# 메모리 참조 공간 연관성을 이용한 효율적인 쉬어-왑 분해 볼륨렌더링

## (Efficient Shear-warp Volume Rendering using Spacial Locality of Memory Access)

계 희 원 † 신 영 길 ††
(Heewon Kye) (YeongGil Shin)

**요 약** 쉬어-왑 분해 볼륨렌더링은 좋은 화질과 빠른 속도를 보이지만 대화형 분류 환경에서 메모리 참조 효율이 떨어지는 문제점이 있다. 본 논문에서는 대화형 분류 환경에서 효율적인 렌더링을 수행하기 위해 객체와 이미지에 대한 메모리 참조 공간 연관성을 갖는 알고리즘을 제안한다. 이를 위하여 쉬어-왑 분해에 회전을 추가한 확장 모델을 제안하여 객체와 이미지 모두에서 스캔라인 단위로 렌더링을 가능하게 한다. 또한 제안 모델이 가지고 있는 전후향 합성 혼란, 홀 발생, 계산 증가의 문제에 대한 원인을 제시하고 해결 방법을 보인다. 본 제안 모델은 렌더링시 효율적 메모리 참조로 우수한 성능을 나타낸다.

**키워드** : 볼륨렌더링, 과학적 가시화, 메모리 참조 연관성

***Abstract*** Shear-Warp volume rendering has many advantages such as good image quality and fast rendering speed. However in the interactive classification environment it has low efficiency of memory access since preprocessed classification is unavailable. In this paper we present an algorithm using the spacial locality of memory access in the interactive classification environment. We propose an extension model appending a rotation matrix to the factorization of viewing transformation, it thus performs a scanline-based rendering in the object and image space. We also show causes and solutions of three problems of the proposed algorithm such as inaccurate front-to-back composition, existence of hole, increasing computational cost. This model is efficient due to the spacial locality of memory access.

**Key words** : Volume Rendering, Scientific Visualization, Spacial Locality

## 1. 서 론

볼륨렌더링은 3차원 볼륨데이타를 가시화 하는 방법으로 의료영상과 과학적 가시화 등에 일반적으로 사용된다[1,2]. 볼륨렌더링은 뛰어난 가시성과 높은 정밀도라는 장점을 가지는 반면 방대한 데이타 처리량과 오랜 수행시간은 단점이 되기도 한다. 이를 해결하기 위하여 볼륨렌더링 수행 각 단계에 걸쳐 데이타 처리량을 줄이고 수행 시간을 단축하는 방법이 연구되고 있다.

최소-최대 테이블(min-max table)[3]이나 거리-맵(distance map)[4]을 전처리 과정에서 구성하여 빈 공간 정보를 이용하는 방법이 연구되었고, 유효한 샘플 값만을 별도의 연속된 공간에 저장하거나[5], 메모리 재배치로[6] 메모리 참조를 빠르게 하는 방법도 제안되었다. 분류함수(classification function)또는 불투명도 전이 함수(opacity transfer function)를 전처리 과정에서 샘플에 미리 적용하거나[7] 렌더링시 참조 테이블(lookup table)을 사용하여[8] 계산을 줄이는 연구도 있었으며, 객체 좌표계(object coordinate)와 관찰자 좌표계(viewing coordinate)간의 매핑에서, 인접한 샘플의 위치 연관성을 이용해 계산을 줄이는 방식도 제안되었다[5,9]. 최근 범용 그래픽스 하드웨어의 발전으로 3D 텍스쳐 기반 렌더링도 많이 연구되고 있다. 하드웨어 렌더링은 속도가 빠르며 화질이 우수하다는 장점이 있지만 최근의 대용량 볼륨에 비해 텍스쳐 메모리가 상대적으로 작고 기능이 하드웨어에 종속적인 단점은 해결해야 할 과제이다.

