

3차원 인체 해부도 작성을 위한 칼라 볼륨 데이터의 입체 영상 재구성

김보형·이철희·신영길·김종효*·강홍식*

서울대학교 전산과학과 및 서울의과대학 진단 방사선과*
(1997년 9월 2일 접수, 1998년 4월 16일 채택)

Reconstruction of Color-Volume Data for Three-Dimensional Human Anatomic Atlas

B.H. Kim, C.H. Lee, Y.G. Shin, J.H. Kim*, and H.S. Kang*

Department of Computer Science, Seoul National University
Department of Radiology, Seoul National University College of Medicine*
(Received September 2, 1997, Accepted April 16, 1998)

요약 : 본 논문에서는 전산화된 인체 해부도의 필수 기능인 3차원 볼륨 가시화 기법을 제시한다. 오브젝트-순서(object-order)에 기반한 광선 추적과 런-길이 인코딩(run-length encoding)의 장점을 이용한 이진 볼륨 렌더링 기법은 경계 추출된 칼라 슬라이스로 구성된 볼륨 데이터를 이용하여, 특정 하드웨어의 도움없이 일반 PC에서 대화식 수준의 속도로 3차원 가시화를 수행한다. 이 방법은 이진 볼륨 렌더링을 위해 필요한 이진 깊이 화상(binary depth image)의 구성을 간소화하고 새로운 법선 벡터 계산 방법을 적용하여 렌더링 시간의 향상을 꾀하였다. 이와 함께 화질의 손실 없이 관련 데이터의 양을 줄이는 3D경계 인코딩 방법도 제시한다. 본 논문에서 제시한 렌더링 방법의 대화식 수준의 속도와 인코딩 방법의 정보 저장면에서의 효율성은 PC에서 운용될 수 있는 의학 해부도 응용 프로그램의 개발을 보다 가속화할 것이다.

Abstract : In this paper, we present a 3D reconstruction method of color volume data for a computerized human atlas. Binary volume rendering which takes the advantages of object-order ray traversal and run-length encoding visualizes 3D organs at an interactive speed in a general PC without the help of specific hardwares. This rendering method improves the rendering speed by simplifying the determination of the pixel value of an intermediate depth image and applying newly developed normal vector calculation method. Moreover, we describe the 3D boundary encoding that reduces the involved data considerably without the penalty of image quality. The interactive speed of the binary rendering and the storage efficiency of 3D boundary encoding will accelerate the development of the PC-based human atlas.

Key words : Volume Rendering, Object-Order Ray Traversal, Run-Length Encoding, Binary Depth Image

서 론

볼륨 렌더링은 의학 영상이나 과학 영상 가시화분야에서 쉽게 얻을 수 있는 스칼라 값들로 이루어진 볼륨 데이터를 가시화하는 작업이다. 일반적으로 볼륨 데이터는 그 양이 많기 때문에 렌더링 시간에 제약이 많아 지금까지 다양한 가속화기법들에 대

한 연구가 많이 이루어져 왔다. 그 중 한가지는 볼륨의 6개 면들중의 하나와 평행하는 중간 투영 평면(intermediate projection plane)을 사용하는 방법이다[1-6]. 쉬어-워프 분해(shear-warp factorization)라고 불리는 이 방법에서는 중간 화상을 구성한 다음 효율적인 2차원 워핑(warping)[7]에 의해서 최종 화상을 얻는다. 또 다른 방법은 데이터 집합에서의 연관성에 대한 정보를 이용하여 렌더링을 수행함으로써 이미지 질의 손실없이 렌더링 시간을 줄이고자 하는 것이다. 이러한 방법들은 비투명 복셀(non-transparent voxel)의 존재 여부를 인코딩하는 공간 데이터 구조를 이용함으로써 투명한 영역에서의 계산 시간을 줄인다[8-12].

<속보논문>

이 연구는 보건복지부에서 주관한 선도기술 의료기공학 기술 개발사업의 지원(HMT-95-G-1-08)에 의하여 이루어진 것임
통신저자 : 김보형, (151-742) 서울시 관악구 신림동 산 56의 1 서울대 전산과학과 Tel. (02)889-1764, Fax. (02)871-4912

쉬어-웁 렌더링[4]은 위의 두 가지 방법을 결합하여 특정 하드웨어의 도움없이 대화식 수준의 가시화를 가능하게 하였다. 하지만, 쉬어-웁 렌더링[4]에서는 임의의 방향에서의 렌더링을 위해서, 각각의 가능한 주요 시각 방향(principal viewing direction)을 위한 3벌의 인코딩 데이터가 필요하다. 이러한 중복을 줄이기 위해 [13]에서는 쉬어-웁 렌더링에서와는 다른 중간 시각 좌표계(intermediate viewing coordinate)를 제안하여 한 벌의 인코딩 데이터만을 사용하여 임의의 방향에서의 가시화를 가능하게 하였다.

지금까지 의학 영상 분야에서 볼륨 가시화를 사용하는 경우는, 컴퓨터 단층 촬영기(CT: computed tomography)나 자기 공명 장치(MRI: magnetic resonance tomography)를 이용하여 인체의 장기나 골격을 촬영한 것을 3D 영상으로 재구성하여 보여줌으로써 질병 치료나 교육에 활용할 수 있도록 하는 것이다. 이러한 MRI나 CT데이터는 조직의 밀도값을 보여주는 음영 단계(gray-scale)화상으로서 물체(object)의 기하학적 특성을 쉽게 얻을 수 있다는 장점이 있긴 하지만, 해상도문제 때문에 볼륨 렌더링에 사용하기에는 어느 정도 제한이 있다. 반면, 단층 사진 촬영에 의해 얻어지는 데이터는 고해상도일뿐만 아니라 단면을 보여주는 칼라 화상이기 때문에 일반 방사선 영상 데이터보다 훨씬 정교한 해부학적 정보를 가지고 있어 보다 정확한 가시화를 가능하게 한다.

최근 들어 의학 해부도 응용 프로그램[14-17]의 필요성이 증가되면서 visible human data(NLM의 'Visible Human' 프로젝트에서 생성한 의료 데이터)의 사진 촬영 데이터를 이용하여 가시화를 수행하거나 고화질의 렌더링을 위한 경계 추출과 관련된 여러 가지 연구가 이루어지고 있다[18-19]. 하지만 이러한 연구 대부분은 대화식 수준의 렌더링을 위해 동일 표면 추출(isosurfacing)[20]과정을 거쳐 특정한 컴퓨터 모델링 소프트웨어나 하드웨어를 사용하는데, 이러한 표면 렌더링(surface rendering) 방법은 면 표현을 위한 전처리 시간이 엄청날 뿐만 아니라, 원래의 데이터가 면 근사(surface approximation)[21]에 부적합하여 상실되거나, 면으로 표현하기가 불가능한 구조물(structure)이 있다는 단점이 있다. 또 면 렌더링에서 투명 표현[22]과 같은 광학 효과를 내기 위해서는 3-D 텍스처 맵핑을 이용해야 하며 이러한 텍스처 맵핑이 가능한 장비의 비용 또한 엄청나다. 이러한 단점을 보완하기 위해 볼륨 렌더링을 이용하여 RGB 칼라 볼륨 가시화를 수행하는 경우에도 방대한 볼륨 데이터의 양 때문에 대화식 수준의 렌더링을 위해 고가의 장비에 의존하고 있어 연구 결과의 대중적인 응용이 현실적으로 어려운 실정이다.

