

# 분할된 볼륨의 가시화에서 객체당 투명도

정동균<sup>†</sup>, 이철희<sup>\*\*</sup>, 신영길<sup>\*\*\*</sup>

## 요 약

볼륨 렌더링에서는 기본적으로 전이함수에 의해 객체를 구분하여 표시한다. 그러나, 경우에 따라서는 전이 함수만으로는 객체를 구분할 수 없는 경우가 있다. 이 때는 객체를 다른 방법으로 미리 분할한 후 분할 정보를 바탕으로 가시화를 수행한다. 본 논문은 분할된 볼륨의 가시화에서 객체당 투명도를 부여하는 방법을 제시한다. 반투명한 렌더링은 관찰의 대상이 되는 객체에 대한 맥락정보를 효과적으로 전달되기 위해 사용된다. 객체당 투명도는 전이함수의 조절에 의해 객체를 반투명하게 조절하기 어려운 경우에 매우 효과적인 가시화 도구로 사용될 수 있다. 객체당 투명도의 의미에 대한 몇가지 해석과, 그에 따른 알고리즘의 변화를 제시하였다. 또한, 제안된 방법을 일반 그래픽스 하드웨어를 사용하여 구현함으로써, 대화식 사용이 가능한 효율적 구현이 가능함을 보였다.

## Per-Object Transparency in Visualization of Segmented Volumes

Dongkyun Jeong<sup>†</sup>, Cheol-Hi Lee<sup>\*\*</sup>, Yeong Gil Shin<sup>\*\*\*</sup>

## ABSTRACT

Basically, objects are discriminated by transfer functions in volume rendering. However, in some cases objects cannot be discriminated only with transfer functions. In these cases, objects are pre-segmented with other methods, and visualized based on the segmentation information. In this paper we present a way of assigning per-object transparency in visualization of segmented volumes. Semi-transparent rendering is used to effectively give context information about the observed object. Per-object transparency can be used as a very effective visualization tool especially when it is difficult to adjust transfer functions to make the object semi-transparent. We present several interpretations of the meaning of per-object transparency, and corresponding variations of the algorithm. We show that efficient implementations for interactive use are possible, by presenting an implementation using general graphics hardware.

**Key words:** Volume Rendering(볼륨 렌더링), Per-Object Transparency(객체당 투명도), Segmented Volume(분할 볼륨), Context Information(맥락 정보), Hardware Volume Rendering(하드웨어 볼륨 렌더링)

### 1. 볼륨 렌더링 (Volume Rendering) 및 분할된 볼륨의 렌더링

볼륨 렌더링은 과학분야 가시화나 의료분야의 가시화에서 사용되어 왔다. 특히 의료분야에서는 256×

256 또는 512×512의 2차원 영상의 스택을 3차원적으로 가시화하여 진단에 활용하는 기법이 연구되어 왔으며, 최근 의료영상 획득 장비 해상도의 획기적 개선과 의료영상저장 및 전송 분야 전산화의 진행에 힘입어, 진료에 일상적으로 사용되기 시작하는 단계

※ 교신저자(Corresponding Author) : 이철희, 주소 : 서울시 서초구 서초동 1603-54, 전화 : 02)3470-5275, FAX : 02) 2055-1405, E-mail : chlee@sit.ac.kr

접수일 : 2004년 10월 1일, 완료일 : 2005년 4월 18일

<sup>†</sup> 서울대학교 컴퓨터공학과 박사수료

(E-mail : babyimp@cglab.snu.ac.kr)

<sup>\*\*</sup> 서울벤처정보대학원대학교 컴퓨터응용기술학과 전임 강사

<sup>\*\*\*</sup> 서울대학교 컴퓨터공학과 교수

(E-mail : yshin@cglab.snu.ac.kr)

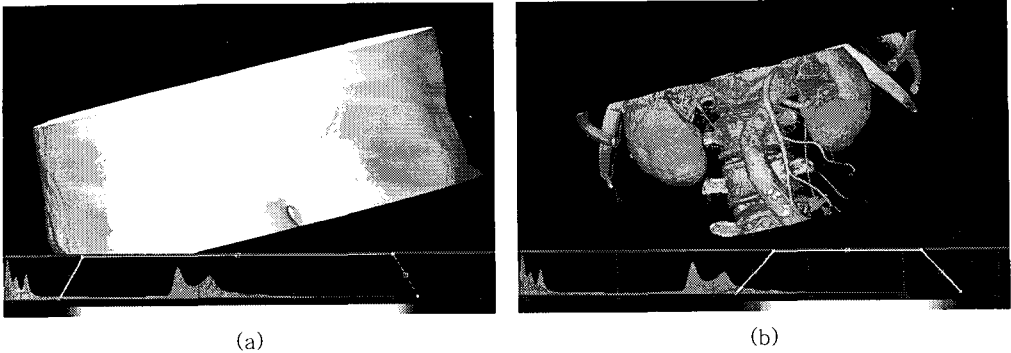


그림 1. OTF 변경에 의한 가시객체의 변경 : CT영상의 볼륨 렌더링의 예이며, 각각의 하단에 나타난 그래프의 x축은 CT 값, y축은 불투명도 (opacity)를 나타낸다. 그래프 하단의 색상은 CTF를 나타낸다: (a) 피부, (b) 골격.

