

# 효율적인 수퍼샘플링과 선-적분을 이용한 고화질 쉬어-왁 볼륨 렌더링

계희원<sup>†</sup>, 김태영<sup>\*\*</sup>

## 요 약

쉬어-왁 분해 렌더링은 볼륨 렌더링 방법 중 가장 빠르지만 영상의 화질이 좋지 않다는 단점이 있다. 본 연구에서는 쉬어-왁 렌더링의 빠른 속도를 유지하며 화질을 개선하는 방법을 제안한다. 화질 개선의 첫 번째 방법은 중간영상(intermediate image) 기반의 수퍼샘플링(supersampling) 기법이다. 객체 좌표와 영상 좌표간의 변환을 효율적으로 수행하여 임의 비율의 영상 확대를 빠르게 수행한다. 화질 개선의 두 번째 방법은 선-적분(pre-integrated) 렌더링을 사용하는 것이다. 기존의 선-적분 기반 쉬어-왁 렌더링은 빈-공간 도약(empty space leaping)을 하지 못하여 속도가 저하되는 문제가 있지만, 본 연구에서 제안하는 겹친 최소-최대 지도(overlapped min-max map) 자료구조를 사용하면 빈-공간 도약을 수행하여 속도 문제를 해결한다. 본 제안 방법을 통해 광선추적법(ray-tracing) 수준의 고화질 렌더링 영상을 빠른 시간에 생성할 수 있다.

## High-quality Shear-warp Volume Rendering Using Efficient Supersampling and Pre-integration Technique

Heewon Kye<sup>†</sup>, Tae-Young Kim<sup>\*\*</sup>

## ABSTRACT

As shear-warp volume rendering is the fastest rendering method among the software based approaches, image quality is not good as that of other high-quality rendering methods. In this paper, we propose two methods to improve the image quality of shear-warp volume rendering without sacrificing computational efficiency. First, supersampling is performed in intermediate image space. We propose an efficient method to transform between volume and image coordinates at the arbitrary ratio. Second, we utilize pre-integrated rendering technique for shear-warp rendering. We propose new data structure called overlapped min-max map. Using this structure, empty space leaping can be performed so that we can maintain the rendering speed even though pre-integrated rendering is applied. Consequently, shear-warp rendering can generate high-quality images comparable to those generated by the ray-casting without degrading speed.

**Key words:** Shear-warp Volume Rendering(쉬어-왁 볼륨 렌더링), Supersampling(수퍼샘플링), Pre-integrated Rendering(선-적분 렌더링), Empty-space Leaping(빈-공간 도약), High-quality Volume Rendering(고화질 볼륨 렌더링)

## 1. 서 론

볼륨 렌더링은 삼차원 볼륨 데이터를 가시화 하는

방법으로 의료영상과 과학적 가시화에 주로 사용된다 [1,2]. 전통적인 볼륨 렌더링은 워크스테이션과 같은 고급 장비가 필요했으며 결과를 생성하는데 오랜 시

※ 교신저자(Corresponding Author) : 계희원, 주소 : 서울시 관악구 신림9동 산 56-1(151-744), 전화 : 02)880-1860, FAX : 02)889-1756, E-mail : kuei@cglab.snu.ac.kr  
접수일 : 2006년 4월 10일, 완료일 : 2006년 5월 24일

<sup>†</sup> 서울대학교 컴퓨터연구소 연구원

<sup>\*\*</sup> 서경대학교 컴퓨터공학과 조교수  
(E-mail : tykim@skuniv.ac.kr)

※ 본 연구는 한국과학재단 우수 여성과학자 도약지원 연구(과제번호:R04-2002-000-20031-0)의 연구비 지원으로 수행되었음

간이 소모되었다. 이후 하드웨어의 발전과 새로운 알고리즘의 개발[3-5]로 범용 PC에서도 만족스러운 결과를 얻을 수 있게 되었다. 그 중 쉬어-왓 렌더링[6,7]은 대표적인 볼륨 렌더링 기법으로, 그래픽스 하드웨어를 사용하는 경우[8,9]를 제외하면 가장 빠르다고 알려져 있다. 그러나 쉬어-왓 렌더링의 화질은 뛰어난 편이 아니기 때문에 고화질의 영상을 필요로 하는 의료영상 등의 응용분야에 활용가치가 떨어진다.

Lacroute는 쉬어-왓 렌더링의 화질이 높지 않은 이유를 다음과 같이 설명하였다[7]. 첫째, 재샘플링(resampling)에서 화질 손실이 발생한다. 쉬어-왓 렌더링은 샘플의 누적 결과를 중간 영상(intermediate image)에 저장하고 이를 2차원 변환(warping)하여 최종 영상을 생성한다. 중간 영상의 해상도(image resolution)는 최종 영상과 무관하게 객체 해상도(volume resolution)에 의해 결정되므로, 최종 영상 해상도가 매우 높은 경우, 확대(magnification) 변환에 의해 화질이 저하된다. 객체 해상도가 과거에 비해 매우 높아지기는 했으나 영상 해상도 또한 높아졌고, 볼륨의 일부를 확대 관찰할 경우도 있어 이는 여전히 해결되어야 할 문제이다. 둘째, 볼륨 샘플링이 이선형 보간(bilinear interpolation)이기 때문에 삼선형 보간(trilinear interpolation)하는 다른 고화질 렌더링에 비해 정보가 손실된다. 쉬어-왓 렌더링은 주 시각(major viewing axis) 방향에 대해서는 0차 보간(nearest neighbor sampling)을 한다. 관찰자의 시선 방향이 객체 좌표축과 기울어진 경우, 시선 방향의 샘플 간격이 길어지게 되어 계단현상이 두드러진다.

본 연구는 쉬어-왓 렌더링의 화질을 개선하여 광선추적법[3] 수준의 고화질 영상을 생성하고자 한다. 동시에 쉬어-왓 렌더링의 장점인 빠른 속도는 유지하도록 한다. 본 연구의 화질 향상 방법은 두 단계로 진행된다. 첫 번째 단계는, 효율적인 영상영역(image space) 수퍼샘플링 방법이다. 제안 방법은 렌더링을 중간영상 주사선(scanline) 방향으로 진행하여, 객체 좌표계(object coordinate)와 영상 좌표계(image coordinate) 사이의 변환 비용을 줄인다. 그리하여 효율적으로 임의의 비율(arbitrary ratio)의 수퍼샘플링을 할 수 있다. 두 번째 방법은 선-적분 볼륨 렌더링(pre-integration volume rendering)[10]을 적용하는 것이다. 선-적분 볼륨 렌더링은 객체영역(object space) 수퍼샘플링을 대신하는 효과가 있어 화질 향

상을 얻을 수 있다. 한편, 기존의 선-적분 렌더링은 후-분류(post classification) 렌더링이므로 쉬어-왓 렌더링에 그대로 적용하면 속도가 저하되는 문제가 있다. 이 문제를 해결하기 위하여 본 연구는, 겹친 최소-최대 지도(overlapped min-max map) 자료구조를 제안한다. 제안 자료구조를 이용하면 빈-공간 도약을 효율적으로 수행할 수 있기 때문에 속도 저하 문제가 해결할 수 있다.