본 연구에서는 의학 해부도 응용 프로그램의 필수 기능인 3-D 렌더링을 방사선 영상 데이터가 아닌 경계추출된 칼라 슬라이스로 구성된 볼륨 데이터를 이용하여 특정 하드웨어의 도움없이 일반 PC에서 대화식 수준의 속도로 수행할 수 있는 가시화 방법을 제안한다. 이 방법은 [4,13]에서의 시각 변환 행렬

분해에 근거하여 1벌의 인코딩 데이터만을 사용하여 임의의 방향에서의 렌더링을 수행한다. 이와 함께 가시화에 필요한 데이터의 양을 최소화하면서 이미지 질의 손실없이 렌더링을 할 수 있는 3-D인코딩 방법에 대하여도 다룬다.

본 연구에서 다루는 데이터는 visible human data중에서도 절단면의 사진 촬영에 의해 얻은 RGB 칼라 화상이다. 일반적인 볼륨 렌더링에서 다루는 데이터는 분류된 볼륨 데이터(classified volume data)-불투명도 변환 함수(opacity transfer function)를 적용하여 얻은 데이터-로서 0에서 1사이의 불투명도와 칼라값을 복셀값으로 가진다. 반면 본 연구에서와 같은 칼라 화상으로만 이루어진 볼륨 데이터의 가시화는 각 샘플 포인트가 r,g,b값 만으로만 구성되어 있기 때문에 불투명도가 0과 1로만 이루어진 볼륨을 가시화하는 이진 볼륨 렌더링이다.

2절에서는 쉬어-웁 분해에 근거하여 보다 효율적으로 이진 볼륨 렌더링을 수행하는 방법을 제시한다. 이 방법은 이진 볼륨 렌더링을 위해 필요한 이진 깊이 화상(binary depth image)의 구성을 간소화하고 새로운 법선 벡터 계산 방법을 사용하여 보다 효율적인 렌더링을 수행한다. 3절에서는 인코딩되는 데이터의 양을 현격하게 줄일 수 있는 3-D 경계 인코딩에 대하여 설명한다. 끝으로 4절에서는 실험 결과를 보여주고 5절에는 결론 및 향후 과제에 대하여 언급한다.

RGB 볼륨 데이터 이진 렌더링

일반적인 볼륨 렌더링에서는 광선(ray)을 따라가면서 그 광선과 만나는 비투명 복셀(non-transparent voxel)들의 칼라값과 불투명도값을 누적하여 최종 투영 평면에서의 음영값을 결정한다. 하지만 RGB-칼라 화상의 슬라이스로 구성된 3차원 볼륨 데이터와 같이 전처리단계에서 복셀의 법선 벡터의 값을 미리 계산할 수가 없는 경우에 사용되는 이진 볼륨 렌더링에서는, 샘플링된 볼륨 데이터에서 이진 깊이 화상(binary depth image)을 만들어내는 일이 필수적이다. 이진 깊이 화상은 3차원 데이터를 시각 방향에 수직인 평면상에 평행 투영하고, 투영된 각 화소(pixel)에는 관측점에서 보이는 복셀의 깊이값(depth value)을 부여하여 만들어진 2차원 이미지이다. 이진 깊이 화상은 관측점이나 시각 방향이 바뀔 때마다 새로이 생성되어야 한다는 문제점이 있기는 하지만 비교적 적은 데이터로 3차원 물체의 표면을 구성하는 점 정보를 쉽게 구성해낼 수 있다는 장점이 있어 입력 데이터로 많이 사용된다.

이진 깊이 화상으로부터 법선 벡터의 추정은, 주어진 점을 중심으로 하여 x 방향과 y 방향으로 2개씩 이웃한 점들을 취한 후, 그 점들간의 깊이값의 차이를 경사도의 근사값으로 이용하는 것이다. 그러면 평면상의 한점에서의 추정 법선 벡터(estimated normal vector)는 식(1)과 같이 구해진다. 식 (1)에서 $\frac{\nabla z}{\nabla x}$ 는 x 방향으로의 화소당 깊이 값의 변화량이고 $\frac{\nabla z}{\nabla y}$ 는 y 방향으로의 깊이 값의 변화량이다.

$$\vec{N}_e = \left(-\frac{\nabla z}{\nabla x}, -\frac{\nabla z}{\nabla y}, 2 \right) \quad (1)$$

식 (1)에 의해 구해진 법선 벡터와 RGB 볼륨 데이터에서의 r, g, b값을 이용하여 최종 투영 평면에서의 각 화소의 음영값을 결정할 수 있다.

칼라-RGB 이미지의 2차원 슬라이스 데이터에서 3차원 입체 영상을 얻어내는 이진 볼륨 렌더링을 위해 필요한 이진 깊이/칼라 화상의 구성은 직접 투영(direct projection)에 의해서 가능하다. 최종 투영 평면의 각 화소에서 출발하는 광선을 따라가면서 처음으로 만나는 비투명 복셀까지의 거리와 그 복셀의 칼라 값을 이용하여 가시화를 수행할 수도 있다. 하지만 이러한 이미지-순서(image-order)에 근거한 볼륨 탐색방법은 각 광선마다 전체 공간 데이터 구조를 한 번씩 살펴봐야 한다는 단점이 있다.

이 절에서는 먼저 투영과 변환(transformation)에 관한 수식을 설명하고, 런-길이 인코딩(run-length encoding)된 데이터를 쉬어-왓 분해에 근거하여 PC기반에서 대화식 수준으로 3-D칼라 영상을 구성하는 이진 볼륨 렌더링 방법에 대하여 설명하겠다.

1. 쉬어-왓 분해(shear-warp factorization)

렌더링 방법에 대한 설명을 하기 전에 투영 변환과 관련된 표기에 대하여 설명하겠다. 먼저 본문에서 사용되는 몇 가지 용어를 정의하면, 실좌표계(world coordinate)는 (u, v, w) 로 표기하고, 시각 변환 행렬(viewing matrix)은 $View$ 로 쓴다. 볼륨 데이터의 저장 방법은 하나의 슬라이스가 다 저장된 후 다음 슬라이스가 저장되며 각 슬라이스내에서는 하나의 스캔라인이 연속적으로 저장되고 다음 스캔라인이 그 뒤를 잇는다. 모든 스캔라인은 u 축과 평행하며, 슬라이스내의 모든 점은 같은 w 값을 갖는다. 이 데이터는 런-길이 인코딩되어 렌더링에 사용된다.

중간 시각 좌표계 시스템(intermediate viewing coordinate system)의 주요 좌표축은 실좌표계 시스템의 3개의 주요 좌표축과 평행하다. 중간 시각 좌표계는 (x, y, z) 로 표기하며, 중간 투영 평면은 xy 평면으로 간주한다. 쉬어-왓 분해에서는 볼륨 데이터의 6개의 면 각각이 중간 투영 평면으로 선택될 수 있다. 실좌표계에서 중간 시각 좌표계로의 변환 행렬은 M_{int} 로 표기하겠다.

시각 방향(viewing direction)을 정의하는 벡터는 $DOP = (u_{dop}, v_{dop}, w_{dop})$ 로 쓴다. 중간 투영 평면은 이 벡터의 각 요소의 절대값을 비교하여 결정이 되는데 절대값이 가장 큰 요소를 가지고 있는 축이 z 축이 된다.