에 와 있다.

볼륨 렌더링은 3차원 공간상에 분포된 데이터를 가시화하는 기법으로서, 공간상에 존재하는 가상의 물체에 대해 투명도와 색상을 지정하고 그것을 렌더링한다[1,2]. 물체에 투명도와 색상을 지정하는 방법으로서 불투명도 전이함수(OTF, Opacity Transfer Function) 및 색상 전이함수(CTF, Color Transfer Function) 라고 하는 함수들을 사용하며, 각 함수는 공간상의 각 위치에 분포한 데이터 값 자체와 그것의 그래디언트 등으로부터 투명도 및 색상을 할당하는 역할을 한다. 즉, 시각적으로 확인하고자 하는 데이터에 대하여는 1에 가까운(불투명한) 값을 할당하고, 다른 데이터에 대하여는 0의(투명한) 값을 할당하는 OTF를 사용하면, 화면상에서 다른 부위는 사라지고 보고자 하는 부위만이 나타나게 된다.

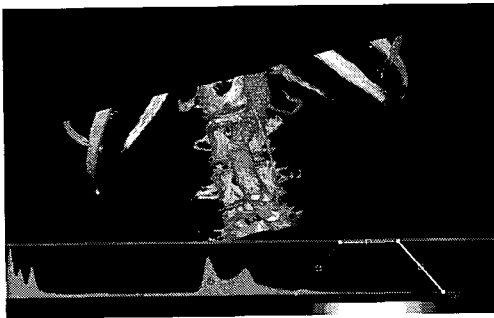


그림 2. 그림 1 (a)에 다른 OTF를 적용 : OTF에 의해 조정된 혈관과 뼈를 구분할 수 없으므로, 혈관을 없애기 위해 OTF를 조절하면 뼈도 일부 사라진다.

그림 1에 나타난 바와 같이 OTF의 변경은 볼륨 렌더링에서 가시 객체를 변경하기 위한 주요한 수단

으로 사용된다. 그러나, 볼륨 렌더링의 대상이 되는 객체가 항상 OTF의 변경에 의해 다른 객체와 구분되는 것은 아니다. 그림 2에서, 조영된 혈관과 뼈의 CT값은 대부분 중첩되므로 이들은 OTF에 의해 서로 구분되지 않는다.

이런 경우, 특히 OTF에 의해 구분되지 않는 객체가 서로 상대방을 가린다면, 별도의 분할 방법에 의해 객체를 서로 구분하고, 구분된 객체에 별도의 가시성을 지정할 수 있다[3-5]. 의료영상의 3차원 가시화에서는 이와 같은 분할방법이 널리 이용되어 왔다. 분할에 관련된 정보는 보통 각 객체당 1비트씩의 분할정보를 저장하는 마스크 볼륨 또는 해당 위치에서의 객체 ID를 저장하고 있는 객체 ID 볼륨에 의해서 저장된다. 이 경우 볼륨 렌더링 알고리즘은 각 복셀을 렌더링하기 전에 복셀이 속한 객체의 가시성을 검사하여 보이지 않는 경우에는 렌더링을 생략하게 된다.

객체의 가시성을 대화식으로 켜고 끄으로써, 볼륨 렌더링 알고리즘은 가시화하고자 하는 객체에 대한 맥락(context) 정보를 제공할 수 있다. 그림 3에서, 혈관의 이상(팽창 또는 막힘)을 진단할 경우에는 혈관을 가리는 뼈의 정보는 불필요하므로 혈관만을 보게 되나, 이상이 생긴 부위의 정확한 위치를 알기 위해서는 뼈를 함께 렌더링하는 것이 도움이 된다. 두 객체를 동시에 렌더링함으로써 객체간 상호간의 위치 관계가 표시되고, 뼈를 기준으로 하여 이상부위의 위치를 정확히 파악할 수 있다.

맥락정보는 맥락이 되는 객체를 반투명하게 렌더링함으로써 효과적으로 제공될 수 있다[3,6]. 객체를 반투명하게 렌더링하는 것은 볼륨 렌더링에서 OTF의 조절을 통해 가능하다. 그러나, OTF를 조절하여 반투명한 객체를 만드는 것은 항상 성공적이지 않음

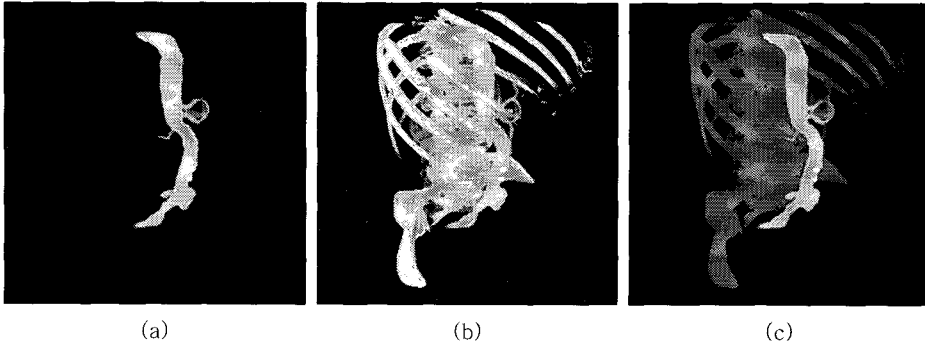


그림 3. 맥락 정보의 표시: (a) 혈관, (b) 혈관+뼈, (c) 반투명한 맥락정보의 표시.