두 가지 제안 방법은 각각 영상영역 수퍼샘플링과 객체영역 수퍼샘플링의 효과를 보이기 때문에, 서로 보완하여 고화질 영상을 생성하는데 기여한다. 이후 논문의 구성은 다음과 같다. 2장에서는 관련된 기존 연구에 대해 살펴보고, 3장에서는 효율적인 영상영역 수퍼샘플링 방법을 소개한다. 4장에서는 선-적분 기반의 쉬어-왓 렌더링을 위한 공간 탐색법을 제안한다. 5장에서 실험 결과를 보이고 6장에서 결론을 맺는다.

## 2. 관련연구

### 2.1 쉬어-왓 볼륨 렌더링

쉬어-왓 볼륨 렌더링은 [그림 1]과 같이 시각행렬을 전치 행렬, 쉬어 행렬, 와핑 행렬로 분해하는 기법이다. 전통적 쉬어-왓 볼륨 렌더링이 속도가 빠른 이유는 다음과 같다[7]. 객체 좌표계로부터 영상 좌표계로 변환이 간단하다. 또한, RLE(run length encoded) 볼륨을 이용하여 비어있지 않은 복셀을 중간 영상 좌표계를 따라 연속적으로 읽을 수 있어 메모리 참조의 효율성이 높다. 마지막으로, 중간영상 좌표계의 불투명한 화소들을 기억하고 건너뛸 수 있기 때문에, 효율적인 조기 광선 종료(early ray termination)를 수행한다. 그러나, 앞서 설명한 바와 같이, 고화질 영상을 생성하기 어렵기 때문에, 객체의 해상도가 증

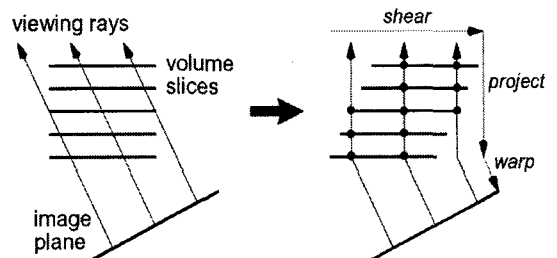


그림 1. 쉬어-왓 볼륨 렌더링(6)의 개념도

분히 크거나 확대가 많이 일어나지 않는 응용분야에 적합하였다.

## 2.2 고품질 볼륨 렌더링

볼륨 렌더링의 화질을 향상시키기 위해 일반적으로 수퍼샘플링을 사용한다. 볼륨 렌더링의 수퍼샘플링은 영상영역 수퍼샘플링과 객체영역 수퍼샘플링으로 나눌 수 있다. 영상영역 수퍼샘플링은 영상의 각 화소에서 보내는 광선의 수를 증가시키는 것이고, 객체영역 수퍼샘플링은 광선을 따라가면서 샘플링 간격을 좁게 하여 한 광선 당 더 많은 수의 샘플을 합성하는 방법이다[11]. 따라서 영상영역 수퍼샘플링을 통해 확대 변환에 의한 화질 저하를 개선할 수 있으며, 객체영역 수퍼샘플링을 할 수 있다면 이선형 보간으로 인한 정보손실을 개선할 수 있다. 그러나 쉬어-왓 렌더링의 영상영역 수퍼샘플링은 효율적 처리를 위한 특별한 주의가 필요하며, 객체영역 수퍼샘플링을 하려면 삼선형 보간을 해야 하므로 알고리즘을 대폭 고쳐야 하는 어려움이 있다[7].

렌더링 과정에서 샘플의 색과 불투명도를 결정하는 방법을 분류(classification)라 하며 선-분류(pre-classification)와 후-분류(post-classification) 방식이 있다. 선-분류 렌더링은 각 복셀에 대해 미리 분류해 색과 불투명도를 결정한 후, 샘플 위치에서 색과 불투명도 값을 보간한다. 반대로 후-분류 렌더링은 밀도값을 보간으로 먼저 얻은 후 분류한다. 선-분류 렌더링은 영상의 확대가 많이 일어나는 경우 영상이 뿌옇게(blur) 되고, 대화적 분류(interactive classification) 환경에서 매우 느리기 때문에 본 연구에서는 고려하지 않는다. 한편, 선-적분 렌더링[10]은 후-분류 함수를 미리 적분하는 방법으로 객체영역 수퍼샘플링의 효과를 얻을 수 있다.

### 2.2.1 수퍼샘플링

볼륨 렌더링 방법 중 일부는 영상영역 수퍼샘플링을 손쉽게 적용할 수 있다. 광선추적법의 경우 영상의 화소를 증가시켜 간단히 수퍼샘플링을 수행할 수 있다. 스플래팅 기반의 볼륨 렌더링[12]은 기저함수의 영상 해상도를 높이는 방법을 사용한다. 그러나 쉬어-왓 볼륨 렌더링은 빠른 속도를 주안점으로 두었기 때문에 많은 쉬어-왓 구현은 수퍼샘플링을 포함하고 있지 않다[13,14]. 원근 투영 쉬어-왓 렌더링에 확대 연

산이 고려되어 있지만[7], 선-분류(pre-classification) 스플래팅 함수를 이용하기 때문에 화질이 나빠지는 단점이 있다. Schulze 등은 Lacroute의 연구를 확장하였는데, 확대 변환(scaling matrix)를 개선하여 원근 투영 쉬어-왓 렌더링에 영상영역에서 임의 비율의 부표본(subsampling)을 가능하게 하였다[15].

평행 투영 쉬어-왓 분해의 경우, Koo 등은 한 화소당 복수 원형(template)을 만들어 수퍼샘플링을 하였고[11], Kim 등은 비균등 볼륨데이터에 대해 한 방향에 대한 수퍼샘플링 방법을 제안하였다[16]. 이들 방법은 확대 비율은 임의의 비율이 아닌 정수로만 제한되었는데, 샘플링시 연관성을 확보하여 빠른 렌더링 속도를 유지하기 위함이었다.

