행하는 것이다. 보간하는 방법은 선형 보간(linear interpolation), 큐빅 보간(cubic convolution), 가우시안(gaussian) 등의 표준 커널들 [8]을 사용하거나, 형태 기반 보간 기법(shape-based interpolation) [9]을 사용하여 수행할 수 있다. 일반적으로 보다 정확한 보간 계산을 수행하기 위해서는 보간하는데 많은 시간이 소요되므로 응용 분야의 특성에 따라 적절한 보간 방법의 선택이 필요하다.

보간은 전처리 과정으로 수행될 수도 있고 렌더링할 때 수행될 수도 있다. 첫째, 전처리로 볼륨 데이터의 밀도값을 보간하는 방법[6, 10, 11]은 원본인 비균등 볼륨 데이터를 메모리에 적재한 다음 보간 계산을 거쳐 균등한 볼륨 데이터를 재구성하므로 첫번째 렌더링 결과가 나오기까지 매우 오랜 시간을 기다려야 하는 단점이 있다. 또한 재구성된 볼륨 데이터의 크기가 원본 데이터에 비하여 증가될 뿐 아니라, 렌더링 할 때 볼륨 데이터의 샘플을 처리하기 위한 메모리 접근 시간(memory access time)이 증가되어 결과적으로 렌더링의 성능이 저하된다. 이와 같은 문제점은 픽셀 간격 대비 슬라이스 간격이 크면 클수록 더 심각하게 나타난다. 둘째, 렌더링할 때 밀도값을 보간하는 방법 [12]은 부가적인 메모리를 필요로 하지 않으나 렌더링을 수행할 때 보간 계산이 수행되어야 하므로 렌더링 시간이 길어진다.

본 논문은 비균등한 볼륨 데이터를 효율적으로 렌더링하기

위하여 기존 연구에서 적용한 전처리 또는 렌더링 시간에 밀도값을 보간하는 방법 대신에 인접한 두 원본 슬라이스의 샘플 지점에서 광도를 계산한 후 중간 샘플에서의 광도는 두 값을 보간하여 구하는 방법을 제안한다. 이 방법은 렌더링 도중에 중간 복셀의 광도를 계산하므로 보간된 슬라이스를 저장하기 위한 별도의 기억공간을 필요로 하지 않고 중간 복셀에 대한 밀도값 보간과 법선벡터 계산이 필요 없으므로 실행 속도가 빠른 장점이 있다. 또한 본 방법을 가장 빠른 볼륨 렌더링 알고리즘으로 알려진 쉬어웁 분해 알고리즘[6]에 적용한 사례를 통해 제안된 방법의 유효성과 성능을 평가한다. 실험 결과 제안한 방법은 보간된 슬라이스를 저장하기 위한 메모리가 필요없고 결과 영상도 다른 방법을 사용한 경우에 비해 떨어지지 않으면서 속도가 향상됨을 알 수 있었다.

비균등 볼륨데이터의 렌더링 방법

1. 정의 및 기본 전제

처리의 대상이 될 비균등한 볼륨 V 는 다음과 같이 정의된다. 여기서 X 와 Y 축 방향의 복셀 간격은 동일하고, Z 축 방향의 복셀간 간격은 X, Y 축 방향 간격의 λ 배라 하자. 여기서 λ 는 1보다 큰 정수 또는 실수이다.

$$V = \{v(x, y, z) | 0 \leq x < X_{\max}, 0 \leq y < Y_{\max}, 0 \leq z < Z_{\max}\} \\ \text{where } \text{space}(Z) = \lambda \cdot \text{space}(X) = \lambda \cdot \text{space}(Y) \quad (1)$$

기존 방법은 비균등한 볼륨을 이용하여 균등한 볼륨을 재구성한다. V 로부터 생성된 균등한 볼륨 V' 는 다음과 같이 정의된다.

$$V' = \{v'(\bar{x}, \bar{y}, \bar{z}) | 0 \leq \bar{x} < X_{\max}, 0 \leq \bar{y} < Y_{\max}, 0 \leq \bar{z} < Z'_{\max}\} \\ \text{where } Z'_{\max} = \lambda \cdot (Z_{\max} - 1) + 1 \quad (2)$$

본 연구에서는 보간법으로 계산이 간단하면서도 비교적 좋은 성능을 보이는 선형 보간법을 사용하고, 경사도 계산방법으로 중심차분 연산자(central difference operator)를 사용하는 것