중간 투영 평면으로의 투영은 다음 행렬에 의하여 수행된다.

$$M_{shear} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ Sh_x & Sh_y & 0 & 0 \\ t_{sh,x} & t_{sh,y} & 0 & 1 \end{pmatrix}$$

$$\begin{aligned} Sh_x &= -x_{dop} / z_{dop} \\ Sh_y &= -y_{dop} / z_{dop} \end{aligned}$$

$$t_{sh,x} = \begin{cases} 0 & \text{if } Sh_x \geq 0 \\ -Sh_x \times z_{max} & \text{otherwise} \end{cases}$$

$$t_{sh,y} = \begin{cases} 0 & \text{if } Sh_y \geq 0 \\ -Sh_y \times z_{max} & \text{otherwise} \end{cases}$$

$t_{sh,x}, t_{sh,y}$ 는 복셀을 중간 투영 평면에 투영시킨 후의 좌표값이 항상 양이 되도록 보장해준다.

최종 투영 평면으로의 2차원 워핑은 행렬 M_{warp} 를 따른다.

이 모든 변환을 한데 묶으면

$$M_{int} \times M_{shear} \times M_{warp} = View \quad (2)$$

식(2)의 양변에 M_{int}^{-1} 를 곱하면 $M_{shear} \times M_{warp} = M_{int}^{-1} \times View$ 이 된다. 여기서 $M_{int}^{-1} \times View$ 를 permuted viewing matrix라 하고 $View'$ 으로 쓴다.

$$View' = M_{int}^{-1} \times View = \begin{pmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ t_x & t_y & t_z & 1 \end{pmatrix} \quad (3)$$

라고 할 때

$$M_{warp} = \begin{pmatrix} a & b & 0 & 0 \\ d & e & 0 & 0 \\ 0 & 0 & 0 & 0 \\ t_x - a \times t_{sh,x} - d \times t_{sh,y} & t_y - b \times t_{sh,x} - e \times t_{sh,y} & 0 & 1 \end{pmatrix} \text{이다.}$$

2. 쉬어 왓 분해에 기반한 이진 렌더링 : 비교를 위해

이 절에서는 쉬어-왓 분해에 근거하여 칼라 볼륨 데이터의 렌더링을 수행하는 직관적인 방법에 대하여 설명하겠다. 설명의 편의를 위해 이 방법은 워핑 후 음영 처리 이진 렌더링(Post-warp shading binary rendering)이라 부른다. 워핑 후 음영 처리 이진 렌더링은 쉬어-왓의 합성(composition)단계에서 중간 깊이/칼라 화상(intermediate depth/color image)를 얻고 그것을 각각 워핑하여 최종 깊이/칼라 화상(final depth/color image)를 구성한 후 음영 처리를 수행하여 원하는 영상을 얻는다.

1) 합성(쉬어/투영) : $M_{int} \times M_{shear}$ 에 따라서 볼륨 데이터를 중간 평면에 투영하여 중간 깊이 화상과 중간 칼라 화상을 만든다.

2) 워핑 : M_{warp} 에 따라서 중간 화상들을 워핑시켜 최종 깊이 화상과 최종 칼라 화상을 얻는다.

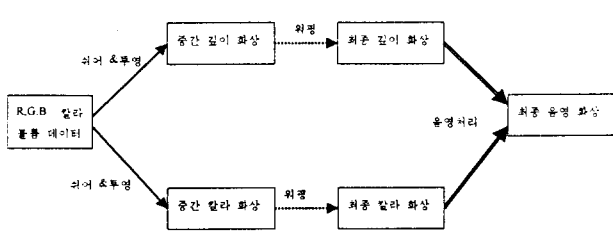


그림 1. 워핑 후 음영처리 이진 렌더링
Fig. 1. Post-warp shading binary rendering

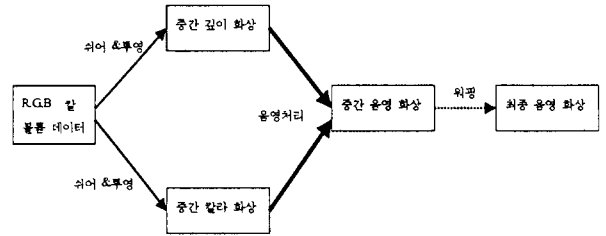


그림 3. 워핑 전 음영 처리 이진 렌더링
Fig. 3. Pre-warp shading binary rendering

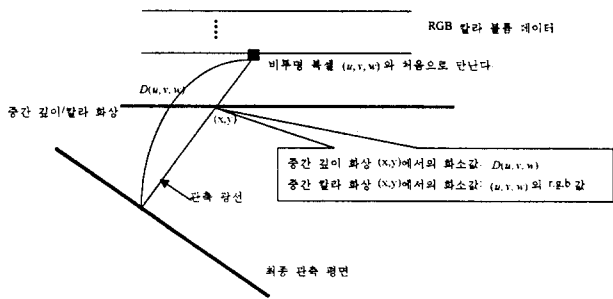


그림 2. 중간 깊이/칼라 화상의 구성 : 주어진 화소를 통과하는 관속 광선과 처음으로 만나는 비투명 복셀까지의 거리와 r, g, b값을 중간/깊이 화상의 화소값으로 부여한다.
Fig. 2. Construction of intermediate depth/color image

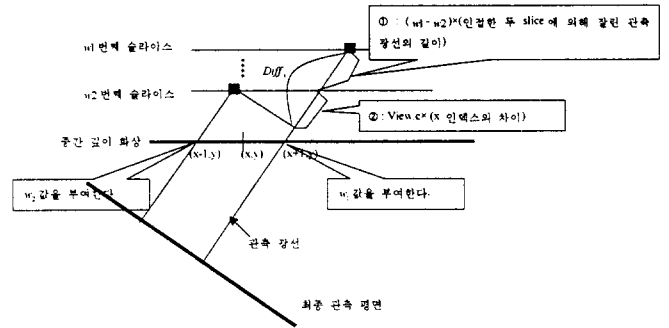


그림 4. 중간 깊이 화상의 구성
Fig. 4. Construction of intermediate depth image

3) 음영처리 : 2)에서 얻어진 화상들을 사용하여 최종 음영 화상(final shade image)을 구성한다. 위 과정을 그림으로 나타내면 그림 1과 같다.

중간 깊이 화상의 주어진 화소에서의 깊이 값은, 최종 투영 평면으로부터 그 화소를 통과하는 광선을 따라가면서 가장 처음으로 만나게 되는 비투명 복셀까지의 거리를 저장하고 있으며 중간 칼라 화상은 그 복셀의 r, g, b값을 화소값으로 가진다(그림 2). 처음으로 만나는 비투명 복셀의 볼륨 좌표계상의 좌표값이 (u, v, w) 일 때 이 복셀의 최종 투영 평면으로부터의 거리를 $D(u, v, w)$ 라 하면 그 값은 투영 행렬에 의해 아래와 같이 계산된다.

$$D(u, v, w) = View.c \times u + View.f \times v + View.i \times w + View.t_z \quad (4)$$

두번째 단계인 워핑단계에서는 첫번째 단계에 의하여 만들어진 두 개의 중간 화상들에 효율적인 2차원 워핑[7]을 적용하여 최종 깊이/칼라 화상을 구성한다.