한편, 쉬어-왓 볼륨 렌더링의 객체영역 수퍼샘플링은 삼선형 보간을 해야 하므로 알고리즘이 복잡해진다. Sweeney 등은 영상영역과 객체영역 모두에서 수퍼샘플링을 시도하였다[17]. 이들은 RLE 볼륨에 분류된 색상정보 대신 밀도와 음영을 저장하여, RLE 볼륨을 사용하면서도 후-분류(post-classification) 렌더링을 하였다. 하지만 알고리즘이 복잡하여 객체영역 수퍼샘플링에서 2배 가까이 렌더링 성능이 저하되는 문제가 있으며, 후-분류 렌더링임에도 불구하고 대화적 분류를 하지 못한다는 단점이 지적된다. 볼륨 렌더링 가속화 하드웨어 VolumePro는 쉬어-왓 분해 기반의 광선추적법을 사용하며 수퍼샘플링을 지원하지만 전용 하드웨어를 요구한다는 점에서 차이가 있다[9].

수퍼샘플링은 인접한 공간 좌표에서 여러 개의 샘플을 얻게 되므로 메모리 참조 연관성이 높아져 캐시 효율이 향상되는 효과가 있다. 샘플링 위치가 불규칙하여 메모리 참조 연관성이 떨어지는 경우라면, 한번 샘플링한 결과를 별도의 버퍼에 복사해두고 캐시 메모리처럼 이용하여 성능을 향상시킬 수 있다. 광선추적법 응용으로는 Knittel 등의 연구[18]가 있으며 쉬어-왓 기반 렌더링에서는 광도 버퍼를 사용하는 예가 있었다[16].

### 2.2.2 선-적분 볼륨 렌더링

선-적분 볼륨 렌더링은 볼륨 광선 적분을 효과적으로 구현하는 렌더링 방법이다. [그림 2]와 같이, 광선상의 연속한 샘플 값이 선형성을 가지고 있다고 가정하고, 광선 조각 양 끝점의 값을 이용하여, 그 구간의 색과 투명도의 적분 값을 참조표(lookup ta-

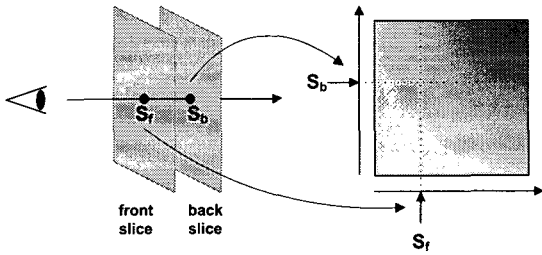


그림 2. 선-적분 볼륨 렌더링은 광선을 따라 연속한 두 샘플 값을 참조표(lookup table)의 색인으로 삼아, 구간의 색과 불투명도를 얻는다.

ble) 또는 선-적분 표로 미리 적분하여 구성해 놓는다. 렌더링시 연속한 샘플 값을 선-적분 표의 색인으로 삼아 참조표를 읽어 색과 투명도를 얻는다. 참조표에 저장된 값은 많은 샘플링 결과의 누적 값을 반영하므로 적은 샘플링 비율로도 높은 화질을 얻을 수 있다는 장점이 있다. 객체영역 샘플링 길이 증가로 인한 화질 저하를 선-적분 렌더링으로 개선하는 연구는 활발하게 진행되고 있다[10,19-21].

이선형 보간을 하는 쉬어-왓 렌더링은, 광선추적법에 비해 최대 1.7배 가량 객체영역에서 샘플링 길이가 길어지게 되므로, Schulze 등이 지적한 바와 같이 선-적분 렌더링을 적용하여 화질을 향상시킬 수 있다[19]. 그러나 이들의 방법은 빈-공간 도약을 하지 못하여 속도가 많이 저하되는 단점이 있다.

### 3. 쉬어-왓 분해 기반 영상영역 슈퍼샘플링

쉬어-왓 렌더링은 효율적인 계산을 위해 중간영상의 화소 크기가 복셀 크기와 같다고 가정한다[6]. 따라서 볼륨을 확대 렌더링 하면 중간영상의 화소 하나가 결과 영상의 많은 화소에 대응되므로 화질이 저하된다. 이때 중간영상 해상도를 높여, 중간영상 화소가 결과 영상 화소와 비슷한 크기가 되도록 하면 화질저하 문제를 피할 수 있다[15,17]. 그러나 슈퍼샘플링을 쉬어-왓 렌더링에 적용하면 알고리즘의 복잡도가 증가하여 일반적으로 속도가 저하되는 문제가 있다.

쉬어-왓 렌더링의 가속화 방법은 빈-공간 도약과 조기 광선 종료이다. 빈-공간 도약은 객체 좌표계의 크기로 계산하고, 조기광선종료는 중간영상 좌표계의 크기로 계산한다. 게다가 복셀의 크기와 중간영상의 화소 크기가 다르면 샘플링시 이선형 보간에서

가중치가 계속 변화하게 된다. 따라서 두 좌표계 사이의 빠른 변환이 필수적이다. 쉬어-왓 렌더링의 경우 슈퍼샘플링의 비율을 정수로 근사하여 규칙성을 이용할 수 있으나[11,16], 슈퍼샘플링 비율의 값이 변화하는 순간 화질의 변화가 생긴다는 점과 슈퍼샘플링 비율이 매우 커졌을 때 규칙성이 감소한다는 점이 단점으로 지적된다.

빠른 좌표변환을 위해 본 연구는 중간영상 좌표계를 기준으로 렌더링을 수행한다. 쉬어-왓 렌더링의 중간영상 주사선과 볼륨 주사선은 평행하므로, 하나의 중간영상 주사선에 대해 이선형 보간의  $y$  가중치는 일정하다. 따라서 중간영상 주사선을 기준으로 렌더링 하면,  $x$  좌표의 변환만을 고려할 수 있다는 장점이 있다. 이를 가상 코드로 설명하면 [그림 3]과 같다.

볼륨을 구성하는 각각의 이차원 영상의 영역을 중간영상 좌표계로 변환하면 중간영상 좌표에서의 범위를 얻을 수 있다. 범위에 속하는 각 중간영상 주사선에 대해 객체 좌표로 역 투영하여 객체 좌표의 시작점과 끝점을 얻을 수 있으며  $y$  가중치를 얻을 수 있다. 이후 일반적인 쉬어-왓 렌더링과 마찬가지로 한 주사선에 대한 연산을 수행한다. 이때  $x$  방향의 확대비율을 고려해야 하며 이를 [그림 4]에서 예를 들어 설명한다. 단, 설명의 편의를 위해 일반성을 해치지 않는 범위에서  $y$  방향의 보간은 생략하였고  $x$  방향의 보간만을 가정하였다.