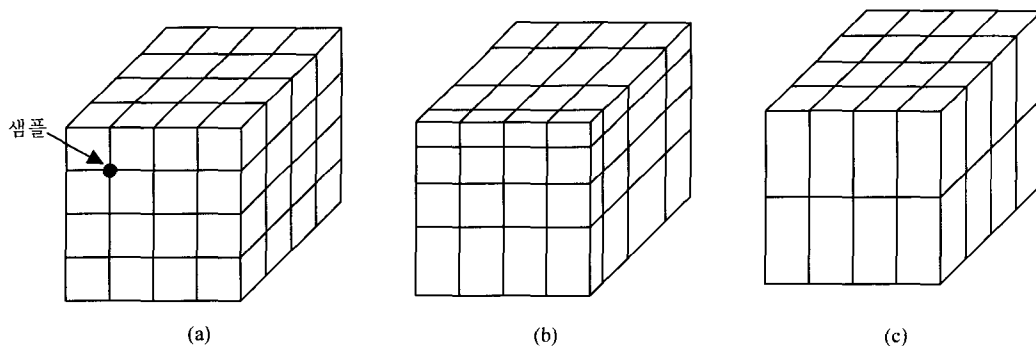


그림 1. 균등한 볼륨과 비균등한 볼륨 : (a) 균등한 볼륨 데이터, (b) 일반적인 비균등한 볼륨 데이터, (c) 의학영상에서 사용되는 비균등한 볼륨 데이터

Fig. 1. Isotropic volume and anisotropic volume : (a) Isotropic volume data, (b) General anisotropic volume data, (c) Anisotropic volume data used in medical applications

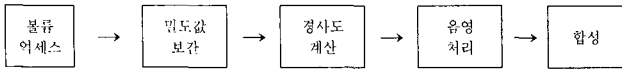


그림 2. 광선 추적법 기반의 볼륨 렌더링 과정
Fig. 2. Typical volume rendering process based on ray casting

을 전제로 한다.

2. 기존 방법과 본 연구 방법의 비교

그림 2는 일반적인 광선 투사법(ray casting)에 기반한 비균등 볼륨 데이터의 렌더링 과정을 보여준다 [10]. 볼륨 데이터가 입력되면 슬라이스 간의 대응되는 복셀의 밀도값을 보간하여 균등한 볼륨으로 만들어 준다. 그리고 광선을 일정 간격으로 진행시키면서 샘플링하여 샘플링 위치에서의 경사도를 계산한 후 경사도 벡터를 이용하여 음영 계산을 수행한다. 그리고 난 다음 이전까지 수행한 샘플링 위치에서의 광도와 불투명도를 반복적으로 합성한다.

전처리 과정에 비균등 볼륨 데이터를 균등한 볼륨 데이터로 보간하지 않고 렌더링을 수행하는 경우에는 렌더링 시에 보간이 필요한 각 샘플 위치에서 밀도값 보간과정이 필요하다. 그림 3은 기존 방법에서 보간이 필요한 위치 v_i 에서의 광도를 계산하는 과정을 보여준다. 여기서 v_i 는 인접한 두 슬라이스 상의 복셀 v_k, v_{k+1} 의 $m: (1-m)$ 내분점이다. 그림에서 $d(v_i), \vec{N}(v_i), I(v_i)$ 는 각각 v_i 의 밀도값, 법선 벡터, 그리고 광도를 나타낸다. 이 방법은 $d(v_i)$ 와 인접 복셀들의 밀도값을 보간한 후, 다음과 같이 중심 차분 경사도 연산자를 이용하여 $\vec{N}(v_i)$ 를 계산한다. 여기서 $G_x(v_i), G_y(v_i), G_z(v_i)$ 는 각각 X, Y, Z 축 방향의 경사도를 나타낸다. 여기서는 종축방향의 경사도 $G_z(v_i)$ 만 고려한다.

$$\vec{N}(v_i) = (G_x(v_i), G_y(v_i), G_z(v_i)) \quad (3)$$

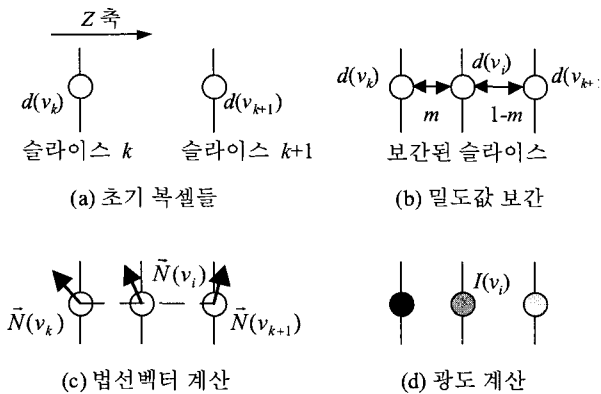


그림 3. 기존 방법을 이용할 경우 보간점에서의 광도 계산 방법
Fig. 3. Intensity value computation at an interpolated position in the conventional method

$$G_z(v_i) = \frac{\left(\left(1-m-\frac{1}{\lambda}\right)d(v_k) + \left(m+\frac{1}{\lambda}\right)d(v_{k+1}) \right) - \left(\left(1-m+\frac{1}{\lambda}\right)d(v_k) + \left(m-\frac{1}{\lambda}\right)d(v_{k+1}) \right)}{2} = \frac{(d(v_{k+1})-d(v_k))}{\lambda} \quad (4)$$

다음 과정으로 계산된 법선벡터를 이용하여 $I(v_i)$ 를 계산한다. 결과적으로 기존 방법은 모든 보간점에서 밀도값을 보간하고 법선벡터를 계산해야 광도를 구할 수 있다.