며, 매우 어려울 수 있다. 일반적으로 볼륨 렌더링에서 좋은 OTF와 CTF를 지정하는 것은 매우 어려우며, 그 방법은 별도의 연구 주제가 되기도 한다[7]. 그리고, 효율적인 가시화를 위해서는 그림 4에 보는 바와 같이 한 객체에 대해서도 다중의 OTF 및 CTF를 사용하기도 한다. 이는 OTF 및 CTF의 조정을 더욱 어렵게 만든다. 통상 상용 제품에서는 기정의된 OTF 및 CTF를 프리셋(preset) 또는 프로토콜(protocol) 형태로 제공하고, 최종 사용자는 기정의 함수 값 전체의 이동(translation)이나 크기변경(scale)등의 단순한 조절만을 시행하도록 한다.

본 논문은, 분할된 볼륨의 가시화에서 객체당 OTF와 별도로 객체당의 투명도를 부여하는 방법과 이를 구현하는 볼륨 렌더링 알고리즘을 제시한다. 이를 통해 사용자는 기정의되거나 복잡한 과정을 통해 얻어진 전이함수(transfer function)을 쉽게 조작하

여 효과적으로 객체당의 가시성을 조작하고 맥락정보를 제공하는 방법을 얻게 된다. 또한, 구현된 볼륨 렌더링 방법을 사용하여 효과적으로 맥락 정보를 제공하는 사례를 제시하고, 본 방법이 기존의 방법의 어떤 문제들을 해결하는지 설명한다.

## 2. 관련연구

객체의 가시성을 지정하고 효과적인 맥락정보를 제공하기 위해 OTF를 지정하는 방법에 대해 많은 연구가 행해졌다[7-11]. 특히[6,12]에서는 일반적으로 볼륨 렌더링에서 사용되는 단순한 Phong 셰이딩 방법을 개선하여 반투명 객체 렌더링을 제공하기 위해 새로운 볼륨렌더링 식을 제시하였다.

이단계 볼륨렌더링[3,4]은 분할된 볼륨에서 각 볼륨당 MIP, 볼륨 렌더링, 비사실적 렌더링 등 다양한 렌더링 기법을 제공함으로써 맥락 정보를 제공하고 자 하였다. 결과적으로 볼륨렌더링 기법만을 사용하는 단일패스의 이단계 볼륨 렌더링은 본 논문과 매우 유사한 렌더링 알고리즘을 제시한다. 이는 양 연구가 대상으로 하는 데이터가 유사함을 고려하면 당연한 일이다. 이단계 렌더링의 목표는 다양한 렌더링 알고리즘의 수용, 본 논문의 방법은 객체당 투명도 부여라는 차이가 있다.

[2]는 분할에 대한 다른 접근방법을 제시한다. 여기에서 분류(classification)는 OTF 및 CTF 지정의 의미 외에 분할의 의미를 포함한다. 즉, 퍼지(fuzzy) 분할 알고리즘에 의해 분할된 객체는 고정된 색상 및 투명도를 가지며, 객체에 속한 각 복셀은 (확률) $\times$ (객체당 색상 및 투명도)에 의해 계산되는 색상 및 투명도를 가지게 된다. 또한, 별도의 합성마스크 볼

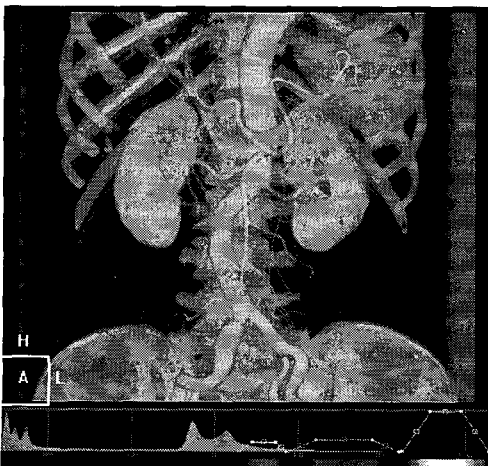


그림 4. 다중의 OTF 및 CTF 사용의 예

룸(matte volume)은 백분율로 분할 정보를 저장하고 있어, 다른 단계의 객체 지정 방법이 된다. 다만, 여기서는 분할에 대해 직접적으로 언급하고 있지 않으며, 상기 두 방법이 분할 방법에 해당한다 하겠다. 이 경우에 분할은 백분율로 지정되는 셈이며, 따라서 단순히 복셀당 1비트씩으로 객체에 포함되는지 여부를 나타내는 일반적인 방법과는 차이가 있다.