하나의 중간영상 주사선의 렌더링 과정은 다음과 같다. 복셀의 시작 위치에서 중간영상으로 투사[그림 4-1]하여 첫 번째 렌더링 할 화소를 찾는다[그림 4-2]. 샘플링이 필요한 경우 객체 좌표를 얻어야 한다[그림 4-3]. 불투명한 화소를 만나면 건너뛰며[그림 4-4] 투명한 화소에 대해서만 샘플링을 수행하여 영상에 누적해 나간다. 샘플링 할 위치의 인접 복셀

```

Shear_Warp
for each slice  $z_v \in [0, \text{number of volume slice}-1]$ 
( $y_{0v}, y_{1v}$ )  $\leftarrow$  Get_Image_Scanline_Range ( $z_v$ )
for each scanline  $y_i \in \text{Intermediate_Scanline_Range}$ 
 $y_v \leftarrow \text{Inverse Mapping}(y_i)$ 
( $x_{0v}, x_{1v}$ )  $\leftarrow$  Get_Voxel_Range ( $y_v$ )
Process_One_Intermediate_Scanline( $x_{0v}, x_{1v}$ )
    
```

그림 3. 중간영상 기준의 렌더링 가상코드. 각 슬라이스에 대해 중간 영상에 차지하는 범위를 얻고 각 중간영상의 주사선에 대한 렌더링을 수행한다.

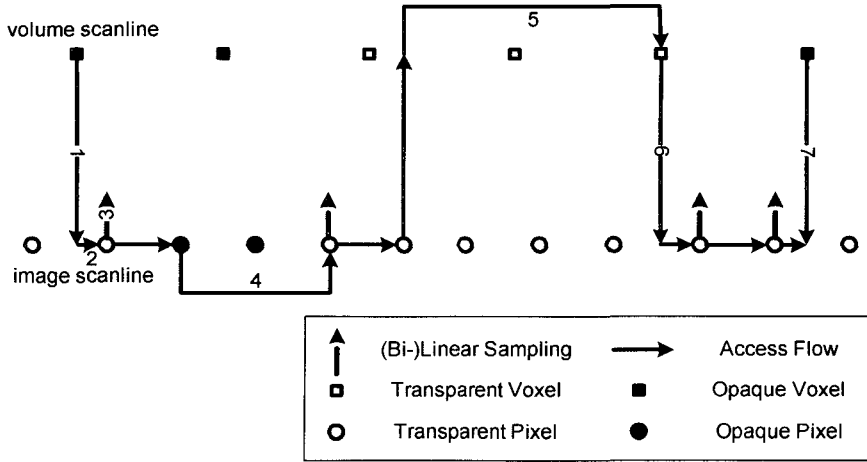


그림 4. 중간영상 주사선의 렌더링 예시. 볼륨 주사선의 범위(화살표 1, 7)에 해당하는 중간영상 주사선을 기준으로 렌더링을 진행한다. 빈-공간을 건너뛰는 경우(화살표 5)를 제외하면 항상 중간영상 좌표를 유지한다.

이 빈-공간일 경우 객체 좌표에서 건너뛴다[그림 4-5]. 연속된 투명한 복셀 중 마지막 점에서 중간영상 좌표로 변환한다[그림 4-6]. 주사선의 마지막 복셀 위치를 중간영상으로 투사한 지점[그림 4-7]까지 렌더링을 수행한다. 정리된 가상 코드를 [그림 5]에 보였다.

중간영상 좌표를 기반으로 순환하며 영상좌표와 객체 좌표를 분리하면 기존의 쉬어-왓 렌더링과 비슷한 형태를 가지게 된다. 기존 쉬어-왓 렌더링과 비교하여 추가된 것은 객체 좌표에서 영상좌표로 변환하는 Mapping과 영상좌표에서 객체 좌표로 변환하는 Inverse Mapping과정이다. 따라서 본 알고리즘은 기존 알고리즘을 크게 변경하지 않고 적용 가능하다.

```

Process_One_Intermediate_Scanline(x0v, x1v)
pixel_start = RoundingUp( Mapping(x0v) )
pixel_end = RoundingOff( Mapping(x1v) )
for (xi ← pixel_start; xi ≤ pixel_end; xi ← xi + 1)
if(Is Opaque Pixel (xi))
    xi ← xi + opaque_pixel_length
    xv ← Inverse Mapping(xi)
if(Is Transparent Voxel(xv))
    xv ← xv + transparent_voxel_length
    xi ← Mapping(xv)
Compositing (xi, Sampling(xv))
    
```

그림 5. 중간영상 주사선의 렌더링 가상코드. 객체 좌표에서 중간영상 좌표로 변환하는 Mapping과 중간영상 좌표에서 객체 좌표로 변환하는 Inverse Mapping이 추가된 것을 제외하면 기존 쉬어-왓 렌더링과 유사한 형태이다.

동일한 볼륨 슬라이스(slice)에서 입력에 대한 결과 값이 변화하지 않으므로, 새로운 슬라이스가 시작될 때 각각 1차원 참조표를 생성하면 적은 비용으로 참조표를 생성할 수 있고, 좌표 변환도 각각 참조표를 한번만 참조하여 빨리 수행할 수 있다. 즉, 임의의 슈퍼샘플링은 확대 변환이므로, 슈퍼샘플링을 포함하여 객체 좌표 (x, y, z)에서 중간영상 좌표 (i, j, k)로의 변환 행렬을  $M_{v\_to\_i} = M_{supersample} * M_{shear}$ 로 표현할 수 있다(여기서,  $M_{supersample}$ 은 확대 행렬,  $M_{shear}$ 은 쉬어 분해 행렬이다). 여기에 1차원 참조표  $T_x$ ,  $T_y$ ,  $T_z$ 를 생성하여 식(1)과 같이 표현할 수 있다[24].

$$\begin{bmatrix} i \\ j \\ k \\ 1 \end{bmatrix} = \mathbf{M}_{v\_to\_i} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \\
 = \mathbf{M}_{v\_to\_i} \begin{bmatrix} x \\ 0 \\ 0 \\ 1 \end{bmatrix} + \mathbf{M}_{v\_to\_i} \begin{bmatrix} 0 \\ y \\ 0 \\ 1 \end{bmatrix} + \mathbf{M}_{v\_to\_i} \begin{bmatrix} 0 \\ 0 \\ z \\ 1 \end{bmatrix} \\
 = \mathbf{T}_x[x] + \mathbf{T}_y[y] + \mathbf{T}_z[z] \tag{1}$$

샘플의 투영은 객체 좌표 (x, y, z)에서 중간영상 좌표 (i, j)를 얻는 과정이므로 식(2)을 얻는다. 한 주사선의 렌더링 과정에서 y, z는 변하지 않으므로  $T_x[x].x$ 만 갱신되면 새로운 투영좌표 (i, j)를 얻을 수 있다. 따라서 [그림 5]의 Mapping과정은 한번의 참조와 덧셈으로 수행된다. 역으로 [그림 5]의 Inverse Mapping과정은  $M_{v\_to\_i}$ 의 역행렬을 이용하여 마찬가지로

지 과정을 거치면 역시 한번의 참조와 덧셈으로 수행할 수 있다.

$$\begin{aligned}
 i &= T_x[x].x + T_y[y].x + T_z[z].x \\
 j &= T_y[y].y + T_z[z].y \\
 (\ominus T_x[x].y &= 0)
 \end{aligned}
 \tag{2}$$

[그림 5]를 [그림 3]과 연결하면 임의의 확대 비율로 영상공간 수퍼샘플링 가능하며 좌표 변환의 비율이 적은 렌더링 알고리즘이 완성된다. 또한 영상의 확대 비율이 크다고 하더라도; 영상의 크기(화소 수)는 모니터와 같은 물리적인 장치에 종속적이므로 일반적으로 변화가 없다. 따라서 시각-클리핑(view clipping)을 적용하면 볼륨의 일부를 렌더링 하지 않아도 되므로 성능이 향상된다. 시각-클리핑은 [그림 5]의 Get\_Image\_Scanline\_Range와 Get\_Voxel\_Range를 수정하여 쉽게 구현할 수 있어 추가적인 속도 개선 효과를 얻을 수 있다.