기존 방법과 달리 본 방법은 보간점의 광도를 식 (5)와 같이 v_k, v_{k+1} 의 광도 $I(v_k)$ 과 $I(v_{k+1})$ 를 보간하여 계산한다.

$$\tilde{I}(v_i) = I(v_k)(1-m) + I(v_{k+1})m \quad \text{where } 0 \leq m \leq 1 \quad (5)$$

그림 4에서 보듯이 본 방법은 보간점에 대한 밀도값과 법선 벡터를 계산하지 않고 원본 비균등 볼륨 슬라이스의 각 복셀들에 대해서만 법선벡터를 계산하여 보간점의 광도를 구한다. 또한 비균등 볼륨 슬라이스 상의 각 복셀에 대한 법선벡터 계산은 밀도값 보간 계산이 필요없으므로 계산이 간단하다. 따라서 이 방법을 이용할 경우 보간점에 해당하는 중간 복셀들을 저장하기 위한 기억장소도 필요하지 않고, 계산시간이 오래 걸리는 법선벡터 계산 횟수가 크게 감소한다. 특히 λ 가 클수록 기억장소의 절감효과와 처리속도가 크게 향상된다.

기존 방법에 의한 보간점에서의 광도와 본 논문의 방법으로 계산한 광도를 비교해 보자. 먼저 광도를 계산할때는 음영(illumination) 모델중 정반사 요소(specular component)는 고려하지 않고 배경광 요소(ambient component)와 난반사 요소(diffuse component)만 이용하는 것으로 가정한다.

$$I(v_i) = (K_a I_a + K_d (L \cdot \vec{N}(v_i)))$$

단,

K_a : 배경광 상수(ambient-reflection coefficient)

I_a : 배경광으로부터 복셀 v_i 에 반사되어 관찰자에게 도달하는 광도

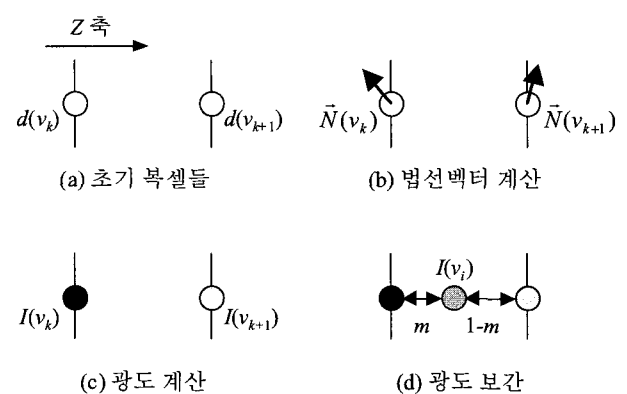


그림 4. 본 연구방법에서 제시한 보간점에서의 광도 계산 방법
Fig. 4. Intensity calculation at an interpolated position in our method

K_d : 난반사 상수(diffuse-reflection coefficient)

L : 광원에 대한 방향 벡터(light vector)

광도는 법선벡터에 의해 결정되므로 법선벡터를 계산하는 연산자의 특성을 이용하여 비교할 수 있다. 기존 방법에서 종축 방향의 경사도 벡터는 식 (4)와 같다. 본 방법에서는 법선 벡터를 보간하지 않지만, 식 (6)에서 볼 수 있듯이 v_i 의 광도는 그 점에서의 가상 법선벡터 $\tilde{N}(v_i)$ 를 계산하여 음영처리한 것과 동일하다.

$$\begin{aligned}
 \tilde{I}(v_i) &= I(v_k)(1-m) + I(v_{k+1})m \\
 &= (K_a I_a + K_d(L \cdot \tilde{N}(v_k)))(1-m) + (K_a I_a + K_d(L \cdot \tilde{N}(v_{k+1})))m \\
 &= K_a I_a(1-m) + K_a I_a m + K_d L \cdot (\tilde{N}(v_k)(1-m) + \tilde{N}(v_{k+1})m) \\
 &= K_a I_a + K_d L \cdot (\tilde{N}(v_k)(1-m) + \tilde{N}(v_{k+1})m) \\
 &= K_a I_a + K_d(L \cdot \tilde{N}(v_i))
 \end{aligned} \tag{6}$$