세번째 음영 처리단계에서는 최종 깊이 화상에서 인접한 두 화소에서의 깊이 값의 뺄셈을 통하여 x방향 경사도와 y방향 경사도를 구해 법선 벡터를 추정(식(1))하고, 최종 칼라 화상의 r,g,b값을 이용하여 음영 처리를 수행한다. 이 방법에서는 깊이 화상의 하나의 화소값을 결정하기 위해서 3번의 부동 소수점(floating-point)곱셈과 3번의 부동 소수점 덧셈이 필요(식(4))하고, 최종 음영 화상을 얻기 위해서 2번의 워핑과 1번의 음영

처리가 필요(그림 1)하다.

3. 새로운 법선 벡터 계산 방법을 이용한 워핑 전 음영처리 이진 렌더링

이 절에서 제시하는 방법은 중간 깊이 화상의 화소값 결정을 간소화하고 중간 깊이/칼라 화상에서 음영 처리를 수행할 수 있는 새로운 법선 벡터 계산법을 적용하여 중간 음영 화상을 구성한 후, 워핑에 의해 최종 영상을 얻는다. 설명의 편의를 위해 이 방법은 워핑 전 음영 처리 이진 렌더링(*Pre-warp shading binary rendering*)이라 부른다.

- 1) 합성 : $M_{int} \times M_{shear}$ 에 따라서 볼륨 데이터를 중간 화상에 투영하여 중간 깊이 화상과 중간 칼라 화상을 만든다.(워핑 후 음영처리 이진 렌더링의 1)과 동일)
- 2) 음영처리 : 새로운 법선 벡터 계산법을 적용하여 중간 깊이 화상과 중간 칼라 화상에서 중간 음영 화상을 얻는다.
- 3) 워핑 : M_{warp} 에 따라서 중간 음영 화상을 워핑시켜 최종 음영 화상을 얻는다.

위의 볼륨 렌더링과정을 그림으로 나타내면 그림 3과 같다. 여기서는 워핑 후 음영처리 이진 렌더링과는 달리, 중간 깊이 화상의 각 화소의 값으로 최종 투영 평면에서의 처음으로 만나는 비투명 복셀까지의 거리를 저장하지 않고, 처음으로 만나는 비투명 복셀의 시각 방향과 가장 작은 각을 이루는 축(이하 "주 시각 좌표축"이라 하겠다)상의 좌표값을 저장한다(그림4). 즉, 깊이 화상의 화소값은 단 한번의 지정 연산(assignment operation)으로 결정된다. 중간 칼라 화상의 화소값으로는 워핑 후

음영처리 이진 렌더링에서와 같이 광선과 처음으로 만나는 비투명 복셀의 r, g, b값을 부여한다.

이러한 방법으로 구성된 두 개의 중간 화상을 이용하여 음영 처리를 수행하기 위해서는 중간 화상의 정보를 가지고 최종 화상에서 한 것과 같은 효과의 음영 처리를 수행할 수 있는 새로운 방법의 개발이 필요하다.

어떤 주어진 중간 화상 (x, y)에서의 음영 처리를 수행하는 과정을 그림을 이용하여 설명하도록 하겠다. (x, y)에서의 음영값을 얻기 위해서는 중간 깊이 화상의 화소값을 이용하여 x 방향 경사도 벡터와 y 방향 경사도 벡터를 구해야 한다.

우선 x방향 경사도 벡터를 구해보자. 중간 깊이 화상의 (x+1, y)에서의 깊이 값은 w₁((x+1, y)를 관통하는 광선이 처음으로 만나는 비투명 복셀의 주 시각 좌표축상의 좌표값이 w₁), (x₁, y)에서의 깊이 값이 w₂라고 가정하자. 그림 4에서 (x+1, y)에서의 최종 투영 평면으로부터의 거리와 (x-1, y)에서의 최종 투영 평면으로부터의 거리의 차이(Diff_x)는 ①+②이다. 표기를 간단히 하기 위해 (인접한 두 슬라이스에 의해 잘린 관측 광선의 길이)는 dist_{slice}라 쓴다.

$$\textcircled{1} : (w_1 - w_2) \times \text{dist}_{\text{slice}} \quad (5)$$

$$\begin{aligned} \textcircled{2} : & \text{View.c} \times (x \text{ 인덱스의 차이}) \\ & = (x+1-(x-1)) \times \text{View.c} = 2 \times \text{View.c} \quad (6) \\ \text{Diff}_x & = \textcircled{1} + \textcircled{2} \quad (7) \end{aligned}$$

식(5)는 각각 (x+1, y)와 (x-1, y)를 지나는 두 개의 광선이 처음으로 만나는 비투명 복셀의 주 시각 좌표축상의 좌표값의 차이에 의한 깊이 값의 차이이다. 중간 화상은 1)합성 단계에서 슬라이스의 쉬어/투영에 의하여 구성이 되기 때문에, 광선이 처음으로 만나는 비투명 복셀의 w값차이에 의한 깊이의 차이는 dist_{slice} × (w 값의 차이)가 되는 것이다. 여기서 dist_{slice}는 시각 방향을 정의하는 벡터 DOP(u_{dop}, v_{dop}, w_{dop})에서 쉽게 구해질 수 있다. 그 값은 DOP의 w성분이 1일 때의 DOP벡터의 절댓값이 된다. 따라서 식(5)는 다음과 같이 쓸 수 있다.

$$(w_1 - w_2) \times | | 1/w_{\text{dop}} \times (u_{\text{dop}}, v_{\text{dop}}, w_{\text{dop}}) | | \quad (8)$$

최종 투영 평면으로부터 광선과 만나는 비투명 복셀까지의 거리는 식(4)에 의하여 계산된다. 평행 투영(parallel projection)에서는 중간 화상의 화소간의 간격과 볼륨 공간상에서의 볼륨간의 간격이 서로 같기 때문에 중간 화상에서의 x좌표 차이가 2이면 볼륨 공간상에서의 u좌표값 차이도 2가 된다. 따라서 중간 화상에서의 x좌표 2 차이에 의한 최종 투영평면으로부터의 거리차는 아래 수식에 의해 계산된다.

$$D(u+2, v, w) - D(u, v, w) = 2 \times \text{View.c} \quad (9)$$

지금까지 x방향으로의 깊이 값의 차이를 계산하였다. 음영 처리를 위해 필요한 법선 벡터 계산을 위해서는 최종 좌표계상의 x방향 경사도 벡터가 필요하다. 이를 위해서는 깊이 차를 구했

던 중간 평면상의 두 점의 좌표 차이에 의한 벡터 (2, 0, 0)의 최종 좌표계상의 벡터로의 변환이 필요하다. (2, 0, 0)을 M_{warp}에 따라서 워핑시키면 (2×View'.a, 2×View'.b, Diff_x)이 얻어지고, 최종적으로 구해진 워핑된 x방향 경사도 벡터(\overrightarrow{G}_x')는 아래와 같다.

$$\overrightarrow{G}_x' = (2 \times \text{View}' .a, 2 \times \text{View}' .b, \text{Diff}_x \quad (10)$$

비슷한 방법으로 워핑된 y방향 경사도 벡터를 구해보면 아래 수식과 같다.

$$\textcircled{1} : (w_1 - w_2) \times \text{dist}_{\text{slice}} \quad (11)$$

$$\begin{aligned} \textcircled{2} : & \text{View.f} \times (y \text{ 인덱스의 차이}) \\ & = (y+1-(y-1)) \times \text{View.f} = 2 \times \text{View.f} \quad (12) \\ \text{Diff}_y & = \textcircled{1} + \textcircled{2} \quad (13) \end{aligned}$$

$$\overrightarrow{G}_y' = (2 \times \text{View}' .d, 2 \times \text{View}' .e, \text{Diff}_y \quad (14)$$

중간 화상의 (x, y)에서의 법선 벡터는 $\overrightarrow{N}_{(x,y)} = \overrightarrow{G}_x' \times \overrightarrow{G}_y'$ (여기서 ×는 벡터외적)이고, 중간 칼라 화상의 r, g, b값을 이용하여 중간 음영 화상의 (x, y)에서의 음영 값을 결정한다.