### 3. 객체별 투명도를 부여하는 볼륨 렌더링

분할된 객체들은 보통 복셀당 1비트씩이 할당된 객체당 하나씩의 마스크 볼륨이나, 각 복셀이 속한 객체의 ID를 저장하는 볼륨을 통해 지정된다. 본 논문에서는 객체 ID 볼륨을 기준으로 설명해 나가겠지만, 마스크 볼륨을 사용하기 위한 알고리즘의 변환은 매우 단순하다. 드물게는 객체 마스크 볼륨이 복셀당 1비트 이상을 사용할 수 있다. 이 경우 각 복셀의 마스크 값의 의미는 해당 객체의 점유율이 된다. 이 경우의 알고리즘 변환 또한 단순한 확장에 의해 가능하다.

본 절에서는 객체별 투명도 부여의 의미를 설명하고, 이를 구현하기 위한 알고리즘의 개요를 제시한다.

#### 3.1 객체별 투명도의 부여

어떤 객체에 대해 불투명도  $a_0$ 를 부여하면, 해당 객체를 렌더링한 결과로 얻어진 불투명도  $a_s$ 에  $a_0$ 를 곱한 값  $a_s a_0$ 를 해당 광선에 대한 객체(세그먼트)의 최종 불투명도로 한다. 만일 렌더링의 결과 완전 불투명, 즉 1의 불투명도를 얻었다면, 세그먼트의 불투명도는 해당 객체의 불투명도  $a_0$ 와 같다. 또한  $a_s < 1$ 인 경우  $a_s a_0 < a_0$ 가 되어 객체별 투명도보다 더욱 투명해진다. 객체별 투명도가 원래 객체 렌더링 결과가 완전 불투명일 경우의 투명도라고 생각하며, 렌더링 결과가 해당 세그먼트가 반투명할 경우 객체별 투명도보다 더 투명해지는 것은 합리적이다. 그림 5의 광선  $\alpha$ 는 객체 1, 2, 3과 만난다. 각각에 부여된 불투명도가 각각 0.5, 0.3, 0.7이고, 각 객체의 세그먼트들을 볼륨 렌더링한 결과 각각 1, 0.7, 1의 불투명도가 계산되었다면, 각 객체 세그먼트들의 최종 불투명도는 각각  $0.5=1 \times 0.5$ ,  $0.21=0.7 \times 0.3$ ,  $0.7=1 \times 0.7$ 이 된다.

객체별 투명도를 곱하는 것은 각 세그먼트의 렌더링이 종료된 후에만 가능하며, 중간에 투명도를 고려하는 일반적인 방법은 찾기 어렵다. 따라서, 자연스

럽게 세그먼트 내부의 렌더링과 세그먼트간의 렌더링을 구분하는 이단계 볼륨 렌더링의 뼈대를 따르게 된다. 즉, 객체별 투명도가 부여되었을 때 한 광선이 여러 객체의 세그먼트들을 지나게 되면, 각 세그먼트당 계산된 색상과 투명도는 합성(composition)의 방법에 의해 최종 영상의 한 화소에 기여하게 된다. 그림 5의 광선  $\alpha$ 의 경우 최종 화소값은,  $C_B$ 가 배경 색상이고,  $c_i$ 가 세그먼트  $i$ 의 색상일 경우

$0.5 \times c_1 + 0.5 \times (0.21 \times c_2 + 0.79 \times (0.7 \times c_3 + C_B))$ 로 계산된다.

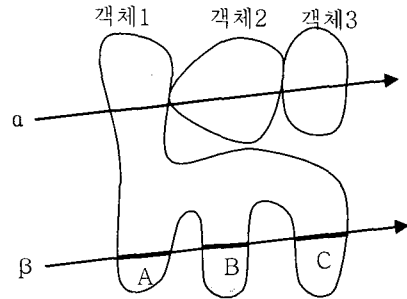


그림 5. 객체별 투명도의 부여

객체당 투명도를 부여하는 대신, 각 객체는 모두 불투명한 것으로 하고, OTF 변환에 의해 투명도를 조절하는 경우를 생각해 보자. 어떤 경우, 예를 들어 대략 비슷한 값들이 비슷한 두께로 나타나는 객체의 경우, OTF 변환에 의해서도 투명도를 조절할 수 있을 것이다. 이 경우는  $a_0$ 는 1이고  $a_s$ 가 1보다 작은 경우이며, 객체가 존재하는 전 화소에 걸쳐  $a_s$ 가 비교적 균일해야 한다. 이러한 방식으로 투명도를 조절할 수 있는 경우에도 객체당 투명도를 부여하는 것이 훨씬 직관적이며, 사용자의 조작성이 편리하다.

또한, 이 방법은 데이터에 의존적이며, 잘 동작하지 않을 가능성도 많이 있다. 예를 들어 객체의 일부는 두껍고 일부가 상당히 얇다면, 정상적인 OTF에서는 전체가 불투명하나, OTF를 투명하게 조절해가면 두꺼운 쪽은 계속 불투명한테 얇은 쪽은 사라져 버릴 수 있다.