여기서 가상 법선벡터의 종축 방향 경사도는 식 (7)과 같다.

$$\begin{aligned}
 \tilde{G}_z(v_i) &= \tilde{N}_z(v_k)(1-m) + \tilde{N}_z(v_{k+1})m \\
 &= \frac{(d(v_{k+1}) - d(v_{k-1}))(1-m) + (d(v_{k+2}) - d(v_k))m}{2\lambda}
 \end{aligned} \tag{7}$$

식 (4)와 식 (7)을 비교하면 특정 보간점의 광도를 계산하는 연산자의 특성을 파악할 수 있다. 기존 방법은 두개의 복셀 밀도값 $d(v_k)$, $d(v_{k+1})$ 에 중심 차분 연산자를 적용하여 경사도를 계산한다. 반면에 본 방법은 이것은 종축 방향의 공간 신호에 대해서 두개의 밀도값 쌍인 $(d(v_{k-1}), d(v_{k+1}))$ 과 $(d(v_k), d(v_{k+2}))$ 에 중심 차분 연산자를 각각 적용하고 그 결과 값에 1차원 고주파 필터(low-pass filter)를 적용한 효과를 낸다. 따라서 종축 방향의 보간점 위치에 적용되는 필터 커널의 크기가 기존 방법보다 커져, 특정 위치에서의 법선벡터는 더 넓은 영역의 밀도값 차이를 반영하여 계산하게 된다. 실제 영상처리에서 이 부분은 종축방향의 비연속적 영상획득으로 인한 계단 현상을 없애고 음영처리된 면을 더 부드럽게 하는 효과를 가져온다.

```

for (k=0; k < Z'_max; k++) {
  kk ← k / λ;
  if ((k % λ) = 0) { // 원본 데이터의 복셀 위치에서의 광도 계산
    for (j=0; j < Y_max; j++) {
      for each non-transparent voxel v(i, j, kk) (0 ≤ i < X_max) {
        I_kk ← Shading(v(i, j, kk));
        project I into the intermediate image;
      }
    }
  }
  else { // 보간점에서의 광도 계산
    for (j=0; j < Y_max; j++) {
      for each non-transparent voxel v(i, j, kk) (0 ≤ i < X_max) {
        I_kk ← Shading(v(i, j, kk));
        I_kk+l ← Shading(v(i, j, kk+1));
        compute I by interpolating I_k and I_kk+l;
        project I into the intermediate image;
      }
    }
  }
}

```

그림 5. 비균등 볼륨 데이터의 중간 영상 생성 알고리즘

Fig. 5. Intermediate image generation algorithm for anisotropic volume data

쉬어왓 분해 알고리즘의 적용

$$M_{view} = M_{warp} \bullet M_{shear} \bullet M_{permute} \tag{8}$$

1. 투영 알고리즘

쉬어왓 분해 알고리즘은 식 (8)과 같이 시각 변환 행렬을 쉬어(shear) 행렬과 와핑(warping) 행렬로 분해하여 볼륨 슬라이스를 영상에 투영하는 연산을 단순화 시킨 가속화 알고리즘이다 [6]. 이 방법을 이용하면 복셀 스캔라인과 중간영상의 픽셀 스캔라인이 평행하므로 복셀에서 픽셀로의 매핑연산이 간단해 진다.

그림 5는 비균등한 원본 볼륨 데이터를 렌더링하여 중간 영상을 얻는 알고리즘을 의사 코드(pseudo code)로 나타낸 것이다. 본 방법에서는 전처리 과정으로 비균등한 볼륨을 균등하게 구성하지 않으므로 렌더링을 수행할 때 보간처리가 이루어져야 한다. 가장보간으로 균등화된 볼륨 데이터의 스캔라인 순서대로 투명하지 않은 복셀을 검색해 나가면서 만약 보간점인 경우는 인접한 두 원본 슬라이스의 각 샘플 지점의 광도를 보간하여 구하고, 그렇지 않은 경우는 원본 볼륨 데이터에서 해당 복셀의 경사도를 구한 다음 광도를 계산한다.

2. 스캔라인 버퍼링

인접한 두 슬라이스상의 동일 위치에 있는 복셀 v_1 과 v_2 사이에 존재하는 보간점들의 광도를 구하기 위하여 앞절의 알고리즘과 같이 복셀 v_1 과 v_2 의 광도를 매번 계산하는 것은 비효율적인 일이다. 대신 인접한 v_1 과 v_2 의 광도를 저장하여 재사용하면 보간 계산을 빠르게 수행할 수 있다.