비록 워핑 후 음영 처리 이진 렌더링에서의 음영 처리보다 계산이 많긴 하지만 시각 방향과 시각 인자(viewing parameter)가 정해지면 ①의 두 번째 요소(dist_{slice})와 ②가 상수가 되기 때문에 \overrightarrow{G}_x' (\overrightarrow{G}_y')의 계산은 한 번의 부동점 곱셈과 한 번의 부동점 덧셈으로 가능하다.

이 방법은 중간 깊이 화상의 구성을 한 번의 지정 연산(assignment operation)으로 간소화하고 중간 화상에서 음영처리를 수행하여 워핑 횟수를 1번으로 줄이므로써 워핑 후 음영처리 이진 렌더링에 비해 보다 빠른 렌더링을 수행한다.

지금까지는 칼라 화상 슬라이스로 구성된 볼륨 데이터를 쉬어-왓 분해에 근거하여 이진 볼륨 렌더링을 수행하는 방법에 대하여 설명하였다. 다음 절에서는 전처리 단계에서 만들어지는 인코딩 데이터의 양을 줄일 수 있는 3-D 경계 인코딩 방법에 대하여 설명하겠다.

3-D 경계 인코딩

앞 절에서 제시한 방법에서 사용하는 데이터는 전처리 단계에서 구성되는 런-길이 인코딩 데이터이다. 이 데이터는 원래 데이터에 비해서는 훨씬 적은 양이기는 하지만, 광선과 만나는 첫 번째 비투명 복셀의 r, g, b값과 깊이 값만을 사용하여 가시화를 수행하는 이와 같은 방법에서는 모든 슬라이스의 볼륨 데이터를 전부 인코딩하여 저장하는 것은 별 의미가 없다. 실제 렌더링에 필요한 데이터는 물체를 구성하고 있는 경계에 해당하는 복셀값 뿐이다. 따라서, 보다 효율적인 렌더링 방법은 물체의 경계에 해당하는 복셀의 정보만을 인코딩 저장하여 가시화를 수행하는 것이다.

경계에 있는 비투명 복셀만을 인코딩하는 방법은 크게 두 가

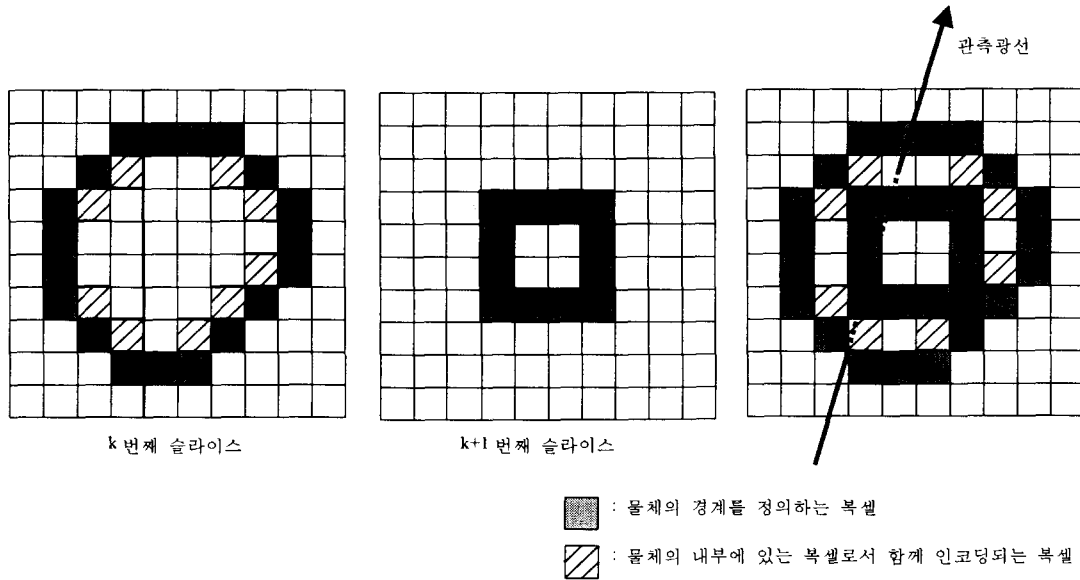


그림 5. 2-D 경계 인코딩의 문제점 :

- (a) k번째 슬라이스를 2-D 경계 인코딩한 결과
- (b) k+1번째 슬라이스를 2-D 경계 인코딩한 결과
- (c) 실제로는 물체의 내부를 지나지만 투명한 복셀의 투영에 의해 최종 화상에서 구멍이 생긴다.

Fig. 5. Problem of 2-D boundary encoding

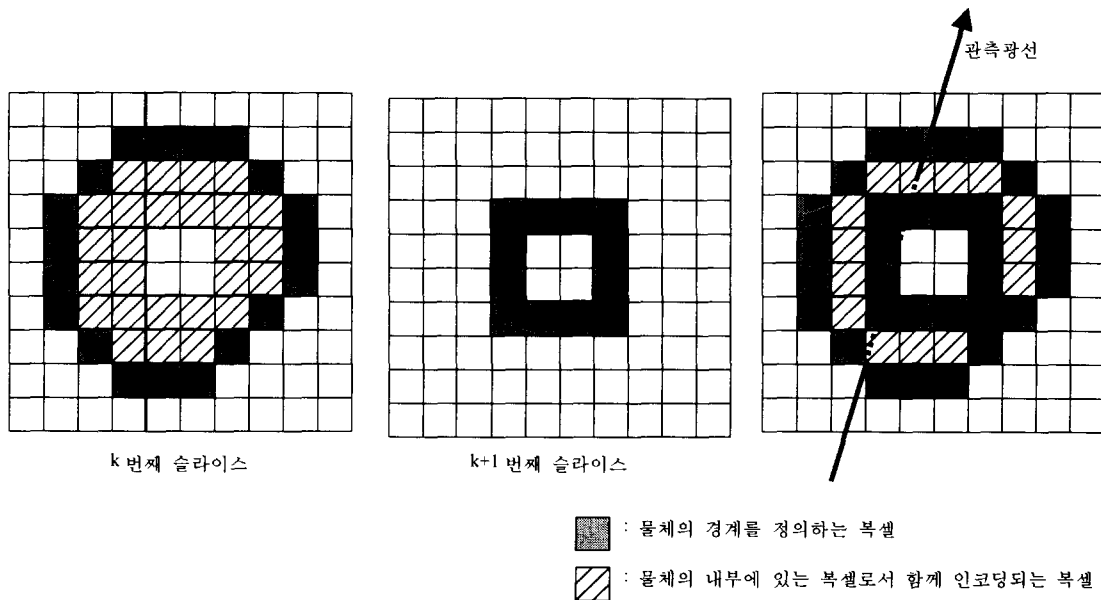


그림 6. 3-D 경계 인코딩 : 그림 5에서와 같은 최종 화상에서의 구멍이 생기지 않는다

Fig. 6. 3-D boundary encoding

지가 있다. 첫 번째 방법은 2-D 경계 인코딩으로서 스캔라인에서의 물체의 경계를 정의하는 비투명 런의 양쪽 끝 복셀뿐만 아니라 물체의 내부에 있는 복셀의 일부도 함께 인코딩하는데, 내부 복셀중에서 2차원 슬라이스 화상에서의 8-이웃(8-neighbor) 복셀들중에서 하나라도 투명한 복셀이 있으면 경계에 있는 것으

로 간주하여 그 내부 복셀을 저장한다. 이 방법의 문제점은, 각 슬라이스에서의 물체모양의 급격한 차이에 의하여 실제 물체 내부를 지나면서도 비투명 복셀과 만나지 못하는 광선의 투영에 의해 렌더링된 화상에서 구멍이나 경계 부분에서의 계단 현상이 생길 수 있다는 점이다.(그림 5)

표 1. PC에서의 렌더링 시간(ms)과 인코딩 데이터의 양(Kbytes)
Table 1.