#### 4. 선-적분 기반의 쉬어-왓 렌더링

쉬어-왓 렌더링의 화질문제 중 하나인 샘플 길이 증가에 대한 해법으로 선-적분 렌더링을 도입하는 것은 자연스러운 일이다. Engel 등의 연구[10]는 선-적분 렌더링이 두 배 가량의 샘플 길이 증가에도 높은 화질을 보이는 것을 보였다. 그러나 쉬어-왓 렌더링의 장점인 빠른 속도를 유지하며 선-적분 렌더링을 도입하는 방법은 많은 고려가 필요하다.

일반적인 쉬어-왓 렌더링은 RLE 볼륨을 사용하기 위해 선-분류를 적용하였다. 그러나 선-적분 렌더링은 광선추적법과 마찬가지로, 보간된 볼륨 밀도 값을 참조표의 색인으로 하여 색과 불투명도를 얻기 때문에, 후-분류 방식이다. 따라서 쉬어-왓 렌더링에도 후-분류를 적용해야 하며 이는 미리 계산해둔 RLE 볼륨을 사용할 수 없음을 의미한다. RLE 볼륨을 이용한 빈-공간 탐색을 할 수 없다면 쉬어-왓 렌더링의 장점인 빠른 렌더링 속도를 전혀 살리지 못하는 것이 된다. 실제로 Schulze 등이 쉬어-왓 렌더링에 선-적분 렌더링을 도입하였으나 빈-공간 탐색을 하지 못하여, 그들의 렌더링 성능은 대화적인 속도에 크게 미치지 못한다[19]. 이번 절에서는 쉬어-왓 렌더링의 빠른 분류를 위해 제안된 블록-스캔라인 최소-최대 지도[22]와, 빠른 등위면 렌더링을 위해 제

안된 LF(loose fitting) 최소-최대 지도 [23]를 결합하여 선-적분 기반의 쉬어-왓 렌더링에 적합한 빠른 빈-공간 탐색 방법을 제안한다.

RLE 볼륨을 사용하지 못하는 경우 대화적 분류를 위해 Lacroute 등은 팔진 나무(octree) 기반의 빠른 분류 방법을 제안하였다. 볼륨을 일정한 크기를 가진 블록으로 구분하고 각 블록의 영역에 포함되는 복셀의 최소값과 최대값을 전처리 과정에서 계산해 둔다. 렌더링시 누적 영역 지도(summed area table) 등을 통해 빈-공간으로 판단한 블록영역에 대해 샘플링 없이 빠르게 건너뛸 수 있다[6]. 일반적으로 최소-최대 팔진 나무는 효율적인 계층적 자료구조이나 쉬어-왓 렌더링은 주사선 기반의 렌더링이므로 한 주사선의 투명도를 판단할 때 비효율적이다. 이에 대응하기 위해 제안된 블록-스캔라인 최소-최대 지도[22]는 주사선의 진행방향으로 길쭉한 모양을 갖는다. 일반적인 블록이  $n \times n \times n$  복셀을 포함한다면 블록-스캔라인 최소-최대 지도는  $n \times 1 \times 1$  복셀을 포함한다.

이선형 보간과 선-적분을 고려하면, 렌더링시 동시에 참조해야 할 주사선은 [그림 6a]에서 보는 것과 같이 모두 4개이다. 예를 들어 [그림 6a]의 1번, 2번, 6번, 7번의 4개의 블록 주사선이 모두 투명하면 (1,2,6,7)에 해당하는 영상 주사선을 건너뛸 수 있다. 4개의 블록-스캔라인의 최소-최대값을 얻어 각각 누적 영역 지도 등을 통해 투명도 검사를 수행하고 모두 투명한 경우 건너뛴다. 그러나 이러한 방법은 다수의 메모리 참조와 비교연산을 필요로 하므로 복잡하고 비효율적이다.

Choi 등의 LF 최소-최대 지도[23]는 빠른 등위면 렌더링을 위한 자료구조이다. 빈-공간 탐색시 광선

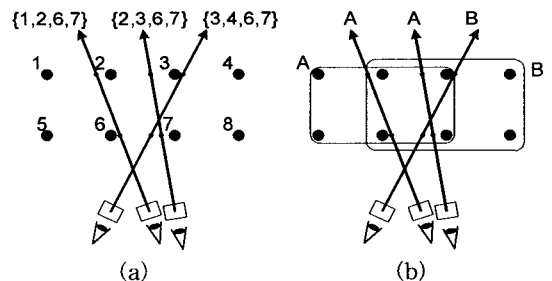


그림 6. 선-적분 기반 쉬어-왓 렌더링의 개념도. 기존의 주사선 최소-최대 지도(a)에서 한 샘플의 투명도는 4개 복셀의 투명도 검사로 결정되는 반면, 제안한 겹친 최소-최대 지도(b)에서는 한 블록의 투명도 검사로 결정되어 효율적이다.

-상자 교차연산(ray-box intersection)을 없애고 상수 거리만큼 진행하기 위해 제안되었다. 기존의 최소-최대 지도의 단말은 점( $nx, ny, nz$ )과 점( $nx + n, ny + n, nz + n$ )을 맞모금으로 하는 육면체로  $n \times n \times n$  크기인 것에 비해, LF 최소-최대 지도의 단말은 6방향으로 각각  $\beta$ 만큼 확장하여 점( $nx - \beta, ny - \beta, nz - \beta$ )과 점( $nx + n + \beta, ny + n + \beta, nz + n + \beta$ )을 맞모금으로 하는  $(n + 2\beta) \times (n + 2\beta) \times (n + 2\beta)$  크기이다. 본 연구에서는 LF 최소-최대 지도를 선-적분 렌더링에 맞도록 수정하였다.