본 연구에서는 효율적인 저장 방법을 사용하여 비균등한 볼륨 데이터의 각 복셀에 대한 광도를 오로지 한번만 계산하면 되도록 한다. 임의의 복셀에 대한 광도가 구해지면 그 값을 스캔라인 기반의 광도 버퍼(scanline-based intensity buffer)에 저장한다. 그리고, 복셀 사이에 있는 샘플의 광도를 구할 때는 저장된 스캔라인 단위 광도 버퍼를 이용한다.

그림 6은 광도 버퍼를 사용하여 보간점에서의 광도를 보간하는 방법을 보여준다. 그림에서 회색으로 표시된 부분이 보간되어야 하는 부분을 나타낸다. 원본 볼륨 좌표계의 각 축을 X, Y, Z 라고 하고, 중간영상 좌표계의 각 축을 I, J, K 라고 했을 때 그림 6 (a)는 주시각축이 X 축일 때($X//K$)의 광도 보간 과정을 보여준다. 여기서 가상 보간으로 균등화된 볼륨 데이터 V' 의 복셀 $v'(2, 0, 0)$ 과 $v'(2, 3, 0)$ 의 광도값은 원본인 비균등한 볼륨 데이터 V 의 복셀 $v(2, 0, 0)$ 과 $v(2, 1, 0)$ 의 광도값과 각각 일치한다. $v'(2, 1, 0)$ 과 $v'(2, 2, 0)$ 의 광도값은 원본 데이터 V 의 복셀 $v(2, 0, 0)$ 과 $v(2, 1, 0)$ 의 광도값을 선형 보간하여 구한다. 이때, 두개의 스캔라인 기반의 광도 버퍼인 전향 버퍼(front buffer)와 후향 버퍼(back buffer)를 사용한다.

V' 의 0번째 슬라이스의 스캔라인 0부터 2까지 따라가면서 투명하지 않은 복셀인 경우는 광도값을 구하여 전향 버퍼에는 V 의 0번째 스캔라인에 해당하는 광도값을 저장하고, 후향 버퍼에는 1번째 스캔라인에 해당하는 광도값을 저장한다. 렌더링을 수행하는 동안 원하는 광도값이 버퍼에 있는 경우에는 그 값을 다시 계산할 필요없이 버퍼에서 가져온다. 버퍼에 없는 경우는 광도값을 구한 후 버퍼에 저장한다. V' 의 3번째 스캔라인을 처리할 때에는 이전의 후향 버퍼의 내용을 재사용할 수 있도록 전향 버퍼와 후향 버퍼를 스와핑(swapping)한다.

그림 6 (b)와 같이 주시각축이 Y 인 경우($Y//K$), 보간 작업이 한 스캔라인 내부에서 일어난다. 따라서 버퍼를 하나만 두

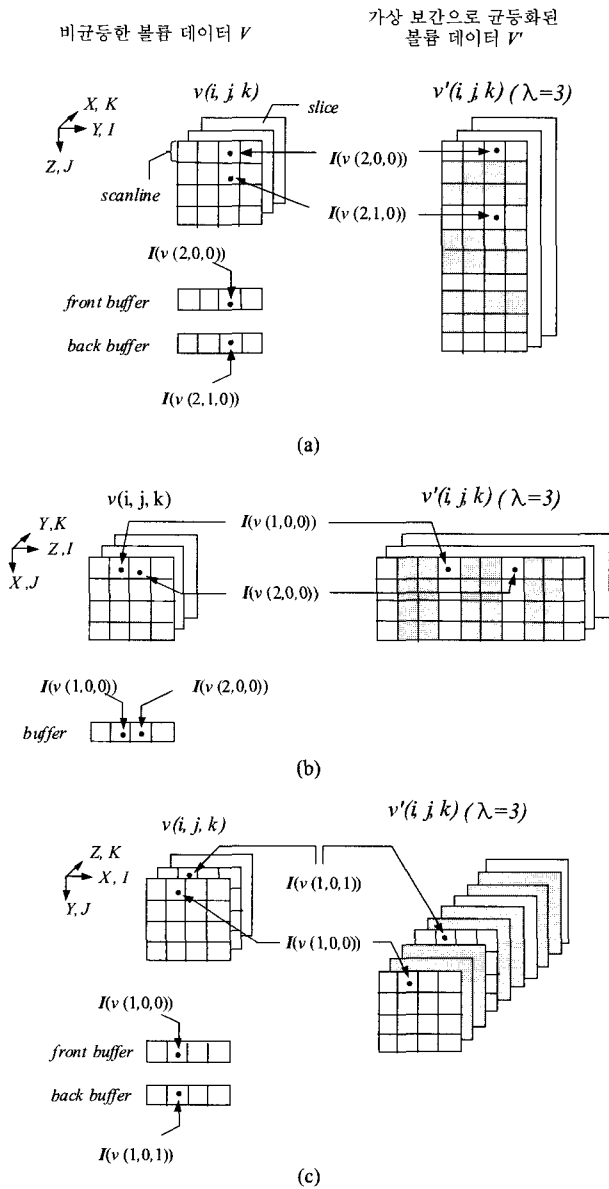


그림 6. 스캔라인 버퍼링 : (a) 주시각축이 X인 경우, (b) 주시각축이 Y인 경우, (c) 주시각축이 Z인 경우

Fig. 6. Scanline buffering : (a) Principle axis is X, (b) Principle axis is Y, (c) Principle axis is Z.