데이터 집합	모든 비투명 복셀 인코딩				3-D 경계 인코딩			
	위핑 후 음영처리 이진 렌더링		위핑 전 음영처리 이진 렌더링		위핑 후 음영처리 이진 렌더링		위핑 전 음영처리 이진 렌더링	
	평균	메모리	평균	메모리	평균	메모리	평균	메모리
128×128×128 볼륨 데이터	312	1681	282	1681	301	271	268	271
290×158×150 볼륨 데이터	496	2064	443	2064	468	449	428	449
580×316×295 볼륨 데이터	1956	13677	1704	13677	1859	2036	1621	2036

두 번째 방법은 3-D 경계 인코딩으로서 물체의 경계에 있는 복셀과, 물체의 내부에 있는 복셀중에서 3차원상에서의 26-이웃(26-neighbor) 복셀들중에서 하나라도 투명한 복셀이 있으면 그 복셀도 함께 인코딩하여 렌더링을 수행한다. 이 방법은 한 슬라이스내에서 뿐만 아니라 앞, 뒤 슬라이스에서의 물체의 모습의 차이를 조사하여 인코딩할 데이터를 결정하기 때문에 2-D 경계 인코딩에서와 같은 문제점을 해결 할 수 있다(그림 6).

이러한 3-D 경계 인코딩에 의한 정보 저장과 계산 시간면에서의 효율성은 매우 크며, 인코딩하는 물체의 크기가 크면 클수록 더욱 높은 효율성을 기대할 수 있다. 3-D경계 인코딩에 의한 데이터 양의 감소는 PC에 기반한 의학 해부도 응용 프로그램의 개발을 용이하게 하고, 그만큼 프로그램 수행시에 메모리에 올라오는 데이터의 양을 줄임으로써 렌더링 기능의 향상을 가져온다.

실험 및 결과

제 2절에서 제시한 두 가지 방법의 성능(performance) 실험 결과는 표 1에 요약되어 있다. 두 가지 방법은 모두 평행 투영에 의한 것이며, 렌더링 시간은 펜티엄 II 200Mhz, 주메모리 128Mbytes의 일반 PC에서 측정하였다. 렌더링 시간은 새로운 시각 인자나 음영 처리 인자가 주어졌을 때 필요한 모든 단계(합성, 위핑, 음영처리)에서 소요되는 시간을 밀리 세컨드(ms)로 보여주고 있으며 전처리 시간은 포함되어 있지 않다. '메모리'는 볼륨 데이터를 런-길이 인코딩하면서 생성되는 모든 데이터의 양을 킬로 바이트(Kilobyte)단위로 나타내고 있다.

렌더링 시간은 화상의 크기와 관련이 있는데 이는 합성 단계에서 처음으로 만나는 비투명 복셀의 값으로 중간 화상의 화소값을 결정하기 때문에, 합성 시간에 비해 상대적으로 위핑 시간이 렌더링 시간에 중요하게 작용하기 때문이다. 쉬어-왓 렌더링[4]에서는 음영 단계 렌더링(gray scale rendering)을 수행하였지만, 칼라 물체를 렌더링하기 위해서는 위핑단계에서 red, green, blue 각 채널의 쌍일차 보간이 필요하기 때문에 보다 많

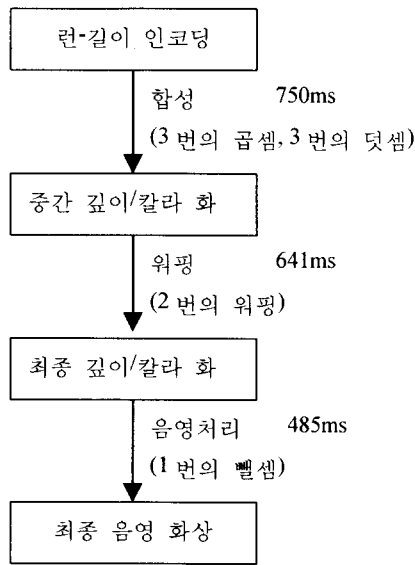
은 시간이 소요된다.

580×316×295의 RGB볼륨 데이터를 저장하는데, 3-D 경계 인코딩을 적용하면 불과 2036 Kbyte의 데이터로 렌더링이 가능하다. 3-D경계 인코딩에 의한 정보 저장면에서의 효율성은 인코딩하는 물체의 크기가 크면 클수록 뛰어나다. 물체의 크기가 2배로 커질 때, 모든 비투명 복셀 데이터를 인코딩하는 경우에는 데이터의 양이 8(2³)배 커지지만 3D경계 인코딩을 적용하면 표면에 있는 복셀만 저장되어 4배만큼 증가하기 때문이다.

그림 7은 제 2절에서 제시한 두 가지 방법을 사용하여 580×316×295볼륨 데이터로 샘플링된 근육을 렌더링하는데 걸리는 시간과 각 화소당 필요한 연산을 단계별로 보여 주고 있다. 왼쪽은 위핑 후 음영처리 이진 렌더링을 적용한 경우이고, 오른쪽은 위핑 전 음영처리 이진 렌더링을 적용한 경우이며, 모두 3-D 경계 인코딩한 데이터를 사용하여 렌더링을 수행하였다. 위핑 후 음영처리 이진 렌더링에서는 실제 깊이 값을 중간 깊이 화상의 화소값으로 부여하기 때문에, 주 시각 좌표축상의 좌표값을 중간 깊이 화상의 화소값으로 정하는 위핑 전 음영처리 이진 렌더링에서보다 합성단계에서 보다 많은 시간이 소요된다. 위핑 단계에서는 위핑 후 음영처리 이진 렌더링을 사용하면 중간 깊이 화상과 중간 칼라 화상의 위핑이 필요하기에 두 번의 위핑을 해야 하지만, 위핑 전 음영처리 이진 렌더링에서는 한 번의 위핑으로 최종 화상을 얻을 수 있다. 음영 처리단계에서는 위핑 후 음영처리 이진 렌더링 사용시 최종 깊이 화상과 최종 칼라 화상이 구성되어 있기 때문에 x, y 방향의 경사도를 구하는 것이 각각 한 번의 부동점 빼기연산만으로 계산이 가능하다. 반면 위핑 전 음영처리 이진 렌더링에서는 새로운 법선 벡터 계산 방법을 적용하여 중간 화상에서 음영 처리를 수행하며, 이것은 한 번의 곱셈과 한 번의 덧셈으로 경사도 벡터를 계산한다.