광선의 진행방향, 즉  $z$ 방향으로는 선-적분 샘플링을 위해 복셀간 간격 1만큼의 길이를 두었다. 그리고 주사선의 진행 방향, 즉  $x$ 방향으로는 블록-스캔라인 최소-최대 지도의 길이  $n$ 에 선-적분 샘플링을 위해 1이 더 추가해서  $n + 1$ 의 길이를 설정하였다. 마지막으로 슬라이스의 진행방향, 즉  $y$ 방향으로는 이선형 보간을 위한 길이 1에 선-적분을 위한 1을 추가해서 2의 길이로 정하였다. 정리하면, 본 연구에서 제안하는 블록-스캔라인 최소-최대 지도는 점( $nx, y, z$ )과 점( $nx + n + 1, y + 2, z + 1$ )을 맞모금으로 하는  $(n + 1) \times 2 \times 1$  크기이다(포함하는 복셀은  $(n + 2) \times 3 \times 2$  이다). 제안 방법을 사용한 투명도 검사는 [그림 6b]에서 볼 수 있듯이 A 또는 B 블록의 투명도 검사 한번으로 종료되므로 투명도 검사 속도를 대폭 향상시킬 수 있다.

부가적으로 쉬어-wap 렌더링의 경우 광선의 진행 방향이 45도를 넘게 되면 주-시각축이 바뀌어, 광선의 진행 방향이 45도를 넘게 되지 않는다. [그림 6]의 광선의 기울기는 45도보다 작게 되므로, 제안한 블록의 크기 밖으로 광선이 벗어나 영상에 구멍(hole)을 만들거나 화질을 저하하는 일은 없다. 그리고 블록의 참조 위치는 광선의 방향에 따라 결정할 수 있는데, 예를 들어 [그림 6b]에서 광선의 진행 방향이 왼쪽이면 A 블록을 참조하고 오른쪽이면 B 블록을 참조하면 된다. 이를 3차원으로 확장하면 광선의 진행에 따른 샘플  $x, y$  좌표의 증가, 감소의 4가지 경우로 일반화 할 수 있다. 평행 투영 렌더링에서 광선의 진행 방향은 모든 광선이 일정하므로, 블록의 주소계산을 효율적으로 할 수 있다.

## 5. 실험

이번 장에서는 본 연구에서 제안한 방법을 통해

생성된 영상의 화질과 생성 속도에 대해 실험 결과를 보인다. 먼저 효율적인 영상영역 슈퍼샘플링으로 인한 화질 개선과 속도 변화를 알아본다. 다음으로 선-적분 기반 렌더링의 화질과 제안 방법인 빈-공간 탐색법을 통한 속도 변화를 분석한다. 본 실험은 1GB 메모리를 가진 2.4GHz의 Pentium4 개인용 컴퓨터에서 수행하였으며 실험에 사용한 데이터는 [표 1]과 같다. AngioS, AngioB 데이터는 모두 뇌혈관 조영(brain x-ray angio) 데이터이고 BrainPET 데이터는 뇌의 양전자 단층촬영(PET) 데이터이다.

### 5.1 영상영역 슈퍼샘플링

제안한 쉬어-wap 렌더링의 슈퍼샘플링으로 기존의 쉬어-wap 렌더링에 비해 향상된 화질의 영상을 얻을 수 있다(그림 7). 이미 알려진 바와 같이 기존 쉬어-wap 렌더링은 화면 확대 시 주변 영상을 생성하나(그림 7a), 슈퍼샘플링을 통해 깨끗한 영상을 얻는다(그림 7b).

슈퍼샘플링은 샘플링을 많이 하여 화질을 향상시키므로 렌더링 시간이 늘어나게 된다. 중요한 문제는 쉬어-wap 렌더링에 슈퍼샘플링을 적용할 때 두드러진 속도 저하가 발생하는지의 여부이다. Lacroute는 쉬어-wap 렌더링의 슈퍼샘플링은 알고리즘을 변경하는 비용이 크기 때문에 전처리(pre-processing)로 슈퍼샘플링 내용량 볼륨을 생성하고, 그 볼륨으로 기존의 쉬어-wap 렌더링을 적용 하는 것이 더 좋은 방법이라고 주장하였다[7]. 그러나 본 연구는 효율적인 좌표계 변환을 통해 슈퍼샘플링이 충분히 빠른 속도로 영상을 생성하는 것을 보이고 있다. [표 2]에 슈퍼샘플링을 통한 렌더링 시간을 제시하였다. 슈퍼샘플링을 수행하면 렌더링 시간이 오래 걸림을 확인할 수 있다[표 2](B). 그러나 슈퍼샘플링시 렌더링되는 화소수가 많은 것을 고려하여, 한 화소를 생성하는데 걸린 시간을 계산해보면 그 차이는 많지 않음을 알 수 있다. [표 2](C). 그리고 확대 비율이 1.13에서 2.26

표 1. 실험에 사용한 볼륨 데이터

볼륨 데이터	영상 크기	복셀 형태	저장공간
AngioS	256x256x256	8bit	16MB
AngioB	512x512x512	8bit	128MB
BrainPET	128x128x84	16bit	21.7MB

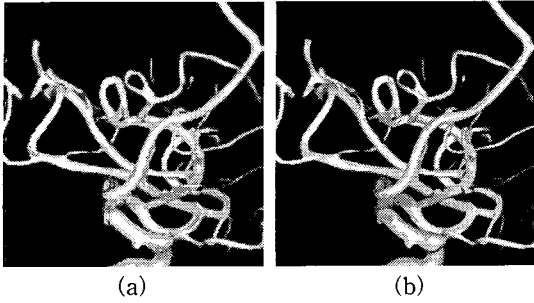


그림 7. 영상영역 수퍼샘플링을 하지 않은 영상(a)과 영상영역 수퍼샘플링 렌더링을 한 영상(b)의 비교

표 2. 수퍼샘플링을 통한 렌더링 시간 변화. 수퍼샘플링을 하게 되면 절대적인 시간은 증가하지만 화소당 시간을 비교해 보면 차이가 크지 않다.

알고리즘	수퍼샘플링 비율(A)	시간 (B)	화소당 상대적 시간(C=B/A)
광선 추적법	1.28	35s	27.3
쉬어-왓	1 (수퍼샘플링 없음)	0.989s	0.989
제안방법	1.13	1.26s	1.16
제안방법	2.26	2.40s	1.06

으로 커지면 화소당 상대적 시간이 1.16에서 1.06으로 개선되는 것을 확인 할 수 있다. 객체순서 볼륨 렌더링의 수퍼샘플링은 반복해서 동일한 볼륨 메모리를 참조하기 때문에, 메모리 참조 효율성이 높이는 효과가 있기 때문이다. 효율적인 좌표 변환으로 알고리즘의 비용이 최소화 되었기 때문에 수퍼샘플링 비율이 클수록 화소당 렌더링 시간은 전체적으로 향상 되게 된다.