이 문제를 극복하기 위한 한가지 방법은 OTF의 정의구역에 그래디언트 크기를 포함시키는 것이다 [1,7]. 이 방법은 그래디언트가 작은 객체의 내부는 렌더링하지 않고, 객체의 표면만 렌더링하는 방법이다. 이 방법은 기존의 볼륨 렌더링 방법을 단순하게

수정하여 그대로 사용할 수 있는 장점이 있으며, 데이터에 따라서는 결과가 더 좋을 수도 있다. 예를 들어, 두께의 차이가 상당히 있으나 심하지 않은 경우, 객체의 뒤쪽으로 튀어나온 부분이 반투명한 렌더링에 의해 사실적으로 보인다거나, 스스로 가리는 객체의 가려진 뒤쪽면이 나타나도록 할 수 있다.

그러나, 이러한 OTF를 단순한 프리셋으로 지정하기 힘들고 데이터 의존적일 수 있으므로[7] 사용이 비교적 복잡하고, 만약 객체의 그래디언트 크기가 작다면 객체의 표면이 잘 정의되지 않아 객체 자체가 잘 보이지 않게 된다. 객체의 그래디언트는 경계선이 불분명하고 완만히 변할 때 작아지게 되며, 또한 분할된 두 객체간의 값차이가 없거나 적을때도 작아지게 되므로, 그래디언트 사용상의 문제는 흔하게 나타나는 일이다[12]. 따라서, 다차원의 OTF를 사용하는 것보다 객체당 투명도를 부여하는 것이 사용자에게 직관적으로 훨씬 편리하고 그래디언트 크기에 의존하지 않는 방법이다.

### 3.2 투명도의 해석

객체당 투명도를 해석하는 문제는 한 광선상에 동일 객체의 세그먼트가 여러 개 존재할 때 생긴다. 그림 5의 광선  $\beta$ 는 객체1과 A, B, C 세개의 세그먼트에서 만난다. 이 경우 가능한 해석은 다음과 같다.

1. 객체당 첫번째 세그먼트만을 렌더링한다.
2. 객체당 처음으로 불투명해지는 샘플에서 객체의 렌더링을 종료한다.
3. 객체의 모든 세그먼트를 렌더링한다.
4. 객체당 투명도를 무시한다. (OTF를 조절하여 투명도를 부여한 것이거나, 가시화하고자 하는 대상 객체이다.)

이러한 해석상의 문제가 생기는 이유는 객체당의 투명도가 물리현상에 기반하지 않은 가상적인 것이기 때문이다. 반면, 볼륨 렌더링 알고리즘은 가상적인 것이기는 하지만, 전이함수에 의해 지정된 물리적 성질을 단순화된 모델로 일관되게 볼륨 전체에 적용한 것이다. 객체당 투명도는 이 모델에 기반하고 있지 않으며, 최종적인 세그먼트 당의 불투명도를 인위적으로 부여한 것이기 때문에, 동일 광선(화소)에 대해 객체당 두개 이상의 세그먼트가 존재할 경우 해석상의 문제가 생긴다.

객체당 첫번째 세그먼트만을 렌더링하는 것은 객체가 원래 불투명한 것이며, 분할은 객체가 보여야 할 모든 화소에 대해 객체가 불투명한 것을 보장하도록 이루어진 것으로 가정한다. 이 경우 객체당 투명도의 의미는, 원래 불투명한 객체를 투명하게 만들어 가려졌던 물체를 원래 객체의 맥락 속에서 가시화하는 것이다.

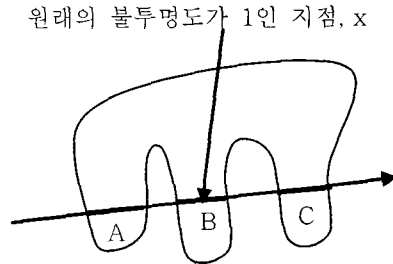


그림 6. 첫번째 불투명한 샘플에서의 종료: 그림에서 세그먼트 A의 불투명도는 <1, 세그먼트 B의 불투명도는 1이라면, 세그먼트 C는 보이지 않는다.

객체당 첫번째로 불투명해지는 샘플에서 렌더링을 종료하는 것은 원래 객체에 일부 반투명한 부분이 있다고 가정한 것이다. 그림 6은 반투명한 부분(세그먼트 A)이 동일 객체의 다른 부분보다 시점으로부터 앞에 있고 중간(세그먼트 A와 세그먼트 B 사이)에 다른 객체의 반투명한 부분이 존재하는 경우를 나타낸다. 이 경우 종료되는 샘플  $x$ 는 불투명도가 1인 해당 객체만을 단독으로 렌더링할 때 종료되는 샘플과 같게 된다. 이렇게 함으로써 세그먼트 A와 세그먼트 B의 합성 결과는 객체당 불투명도를 초과하지 않게 된다. 만일, 세그먼트 C까지 렌더링한다면, 부여된 객체당 투명도보다 불투명해질 수 있다.