그림 8은 3D경계 인코딩한 580×316×295볼륨 데이터를 위핑 전 음영 처리 이진 렌더링을 수행할 때, 가시화하는 물체 개수의 변화에 따른 렌더링 각 단계(합성, 음영처리, 위핑)에서의 소요시간의 변화를 보여주고 있다. 음영 처리와 위핑은 이차원 화상에서 수행되기 때문에 렌더링되는 입력 데이터의 크기와는

<위평 후 음영처리 이진 렌더링>



<위평 전 음영처리 이진 렌더링>

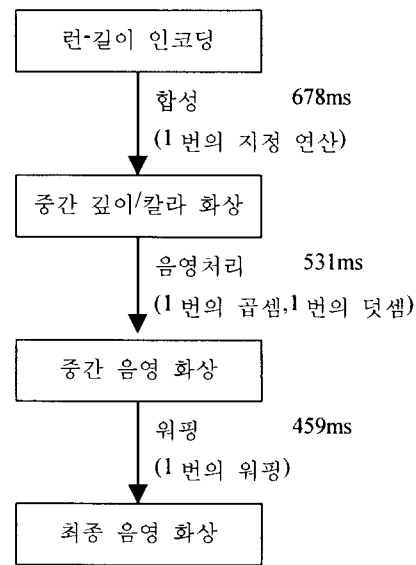


그림 7. 각 렌더링 단계의 수행 능력 결과와 필요한 연산

Fig. 7. Performance results and necessary operations of each step of binary rendering

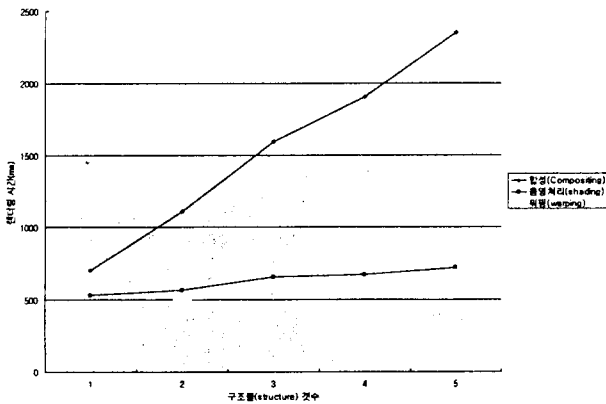


그림 8. 물체 개수 변화에 따른 렌더링 각 단계에서의 소요시간

Fig. 8. Performance results for each stage of rendering with various number of objects

별 상관없이 2차원 평면에 투영되었을 때의 물체의 크기에 의존하기에, 가시화하는 물체의 개수가 많아진다 하더라도 크게 증가하지 않는다. 반면, 가시화하는 물체의 개수가 많아지면 렌더링을 위해 컴퓨터에 로드되는 데이터의 양이 커지며 중간 깊이/칼라 화상의 하나의 스캔라인을 구성하기 위해 동시에 살펴보는 인코딩 데이터가 많아지므로 합성 단계의 시간의 증가량은 크다. 하지만 실험 결과 580×316×295 데이터를 가시화할 때 합성시간은 물체의 개수에 따라 두 배, 세 배 증가하지 않고 일정한 값(411ms)의 증가를 보였다.

그림 9는 신체 하지부분에 있는 23개의 근육들과 뼈를 경계 추출하여 얻은 RGB칼라 볼륨 데이터를 반투명한 피부와 함께

가시화한 그림이다. 데이터는 3D경계 인코딩하였고 위평전 음영처리 이진 렌더링을 적용하였다. 그림 10는 그림 9에서 피부는 삭제하고, 23개의 근육들을 인공적인 색상으로 렌더링하여 얻은 영상이다.

결론 및 향후 과제

최근 들어 의학 해부도 응용 프로그램의 필요성이 증가하면서 visible human data를 이용한 3-D 렌더링 기법에 대한 연구가 활발해지고 있다. 하지만 기존의 방법들은 현실감있는 영상을 얻기 위해 먼 렌더링에서 3-D 텍스처 맵핑을 수행하거나, 고가의 장비를 이용한 볼륨 렌더링을 수행해 왔다. 본 논문에서는 방사선 영상 데이터가 아닌 visible human data에서 얻은 경계 추출된 칼라 슬라이스로 구성된 볼륨 데이터를, 일반 PC에서 대화식 수준의 속도로 3-D 가시화하는 이진 볼륨 렌더링 방법을 제시하였다. 이 방법은 이진 볼륨 렌더링을 위해 필요한 깊이 화상의 구성을 간소화하고 새로운 법선 벡터 계산 방법을 적용하여 렌더링 시간의 향상을 꾀하였다. 또한, 이미지 질의 손실 없이 인코딩 데이터의 양을 줄이는 3-D경계 인코딩 방법을 이용하여 렌더링 시간의 감소를 꾀하였다. 본 논문에서 제시한 렌더링 방법의 대화식 수준의 속도와 인코딩 방법의 정보 저장면에서의 효율성은 PC에서 운용될 수 있는 의학 해부도 응용 프로그램의 개발을 보다 가속화할 것이다.

이진 볼륨 렌더링에서 속도 저하의 주된 요인은 관측점이나 시각 방향이 변경되면 이진 깊이 화상을 다시 만들어야 하고 매번 법선 벡터가 새로이 계산되어야 한다는 점이다. 밀도 볼륨

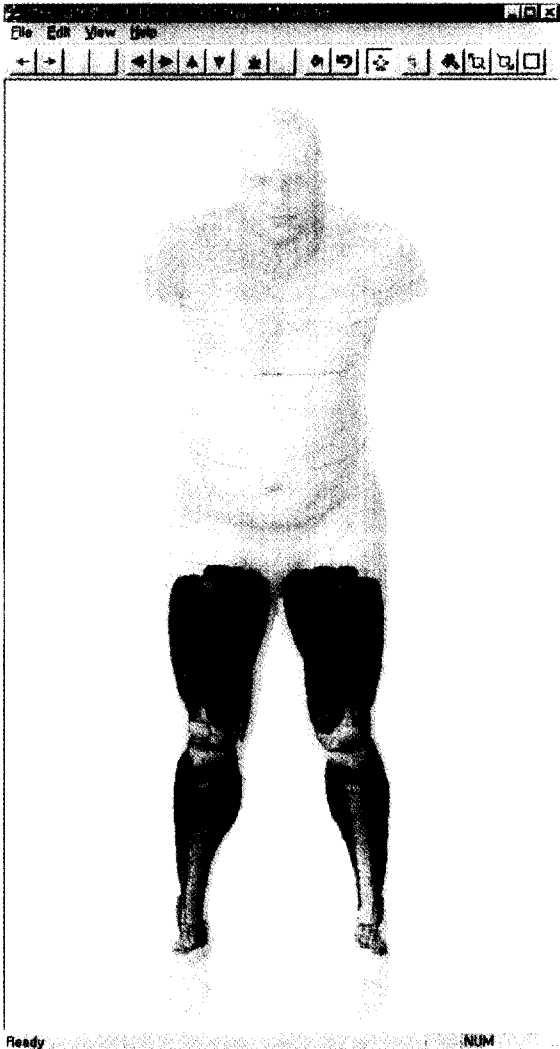


그림 9. 전체 피부와 하지부분 23개의 근육과 뼈의 가시화 영상 (피부는 반투명하게 렌더링)
 Fig. 9. Rendering of semi-transparent skin, 23 muscles and bones at the low extremity of human body