### 5.2 선-적분 볼륨 렌더링

선-적분 렌더링의 화질을 알아보기 위해 [그림 8]을 제시하고 있다. [그림 8]은 영상영역 수퍼샘플링이 적용된 렌더링 결과의 확대 영상이다. [그림 8]의 (a), (b), (c)는 각각 광선 방향 샘플 간격을 1과 0.5, 0.2로 조정하여 실험한 광선추적법의 결과영상이다. (d)는 본 렌더링 기법인 선-적분을 적용한 쉬어-왓 볼륨 렌더링의 결과영상이다. 객체방향 샘플링 비율이 낮은 경우 [그림 8a]와 같이 물결무늬를 발견할 수 있다.

광선추적법의 경우 샘플링 비율을 높일수록 물결 무늬가 줄어들긴 하지만 샘플간격이 0.2인 경우에도

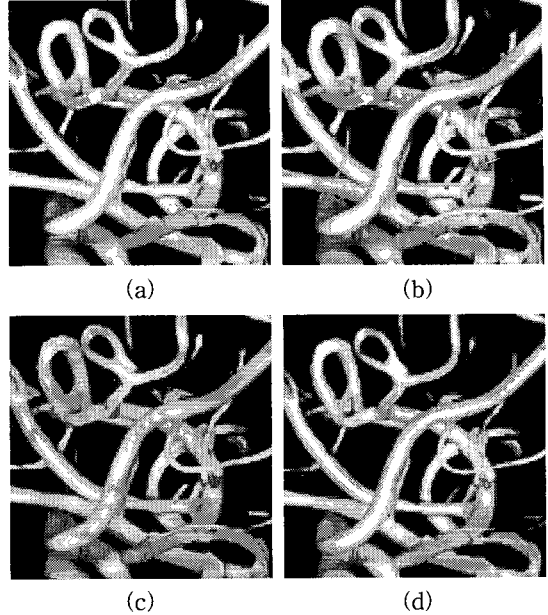


그림 8. 선-적분을 사용하지 않은 광선추적법의 샘플 간격을 변화시킨 경우와 선-적분 쉬어-왓 기법의 화질비교. (a) 광선추적(1.0), (b) 광선추적(0.5), (c) 광선추적(0.2), (d) 선-적분 (1.0)

여전히 물결무늬를 발견할 수 있다. 반면 선-적분 기법을 사용한 영상의 경우 샘플링 비율이 2배 높은 광선추적법의 영상보다 더 좋은 결과영상을 보이고 있다. 물결무늬는 샘플링 간격이 길어서 발생하는 문제이므로, 선-적분 기법을 쉬어-왓 렌더링에 적용하면, 샘플링 간격이 길어서 생기는 화질 저하가 해소됨을 알 수 있다. (광선추적법의 경우에도 선-적분 기법을 적용하게 되면 물결무늬를 제거할 수 있다.)

[그림 9]는 BrainPET 데이터를 사용한 결과이다. [그림 9a], [그림 9b]와 같이 샘플링 간격이 충분히 작지 않을 때 발생하는 물결 무늬가 [그림 9d]와 같이 선-적분 렌더링을 사용하면 제거되는 것을 확인할 수 있다.

본 연구에서 제안한, 선-적분 볼륨 렌더링의 가속화 방법의 속도 향상을 확인하기 위해 AngioS, AngioB 데이터를 이용하여 렌더링 시간을 측정하였다[표 3]. 빈-공간 도약을 하지 않는 경우 매우 느린 속도(16.70s, 33.55s)를 보이는데 Schulze 등의 연구에서도 마찬가지로 결과를 얻는다[19]. 제안 방법은 빈-공간을 효율적으로 도약하여 렌더링 속도를 크게 향상(0.48s, 0.85s)시킬 수 있으며, 광선추적법[3](0.98s, 1.49s)에 비해 2배 가량 우수한 렌더링 속도를



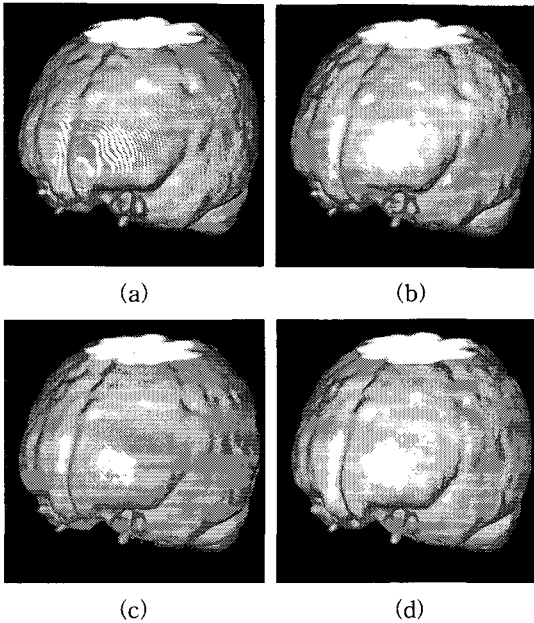


그림 9. 광선-추적법 기반의 선-적분 렌더링 화질비교. (a) 광선추적(1.0), (b) 광선추적(0.4), (c) 광선추적(0.2), (d) 선-적분(1.0)

보인다. 한편, 빈-공간 도약 시 여러 주사선을 모두 검사하는 방법 [그림 6a]는 광선추적법에 비해 빠르긴 하지만 제안 방법 [그림 6b]에 비해서는 눈에 띄게 느리다 (0.72s, 1.72s).

표 3. 고품질 볼륨 렌더링의 속도 비교. 쉬어-왓 렌더링에서 빈-공간 도약을 하지 않는 경우(a) 매우 느린 속도를 나타낸다(19). 빈-공간 도약을 하는 광선추적법(b) [3]보다 오랜 시간이 소요된다. 빈-공간 도약을 하는 경우 모든 주사선을 개별 검사하는 방법(c)에 비해 제안방법인 겹친 최소-최대 지도(d)를 이용하면 1.5 배 이상의 성능 향상을 얻는다.