면 된다. 그림 6 (c)와 같이 주시각축이 Z 인 경우($Z//K$)는 슬라이스 사이에서 보간 작업이 발생한다. 이 경우에는 전향 버퍼와 후향 버퍼를 효율적으로 사용하기 위하여 J, K, I 순서로 스캔라인을 처리한다.

성능평가

1. 속도

(1) 기존 방법

기존 방법을 사용하는 경우의 음영처리 시간은 중간영상에 투영된 투명하지 않은 복셀의 음영처리 시간에 비례한다.

$$\text{기존 방법의 음영처리 시간} \propto c_d m^2 c_s t \quad (9)$$

- 단, c_d : 각 픽셀에 반영되는 평균 복셀 수
- m^2 : 중간 영상의 해상도
- $c_s t$: 각 샘플의 음영계산 시간

(2) 본 연구방법

본 연구에서 제시한 방법 역시 중간영상에 투영된 투명하지 않은 복셀의 음영처리 시간에 비례하나 광도 보간을 수행하므로 보간이 필요한 구간에서는 음영계산 없이 간단한 선형 보간만 수행한다. 따라서 $c_s t \gg c_i t$ 이므로 w 가 아주 크다 ($w \approx 1$)고 하더라도 본 방법이 더 빠르다. 이때, w 가 작으면 작을수록 본 논문에서 제안하는 방법이 더 빠르다. 그런데, w 는 λ 에 반비례

하므로 λ 가 크면 클수록 본 방법이 더 빠르다.

$$\text{본 방법의 음영처리 시간} \propto c_d m^2 (w c_s t + (1-w) c_i t) \quad (10)$$

단, w : 전체 샘플중 음영계산을 수행하는 샘플 비율 ($0 \leq w \leq 1$)

$c_i t$: 각 샘플의 광도 보간을 위한 선형보간 시간

2. 메모리 사용량

기존 방법을 전처리 과정으로 수행한다면 입력된 비균등한 볼륨의 공간 해상도가 $N \times N \times M (N = \lambda \cdot (M-1) + 1)$ 인 경우 필요한 메모리 소요량은 $N^2 M$ 이고 보간된 균등한 볼륨 데이터의 저장에 위한 메모리 소요량은 N^3 이다. 반면 렌더링할 때 기존 방법을 수행하거나 본 방법을 수행하는 경우 원본 볼륨 데이터만 있으면 되므로 $N^2 M$ 의 기억공간만으로 렌더링을 수행할 수 있다.

실험결과

본 실험은 그래픽 전용 하드웨어를 사용하지 않고 512 MB의 주메모리(main memory)를 가진 Pentium IV 1.2 GHz의 개인용 컴퓨터 상에서 수행되었다. 실험에 사용된 데이터는 표 1에서 보는 바와 같이 각 복셀당 2 바이트로 구성된 대용량의 비균등한 볼륨 데이터로서 Head(두부), Colon(대장), Airway

표 1. 실험에 사용된 볼륨 데이터

Table 1. Volume data sets used in our experiments

데이터	볼륨 크기	픽셀 간격 (PS)	슬라이스 간격 (SI)	$\lambda = \frac{SI}{PS}$
Head	512×512×244 (127.9MB)	0.42383	1.00000	2.35945
Colon	512×512×141 (73.9 MB)	0.62500	3.00000	4.80000
Airway	512×512×137 (71.8 MB)	0.29687	1.00000	3.36847

표 2. 속도와 메모리에 대한 성능 비교

Table 2. Comparisons of time and storage

데이터	실험	본 방법 (A)	P-Interp (B)	R-Interp (C)	A/B (%)	A/C (%)
Head	전처리 보간 시간 (ms)	-	10969	-		
	렌더링 소요시간(ms)	619	1738	1071	35.6	57.8
	볼륨 데이터 크기(MB)	127.9	301.1	127.9		
Colon	전처리 보간 시간 (ms)	-	16826	-		
	렌더링 소요시간(ms)	1212	1978	1995	61.3	60.8
	볼륨 데이터 크기(MB)	73.9	352.8	73.9		
Airway	전처리 보간 시간 (ms)	-	12584	-		
	렌더링 소요시간(ms)	908	1492	1601	60.9	56.7
	볼륨 데이터 크기(MB)	71.8	240.8	71.8		