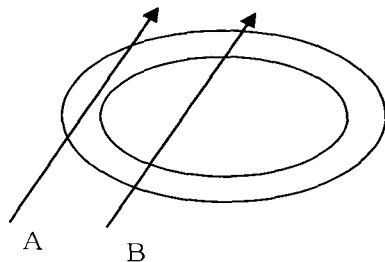


그림 7. 모든 세그먼트 렌더링시의 객체당 불투명도

세번째 방법은 모든 세그먼트를 렌더링하는 것이다. 이것은 예를 들어 뇌와 두개골 세그먼트를 가진

MR 영상의 경우, 뇌 전면과 후면의 두개골 맥락정보를 모두 제공하기 위해 사용될 수 있다. 그러나 이 방법은 그림 7에 보인 바와 같이 한 세그먼트가 길게 연장된 화소 A에서, 짧은 세그먼트 두개가 있는 화소 B보다 더 투명해지게 된다. 이 경우에는 양쪽 화소에서 객체가 동일한 투명도를 가지는 것이 더 자연스럽다. 하지만, 맥락정보의 제공이라는 측면에서는 이런 문제는 사소한 것일 수도 있다.

사실상 객체당 투명도를 사용하지 않는 네번째 해석은 이 문제를 해결하기 위해 채용될 수 있다. 이 때는 OTF를 매우 투명하게 지정하여 물체를 투명하게 만든다. 이 해석은 두께가 균일한 경우와 객체가 다른 객체를 싸고 있는 경우 자연스럽다. 그러나, 3.1절에서 상술한 바와 같이 이 방법은 제한된 경우에만 성공적으로 사용될 수 있으며, 다른 경우 사용이 매우 어려울 수 있다

### 3.3 알고리즘

본 논문의 방법은 기본적인 이단계 합성 알고리즘에  $\alpha$ -플래그를 도입하여 객체당 투명도를 구현한다.  $\alpha$ -플래그는 객체별로 할당되어 해당 객체의 누적 불투명도를 나타내며, 화소에 대한 객체 합성 여부를 판단하는데 사용된다.

그림 8은 이단계 투명도 부여 알고리즘을 설명한다. 알고리즘은 각 화소에 대해 지역값과 전역값을 나타내는 두개의 중간화소값과 객체 수만큼의  $\alpha$ -플래그를 가지고 있다.

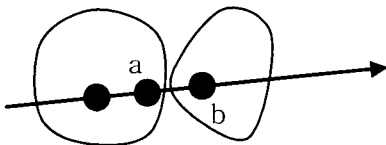


그림 8. 이단계 투명도 부여 알고리즘

샘플 a에 보인 바와 같이 이전 샘플과 현 샘플의 소속 객체가 동일한 경우 샘플값은 지역값에 합성되며, 샘플 b와 같이 이전 샘플과 소속 객체가 달라지면 ①지역값이 객체당의 불투명도를 적용하여 지역값을 전역값에 합성하고, ②이전 샘플이 속한 객체의  $\alpha$ -플래그 값을 표 1에 따라 갱신하고, ③지역값을 초기화하고, ④지역 불투명도를 새로 렌더링될 객체의  $\alpha$ -플래그로 갱신한다.

투명도에 대한 첫번째 해석 경우는 객체의 첫번째 세그먼트를 렌더링 한 후  $\alpha$ -플래그가 1.0이 되어 더 이상 해당 객체의 렌더링이 진행되지 않는다. 두번째 해석의 경우에는  $\alpha$ -플래그에 각 객체 세그먼트의 누적 불투명도값이 저장되어 해당객체가 불투명해질 때 렌더링이 종료된다. 세번째 해석의 경우에는  $\alpha$ -플래그가 항상 0.0이므로 전역불투명도가 지정된 문턱치를 넘지 않는 동안 모든 세그먼트가 렌더링 된다.

광선투사법으로 구현하는 경우 처리는 화소별로 일어나므로 두개의 중간화소 값 변수와 현재 객체를 추적하는 변수와 객체가 N개인 경우 N항목의  $\alpha$ -플래그가 있으면 된다. 객체순서 또는 시각정렬 슬라이스 순서 렌더링의 경우에는 한 화소의 처리가 끝나지 않은 상태에서 다음화소들이 처리되므로, 지역값과 전역값을 저장할 두개의 화면 크기 영상버퍼 및 이전 샘플의 소속 객체를 나타내는 영상버퍼를 두어야 하고, 객체별  $\alpha$ -플래그를 저장하기 위한 N개의 불투명도 영상 버퍼가 필요하다.

투명도 부여 알고리즘은 하나의 해석만을 기준으로 하여 만들어질 수도 있으나, 각 객체별로 4가지중 하나의 해석을 지정하는 것이 가능하다. 이 경우 각 객체에 대한 정보는 객체별 투명도 뿐 아니라 객체별 투명도 해석 방법도 담고 있어야 하며, 전체 알고리즘은 지역 중간값을 전역 중간값에 합성하는 시점에서만 객체별 해석 방법에 따라 작동하면 된다.