렌더링 방법	AngioS (초)	AngioB (초)
(a) 빈-공간 도약 없는 쉬어-왓	16.70	33.55
(b) 빈-공간 도약 광선추적법	0.98	1.49
(c) 여러 주사선 검색법 [그림 6a]	0.72	1.72
(d) 빈-공간 도약 쉬어-왓 [그림 6b]	0.48	0.85

## 6. 결 론

본 연구는 쉬어-왓 렌더링에 선-적분을 도입하고

중간영상 좌표계에서 슈퍼샘플링하여 광선추적법 수준의 고품질 렌더링이 가능함을 보였다. 이는 쉬어-왓 렌더링의 이선형 보간으로 인한 샘플길이 증가와 와핑의 확대변환으로 인한 화질 저하를 제안한 방법으로 해결할 수 있기 때문이다. 구체적 방법으로 중간영상 좌표계에서 슈퍼샘플링을 위해 중간좌표계 기반의 렌더링 알고리즘을 제안하였다. 이 알고리즘은 중간영상과 볼륨간에 효율적으로 좌표 변환을 하여 적은 비용으로 슈퍼샘플링을 가능하게 하며, 슈퍼샘플링 비율도 정수로 제한되지 않고 임의로 설정할 수 있다. 또한, 선-적분 기반 렌더링에서 쉬어-왓 렌더링의 빠른 속도를 유지하기 위해 빈-공간 탐색을 위한 새로운 자료구조인 겹친 최소-최대 지도를 제안하였다. 그 결과 빈-공간 탐색으로 화질의 변화 없이 큰 속도 향상을 얻을 수 있었다. 제안 방법을 통해 쉬어-왓 렌더링의 빠른 속도를 유지하면서 광선추적법 수준의 고품질 렌더링이 가능하다.

## 참 고 문 헌

[1] A. Kaufman, D. Cohen, and R. Yagel, "Volume Graphics," *IEEE Computer*, Vol. 26, No. 7, pp. 51-64, 1993.

[2] R. A. Drebin, L. Carpenter, and P. Hanrahan, "Volume Rendering," *Computer Graphics (SIGGRAPH 88)*, Vol. 22, No. 4, pp. 65-74, 1988.

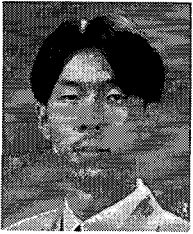
[3] M. Levoy, "Efficient Ray Tracing of Volume Data," *ACM Transactions on Graphics*, Vol. 9, No. 3, pp. 245-261, 1990.

[4] R. Yagel and A. Kaufman, "Template-Based Volume Rendering," *Computer Graphics Forum (Eurographics '92)*, Vol. 11, No. 3, pp. 153-167, 1992.

[5] B. Cabra, N. Cam, and J. Foran, "Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware," *1994 Symposium on Volume Visualization*, pp. 91-98, 1994.

[6] P. Lacroute and M. Levoy, "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation," *SIGGRAPH 94*, pp. 451-458, 1994.

- [7] P. Lacroute, "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation," *Ph.D. Dissertation*, Stanford University, 1995.
- [8] F. Dachille, K. Kreeger, B. Chen, I. Bitter, and A. Kaufman, "High-Quality Volume Rendering Using Texture Mapping Hardware," *ACM Siggraph/Eurographics workshop on Graphics Hardware*, pp. 69-76, 1998.
- [9] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler, "The volumepro real-time ray-casting system," *SIGGRAPH 99*, pp. 251-260, 1999.
- [10] K. Engel, M. Kraus, and T. Ertl, "High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading," *Eurographics / SIGGRAPH Workshop on Graphics Hardware '01*, pp. 9-16, 2001.
- [11] 구윤모, 이철희, 신영길, "복수 원형을 이용한 볼륨 렌더링에서의 효율적인 수퍼샘플링," *정보과학회연구지(A)*, 제26권, 제1호, 1999.
- [12] R. Lee and I. Ihm, "On Enhancing the Speed of Splatting Using Both Object- and Image-Space Coherence," *Graphical Models*, Vol. 62, No. 4, pp. 263-282, 2000.
- [13] K.-L. Ma, "Image graphs—a novel approach to visual data exploration," *IEEE Visualization '99*, pp. 81-88, 1999.
- [14] P. Lacroute, The VolPack Volume Rendering Library, <http://www-graphics.stanford.edu/software/volpack/>, 2006.
- [15] J. P. Schulze, R. Niemeier, and U. Lang, "The Perspective Shear-Warp Algorithm In A Virtual Environment," *IEEE Visualization 2001*, pp. 207-213, Published by IEEE, 2001.
- [16] T. Kim, B. Shin, and Y. G. Shin, "Anisotropic Volume Rendering using Intensity Interpolation," *Lecture Notes in Computer Science*, Vol. 2208, pp. 1201-1203, 2001.
- [17] J. Sweeney and K. Mueller. "Shear-warp deluxe: The shear-warp algorithm revisited," *2002 Symposium on Data Visualization*, pp. 95-104. Published by Eurographics Association, 2002.
- [18] G. Knittel, "The ULTRAVIS system," *Volume Visualization and Graphics symposium 2000*, pp. 71-79, 2000.
- [19] J. P. Schulze, M. Kraus, U. Lang, and T. Ertl, "Integrating Pre-Integration Into The Shear-Warp Algorithm," *Third International Workshop on Volume Graphics*, Tokyo, 2003.
- [20] S. Roettger and T. Ertl, "A Two-Step Approach for Interactive Pre-Integrated Volume Rendering of Unstructured Grids," *Volume Visualization '02*, pp. 23-28, 2002.
- [21] M. Meissner, S. Guthe, and W. Strasser, "Interactive Lighting Models and Pre-Integration for Volume Rendering on PC Graphics Accelerators," *Graphics Interface '02*, pages 209-218, 2002.
- [22] T. Kim and Y. G. Shin, "Fast Volume Rendering with Interactive Classification," *Computers & Graphics*, Vol. 25, No. 5, pp. 819-831, 2001.
- [23] J. J. Choi, B. Shin, Y. G. Shin, and K. Cleary, "Efficient Volumetric Ray Casting for Isosurface Rendering," *Computers & Graphics*, Vol. 24, pp. 661-670, 2000.
- [24] H. Kye, B. Shin, Y. Shin, and H. Hong, "Shear-Rotation-Warp Volume Rendering," *Computer Animation and Virtual Worlds*, Vol. 6, pp. 547-557, 2005.



계 희 원

- 1999년 2월 서울대학교 전산과 학전공(학사)
- 2001년 2월 서울대학교 전기컴퓨터공학부(석사)
- 2005년 8월 서울대학교 전기컴퓨터공학부(박사)
- 2003년 3월~2005년 12월 (주)

인피니트테크놀로지 연구원

2006년 1~현재 서울대학교 컴퓨터연구소 연구원  
 관심분야 : 컴퓨터 그래픽스, 볼륨 가시화, 하드웨어 가속화 렌더링



김 태 영

- 1991년 2월 이화여자대학교 전자계산학과(이학사)
- 1993년 2월 이화여자대학교 전자계산학과(이학석사)
- 2001년 8월 서울대학교 전기컴퓨터공학부(공학박사)
- 1993년 3월~2002년 2월 한국통신

신 멀티미디어 연구소 선임연구원

2002년 3월~현재 서경대학교 컴퓨터공학과 조교수  
 관심분야 : 볼륨그래픽스, 실시간 렌더링, 모바일3D, 증강현실