표 3. 렌더링에 반영된 샘플수에 따른 기존 방법(R-Interp)과의 성능 비교(Colon 볼륨 데이터)

Table 3. Comparison of average rendering times between our method and R-Interp according to the number of contributed samples for Colon data sets

불투명도	샘플 수	본 방법	R-Interp	가속화 정도
1.0	0.6 M	1212	1995	1.65
0.4	8.1 M	4715	8702	1.85
0.1	13.3 M	7112	15043	2.12

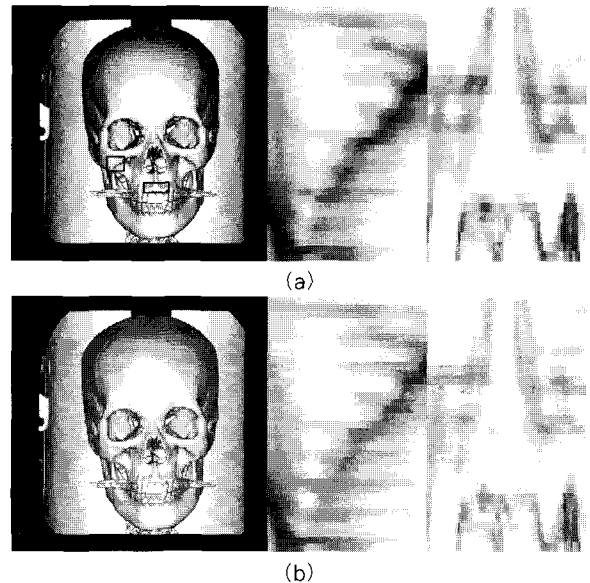
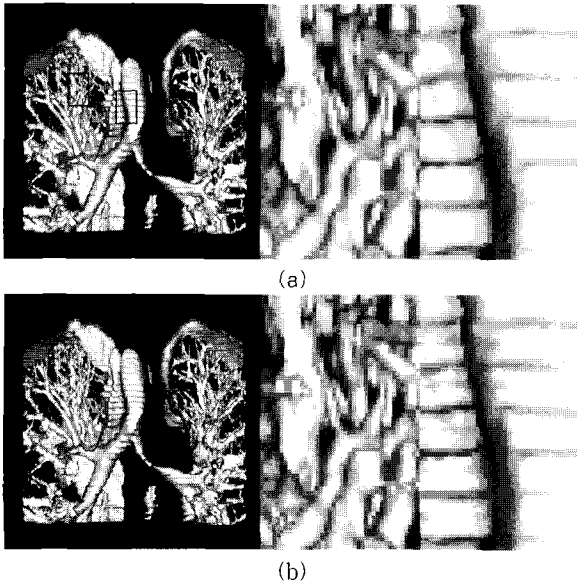


그림 7. Airway 볼륨 데이터에 대한 각 렌더링 방법의 화질 비교 : (각 방법에 대하여 중앙과 우측 그림은 고주파 영역과 저주파 영역에 대한 영상을 500% 확대한 것이다) (a) 본 방법, (b) P-interp와 R-interp

Fig. 7. Comparisons of image quality on detailed parts: (a) our method, (b) P-interp and R-Interp

(기관지) 데이터가 이에 해당된다. 이 데이터는 서울대학교 진단방사선과에서 제공한 것으로 직접 환자의 인체를 컴퓨터 단층 촬영기로 촬영하여 얻어진 DICOM(Digital Image and Communication in Medicine) 파일이다.

본 연구에서 제시한 방법의 성능을 비교하기 위하여 실험에 사용된 기존 방법들을 아래와 같이 정의한다.

P-Interp: 전처리 시간에 보간하는 기존 방법

R-Interp: 렌더링 할 때 보간하는 기존 방법

표 2는 기존의 방법(P-Interp와 R-Interp)과 본 방법을 속도면에서 성능을 비교한 것이다. 위 방법 중 오로지 P-Interp 방법만 전처리 시간이 필요하다. 뿐만 아니라, P-Interp 방법은 늘어난 볼륨 데이터 크기로 인하여 렌더링할 때 메모리 접근 시간이 증가되어 본 방법보다 렌더링에 소요되는 시간이 오래 걸릴 것을 알 수 있다. R-Interp 방법은 렌더링 수행시 각 샘플위치에서 밀도값을 보간하고 법선벡터를 구해야 하는 부담

그림 8. Head 볼륨 데이터에 대한 각 렌더링 방법의 화질 비교 : (각 방법에 대하여 중앙과 우측 그림은 사각형으로 표시한 부분인 고주파 영역과 저주파 영역에 대한 영상을 각각 300% 확대한 것이다) (a) 본 방법, (b) P-interp와 R-interp

Fig. 8. Comparisons of image quality on detailed parts: (a) our method, (b) P-interp and R-Interp

으로 본 방법보다 렌더링 소요시간이 오래 걸린다.