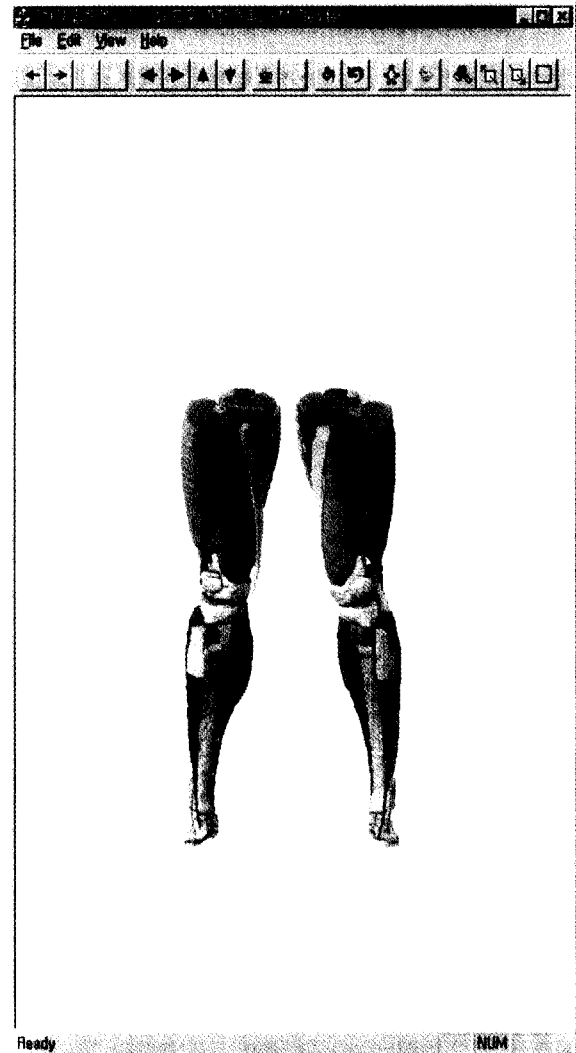


그림 10. 하지 부분 23개의 근육과 뼈의 가시화 영상(인공적인 색으로 렌더링)
 Fig. 10. Artificial highlighting of 23 muscles and bones at the lower extremity of human body

데이터(density volume data)를 사용하는 볼륨 렌더링에서는 샘플 포인트의 법선 벡터값이 런-길이 인코딩데이터에 포함되어 있어서 룩-업(look-up)테이블[23]을 이용한 음영 처리를 수행하기 때문에 보다 빠른 음영값 계산을 할 수 있다. 지금까지 밀도 볼륨 데이터에 있어서 법선 벡터를 계산하는 것과 관련된 연구는 많이 행해져 왔다[24]. 반면 2차원 칼라 데이터와 같이 밀도값이 주어지지 않는 경우에 표면의 법선 벡터를 추정하는 방법에 관한 연구는 아직 미흡하다. 이와 관련된 연구는 이진 볼륨 렌더링에서의 음영처리 속도 향상의 가능성을 제시해 줄 것이다.

참 고 문 헌

1. Ylä-Jääski, J., Klein, F. And K ber, O., "Fast direct display of volume data for medical diagnosis," CVGIP: Graphical Models and Image Processing, Volume 53, Number 1, pp. 7-18, 1992.
2. Cameron, G. G., and Undrill, P. E., "Rendering volumetric medical image data on a SIMD-architecture computer," Proc. Third Eurographics Workshop on Rendering, pp. 135-145, 1992.
3. Schr der, P. and Stoll, G., "Data parallel volume render-

- ing as line drawing*," 1992 Workshop on Volume Visualization, pp. 25-31, 1992.
4. Lacroute, Philippe and Levoy, Marc, "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation," Computer Graphics, 28, 5, pp. 451-458, 1994.
 5. Yagel, R. and Kaufman, A., "Template-Based Volume Viewing," Computer Graphics Forum, Volume 11, Number 3, pp. 153-167, September 1992.
 6. Yagel, R. and Ciula, K., "High Quality Template-Based Volume Rendering," Ohio State University Technical Report, OSU-CISRC-10/93-TR35, 1993.
 7. Wolberg, G., Digital Image Warping, IEEE Computer Society Press, Los Alamitos, CA, 1990.
 8. Danskin, John and Pat Hanrahan, "Fast algorithms for volume ray casting," 1992 Workshop on Volume Visualization, pp. 91-98, 1992
 9. Laur, David and Pat Hanrahan, "Hierarchical splatting: A progressive refinement algorithm for volume rendering," Computer Graphics, 25, 4, pp 285-288, 1991
 10. Levoy, Marc, "Efficient ray tracing of volume data," ACM Transactions on Graphics, 9, 3, pp 245-261, 1990
 11. Meagher, Donald J., "Efficient synthetic image generation of arbitrary 3-D objects," Proceeding of the IEEE Conference on Pattern Recognition and Image Processing, pp 473-478, 1982
 12. Subramanian, K.R. and Donald S. Fussel, "Applying space subdivision techniques to volume rendering," Proceedings of Visualization '90, pp 150-159, 1990
 13. Choel-Hi Lee, Yun-Mo Koo, and Yeong Gil Shin, "Template-Based Rendering of Run-Length Encoded Volumes", Proceedings of Pacific Graphics, pp 138-147, 1997
 14. K. H. Hohne, A. Pommert, M. Riemer, Th. Schiemann, R. Schubert, and U. Tiede, "Anatomical Atlases Based on Volume Visualization", Proceedings of Visualization, pp 115-122, 1992
 15. T. L. Hardy, "Interpretation of tomographic images using automatic atlas lookup", Visualization in Biomedical Computing, pp 447-456, 1994
 16. A. Pommert, R. Schubert, M. Riemer, T. Schiemann, U. Tiede, and K.H. H hne, "Symbolic modeling of human anatomy for visualization and simulation", Visualization in Biomedical Computing, pp 412-423, 1994
 17. Ulf Tiede, Thomas Schiemann, and Karl Heinz H hne, "Visualizing the Visible Human", IEEE Computer Graphics and Applications, 16(1), pp 7-9, 1996
 18. Thomas Schiemann, Jochen Nuthmann, Ulf Tiede, and Karl Heinz H hne, "Segmentation of the Visible Human for High Quality Volume based Visualization" Visualization in Biomedical Computing, pp 13-22, 1996
 19. John Kerr, Peter Ratiu, and Mike Sellberg, "Volume Rendering of Visible Human Data for an Anatomical Virtual Environment", IOS Press and Ohmsha, 1996
 20. Levoy, M., "Display of Surfaces from volume data", IEEE Computer Graphics and Applications, pp 29-37, 1988
 21. Mokrzycki, W., "Algorithms of discretization of algebraic spatial curves on homogenous cubical grids.", Computers and Graphics, pp 477-487, 1988
 22. Thomas Porter and Tom Duff, "Compositing Digital Images", Computer Graphics July 1984, pp 49-55, 1984
 23. Glassner, Andrew S., "Normal Coding", Graphics Gems, pp 257-264, Academic Press, New York, 1990
 24. L. S. Chen, G. T. Herman, R. A. Reynolds, and J. K. Udupa, "Surface Shading in the Cuberille Environment", IEEE Computer Graphics and Applications, 5(Dec.), pp. 33-43, 1985