표 1. 투명도 해석에 따른  $\alpha$ -플래그 값

투명도 해석	$\alpha$ -플래그 값
1	1.0
2	지역 불투명도
3	0.0

### 4. 구현 및 결과

전절에 서술한 대로, 본 방법은 광선 투사법과 객체순서 구현, 하드웨어 구현에 모두 적합하다. 본 알고리즘을 대화식으로 사용될 수 있음을 보이기 위해 PC의 그래픽스 하드웨어를 사용하여 구현하였다. 구현은 시각정렬 슬라이스 순서의 대응 기하체(proxy geometry), 이른 깊이 테스트와 다중 렌더링 대상을 사용하였으며, ATI Radeon X800 Pro GPU상에서 작동하도록 하였다. 다른 연구에서는 1바이트 데이

터의 사용이 일반적이거나, 본 구현은 의료영상에서 보다 일반적인 2바이트 데이터를 직접 렌더링하도록 하였고, 화면의 크기는 512×512이다. 실험은 Pentium4 2.8GHz와 1GB RAM을 장착한 PC에서 수행하였다.



그림 9. 객체별 투명도를 사용한 렌더링

그림 9는 본 방법을 사용하여 반투명한 맥락정보를 제공한 예이다. 다음과 같은 세개의 객체가 사용되었다. (a) 분할된 혈관, (b) 혈관과 뼈 중 혈관 객체에 속하지 않은 부분, 그리고 (c) 혈관과 뼈를 제외한 나머지. 각 객체의 객체별 투명도는, (a) 객체는 불투명하며, (b) 객체는 약 45% (c) 객체는 약 60%를 투과시킨다.

각 객체의 전이함수는 (a)와 (b)가 동일하게 일반 뼈 또는 조영된 혈관을 표현하는 프리셋을 사용하였으며, (c)는 매우 투명하고 단일한 색상을 가지는 전이함수를 사용하였다. 객체 (b)는 객체별 투명도가 없을 경우 객체 (a)와 동일한 느낌의 렌더링이 되어 구분이 불가능하나, 객체별 투명도의 부여로 반투명하게 나타나, 혈관의 각 부위가 척추와 어떤 관계에

있는지를 명확히 보여준다. 객체 (c)는 매우 투명한 전이함수와 객체별 투명도를 동시에 사용하여 보다 효과적이고 통제가 용이한 맥락정보 제공이 가능함을 보여준다. 객체별 투명도가 없더라도 투명한 피부 맥락정보의 제공은 가능하겠지만, 그림에 나타난 것보다 더 진하여 혈관이나 뼈의 렌더링을 더 많이 가리게 된다.

표 2는 본 구현의 성능을 보여준다. 알고리즘이 기본적으로 객체수에 독립적이고, 객체별 투명도의 부여가 렌더링에 큰 부담을 주지 않으므로, 데이터의 내용이나 객체수와 관계없이 렌더링 시간이 일정함을 볼 수 있다. 구현이 최적화되지 않고, 가속 알고리즘을 사용하지 않은 상태에서 이단계 볼륨 렌더링의 하드웨어 구현[4]과 유사한 속도를 보여주며, 대화식 사용이 가능하다. 이 성능의 일부는 사용된 그래픽 하드웨어에 기인한 것으로 생각된다. ([4]는 ATI Radeon 9700 사용) 또한, 이단계 볼륨 렌더링에서 보인바와 같이 가속 알고리즘을 사용할 경우 수배 이상의 훨씬 향상된 성능을 보일 것으로 기대된다.

### 5. 토의 및 전망

분할된 객체의 볼륨 렌더링에서 효과적인 맥락 정보를 제공하기 위해 객체당 투명도를 부여하는 방법을 제시하였고, 객체당 투명도의 의미에 대해 자세히 살펴보았다. 객체당 투명도는 볼륨 렌더링의 이론에 대한 깊은 이해가 없는 일반 사용자가 쉽게 사용할 수 있는 도구를 제공하여 객체의 효과적인 해석을 가능하게 한다. 또한, 일반적으로 사용되는 볼륨의 미세한 회전과 함께, 객체 불투명도의 애니메이션은 객체간의 구분을 명확히 할 수 있는 추가적인 시간적 단서를 제공할 수 있다.

본 논문의 기본 알고리즘은 이단계 볼륨 렌더링[3]과 유사하므로, 본 연구를 확장하여 다양한 렌더링 기법을 제공하는 것은 매우 단순하다. 더욱이, 효율적인 다른 볼륨 셰이딩 기법[6,12]과 조합할 수도 있으므로, 사용자에게 보다 강력한 가시화 도구를 제공할 수 있다.

본 논문에서는 기초적인 하드웨어 구현을 제시하였으나, 성능의 향상을 위해 보다 정교한 알고리즘과 가속화 기법의 사용에 대한 연구가 필요하다. 이를 통해 대화식 사용에 충분한 속도를 얻을 수 있을 것으로 기대한다.