표 3은 Colon 볼륨 데이터에 대하여 불투명도 전이함수의 최대 불투명도를 다양하게 줌으로써 렌더링에 필요한 샘플 수를 달리하여 본 방법과 R-Interp 방법의 렌더링 속도를 비교한 것이다. 표에서 보는 바와 같이 0.6 M개의 샘플을 필요로 하는 불투명한 장벽 (colon wall) 영상을 렌더링하는데, 본 방법은 R-Interp 방법보다 1.65 배 빠른 반면, 13.3 M의 샘플을 필요로 하는 반투명한 영상을 렌더링 하는데, 본 방법이 두 배 이상 빨랐다. 그러므로, 본 방법은 많은 샘플을 요하는 영상, 즉 총 렌더링 소요시간 중 보간 계산 시간이 차지하는 비중이 크면 클수록 더 효과적임을 알 수 있다.

그림 7과 8은 화질을 비교한 것이다. P-Interp 방법과 R-Interp 방법은 보간 연산이 수행되는 시점이 다를 뿐 같은 영상을 생성한다. 본 방법으로 생성된 영상은 P-Interp 방법과 R-Interp 방법으로 생성된 영상과 비교하여 고주파 영역의 화질이 희미해지는 현상이 없으면서, 비균등한 슬라이스 간격으로 인하여 발생하는 저주파 영역의 계단 현상이 많이 줄어들

을 볼 수 있다.

결 론

의료 진단이나 가상 내시경과 같은 응용 분야에서 3차원 의학 영상정보를 이용하려면 획득된 데이터로부터 되도록 전처리 과정을 거치지 않고 대화식으로 결과 영상을 볼 수 있도록 하는 것이 실용적 측면에서 매우 중요하다. 본 논문에서는 비균등 볼륨 데이터를 대화식으로 렌더링하기 위하여 인접한 두 원본 슬라이스 영상의 이웃하는 두 복셀 지점에서 광도를 계산한 후, 중간 샘플에서의 광도는 두 값으로부터 보간하는 효율적인 렌더링 방법을 제안하였다. 본 제안된 방법의 성능을 평가하기 위하여 쉬어왓 분해 알고리즘에 적용하여 화질의 손상없이 기존 방법보다 속도면에서 성능이 우수함을 보였다.

본 알고리즘은 비균등한 정도가 크면 클수록 기존 방법보다 속도면에서 효과적이다. 본 논문에서는 렌더링 알고리즘으로 가장 빠른 알고리즘으로 알려진 쉬어왓 분해 알고리즘을 적용하였지만 기존의 볼륨렌더링 알고리즘들에 대해서도 쉽게 적용할 수 있다.

참 고 문 헌

1. D. Speray and S. Kennon, "Volume Probes: Interactive Data Exploration on Arbitrary Grids," *Computer Graphics, Vol. 24, No. 5*, pp. 5-12, 1990
2. J. Danskin and P. Hanrahan, "Fast Algorithms for Volume Ray Tracing," *Proceedings of 1992 Workshop on Volume Visualization*, pp. 91-105, 1992
3. R. Yagel and A. Kaufman, "Template-based Volume Viewing," *Computer Graphics Forum*, pp.153-167, 1992
4. K.J. Zuiderveld, A.H. Koning, and M.A. Viergever, "Acceleration of Ray-Casting using 3D Distance Transforms," *Proceedings of Visualization in Biomedical Computing*, pp. 324-335, 1992
5. J.K. Udupa and D. Odhner, "Shell Rendering," *IEEE Computer Graphics and Applications, Vol. 13, No. 6*, pp. 58-67, 1993
6. P. Lacroute and M. Levoy, "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation," *Computer Graphics*, pp. 451-458, 1994
7. B. Marincek, P. R. Ros, M. Reiser, and M. E. Baker, "Multislice CT: A Practical Guide," *Proceedings of the 5th International SOMATOM CT Scientific User Conference*, 2000
8. G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press, 1990
9. G.T. Herman, J. Zheng, and C.A. Bucholtz, "Shape-based Interpolation," *IEEE Computer Graphics and Application, Vol. 12, No. 3*, pp. 69-79, 1992
10. M. Levoy, "Display of Surfaces from Volume Data," *IEEE Computer Graphics and Applications, Vol. 8, No. 3*, pp. 29-37, 1988
11. R. Yagel, "Volume Viewing: State of the Art Survey," *SIGGRAPH '97 Course Note 31*, 1997
12. K. Mueller, N. Shareef, J. Huang, R. Crawfis, "High-Quality Splatting on Rectilinear Grids with Efficient Culling of Occluded Voxels," *IEEE Transactions on Visualization and Computer Graphics, Vol. 5, No. 2*, pp. 116-134, 1990