표 2. 하드웨어 구현의 성능(fps)

객체수	1개	2개	3개
Aneurism	2.09	-	-
Abdomen	1.96	1.95	1.95

(객체의 수에 따른 렌더링 시간의 변화(Aneurism은 256×256×256×8비트 데이터를 12비트로 스케일하였고, Abdomen은 256×256×256×12비트 데이터이다.)

또한, 스스로 가리는 객체에 대한 투명도 해석에 대해 보다 검토가 필요하다. 볼륨 렌더링의 물리적 해석과의 연관과, 투명도 해석 자체의 물리적 의미가 부여될 수 있다면 그 활용에 대한 더 깊은 통찰이 가능할 것이다. 또, 이론적인 검토 뿐 아니라 객체별 투명도가 제공하는 기능의 임상적인 유용성에 대해 보다 깊은 연구가 필요하다.

### 참 고 문 헌

[1] Levoy, M., "Display of Surfaces from Volume Data," *IEEE Computer Graphics & Applications*, 8(5), pp. 29-37, 1988.

[2] Drebin, R. A., Carpenter, L., and Hanrahan, P., "Volume Rendering," *Computer Graphics*, 22, 4, pp. 65-74, August 1988.

[3] Hauser, H., Mroz, L., Bischi, G. I., and Gröller, M. E., "Two-Level Volume Rendering," *IEEE Transactions on Visualization and Computer Graphics*, 7, 3, pp. 232-252, 2001.

[4] Hadwiger, M., Berger, C., and Hauser, H., "High-Quality Two-Level Volume Rendering of Segmented Data Sets on Consumer Graphics Hardware," *IEEE Visualization 2003 Proceedings*, 301-308, pp. 19-24 Oct. 2003.

[5] Bullitt, E. and Aylward, S. R., "Volume rendering of segmented image objects," *IEEE Transactions on Medical Imaging*, 21, 8, pp. 998-1002, Aug. 2002.

[6] Kniss, J., Premoze, S., Hansen, C., and Ebert, D., "Interactive translucent volume rendering and procedural modeling," *IEEE Visualization 2002 Proceedings*, pp. 109-116, 27 Oct.-1 Nov. 2002.

[7] Kniss, J., Kindlmann, G., and Hansen, C., "Multi-dimensional transfer functions for interactive volume rendering," *In Proceedings of IEEE Visualization 2001*, pp. 255-262. 2001.

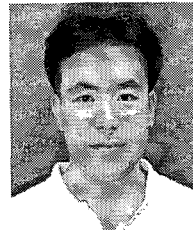
[8] He, T., Hong, L., Kaufman, A., and Pfister, H., "Generation of Transfer Functions with Stochastic Search Techniques," *Proc. IEEE Visualization 1996*, pp. 227-234, 1996.

[9] Marks, J. et al., "Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation," *ACM Computer Graphics(SIGGRAPH '97 Proc.)*, pp. 389-400, Aug. 1997.

[10] Bajaj, C.L., Pascucci, V., and Schikore, D.R., "The Contour Spectrum," *Proc. IEEE Visualization 1997*, pp. 167-173, 1997.

[11] Kindlmann, G. and Durkin, J. W., "Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering," *Proc. IEEE Symp. Volume Visualization*, pp. 79-86, 1998.

[12] Kniss, J., Premoze, S., Hansen, C., Shirley, P., and McPherson, A., "A model for volume lighting and modeling," *IEEE Transactions on Visualization and Computer Graphics*, 9, 2, pp. 150-162, April-June 2003.



정 동 균

1997년 2월 서울대학교 계산통계학과 졸업  
 1999년 2월 서울대학원 전산과학과 석사  
 2001년 5월~2002년 12월 (주)3D Med 연구원  
 2003년 1월~현재 (주)INFINITT

책임연구원

2003년 2월 서울대학원 컴퓨터공학과 박사수료  
 관심분야: 볼륨 렌더링, 의료 영상 가시화



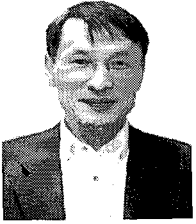
이 철 희

1991년 2월 서울대학교 컴퓨터공학과 졸업  
 1993년 2월 서울대학교 컴퓨터공학과 석사  
 1997년 8월~2000년 2월 (주)비트컴퓨터 기술연구소  
 1998년 2월 서울대학교 계산통계

학과 박사

2000년 3월~2002년 7월 (주)이엑스폼닷컴 대표이사  
 2002년 8월~2003년 11월 (주)인피니트 테크놀로지 기술연구소 수석연구원  
 2004년 3월~현재 서울벤처정보대학원대학교 컴퓨터응용기술학과 전임강사  
 2005년 3월~현재 (주)브이피랩 대표이사  
 관심분야: 볼륨 렌더링, 의료 영상 처리, 비디오 압축





신 영 길

1982년 서울대학교 계산통계학과 학사

1984년 서울대학교 계산통계학과 석사

1990년 미국 University of Southern California 전산학과 박사

1990년~1992년 경북대학교 전자계산학과 전임강사

1992년~현재 서울대학교 컴퓨터공학부 교수

관심분야: 의료 영상 처리, 볼륨 렌더링, 컴퓨터를 이용한 자동 진단