· 이 논문은 2002년도 두뇌한국21사업에 의하여 지원되었음
† 정 회 원 : (주)인피니트 테크놀로지
kuei@cglab.snu.ac.kr
†† 종신회원 : 서울대학교 전기컴퓨터공학부 교수
yshin@cglab.snu.ac.kr
논문접수 : 2002년 7월 16일
심사완료 : 2003년 11월 27일

쉬어-왑 분해[5]는 대표적인 최적화된 렌더링 방법으로 전처리 과정에서 분류를 수행하여 데이타 처리량을 줄이고 볼륨 메모리 참조를 빠르게 할 뿐 아니라 중간 좌표계를 도입하여 좌표계간 매핑을 간략화하고 이미지 메모리 참조를 효율적으로 한다. 또한 조기 광선 종료(early ray termination)나 RLE(Run-Length Encoding)로 공간 탐색을 최적화한다. 그러나 최근 요구되는 대화적 분류(classification)에서는 전처리로 분류를 할 수 없으므로[10] 거리-맵을 사용하거나 유효한 데이타만을 별도의 공간에 저장하는 방식[11]이 더 이상 가능하지 않으며, 옥트리(octree)를 사용해 볼륨데이타를 참조해야 한다. 이 경우 볼륨 메모리 참조가 비순차적으로 바뀌게 되어 성능 저하의 원인이 되며, 최근의 하드웨어는 CPU의 속도에 비해 메모리 참조 속도가 현저히 느리므로 성능저하의 폭은 더욱 크다.

본 논문에서는 쉬어-왑 분해에 회전 변환을 추가해 복셀 스캔라인 방향과 픽셀 스캔라인 방향을 일치시켜 볼륨과 이미지 메모리를 동시에 순차적으로 참조할 수 있는 렌더링 모델을 제안한다. 제안 모델은 메모리 참조 공간 연관성을 높이고 캐쉬 메모리의 적중률(cache hit ratio)을 증가시켜 효율적인 렌더링이 가능하다. 그러나 제안 모델은 전후향 합성(front-to-back projection) 혼란, 홀(hole) 발생, 좌표계간 매핑의 계산 증가라는 세 가지 문제점을 가지고 있는데 렌더링 순서 변화, 축소 변환 도입, 템플릿 이용으로 해결할 수 있다. 본 제안 모델은 수학적으로 견고하며 화질 손실을 최소화하면서 우수한 속도를 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 쉬어-왑 분해와 본 연구에서 제안한 모델을 수학적으로 비교하고, 제안 모델이 가지고 있는 문제점에 대한 원인을 분석한다. 3장에서는 2장에서 제시한 문제점에 대한 해결 방법을 제시한다. 4장에서는 본 논문에 적용 가능한 최적화 기법을 소개하며 5장에서는 실험 결과를 보이고 6장에서 결론을 내린다.

## 2. 제안 확장 모델

### 2.1 기존 모델

쉬어-왑 분해는 행렬의 분해를 통해 계산과 메모리 참조 효율을 향상시키는 방법으로 식 (1)과 같이 설명된다[5].

$$\begin{aligned} M_{view} &= P \cdot S \cdot (P \cdot S)^{-1} \cdot M_{view} \\ &= P \cdot S \cdot M_{warp} \qquad (1) \\ (M_{warp} &= (P \cdot S)^{-1} \cdot M_{view}) \end{aligned}$$

이때, $M_{view}$는 객체 공간에서 영상 공간으로의 변환 행렬이며 P는 전치 행렬, S는 쉬어 행렬이다.

쉬어-왑의 장점은 다음의 네 가지로 요약된다[13]. 첫째, 투영과정에서 샘플의 좌표와 중간 이미지 픽셀의 대응이 1:1 대응으로 간단하다. 즉, 좌표 변환의 연관성을 높여 계산을 감소시키는 효과가 있다. 둘째, 볼륨데이타의 스캔라인 방향과 중간 이미지의 스캔 라인 방향이 일치해 메모리 참조 효율이 높다. 스캔라인 단위의 작업을 하면 렌더링 공간 연관성이 높아져 캐쉬 적중률이 좋아진다. 셋째, RLE를 사용하여 렌더링에 불필요한 복셀을 제거하여 데이타를 줄인다. 단, 주시각축 각각에 대하여 구성해야 하므로 세 벌의 서로 다른 데이타가 필요하다. 넷째, 중간 이미지에서 빠른 광선 종료(early ray termination)가 가능하도록 중간 이미지의 크기와 같은 해상도의 마스크 이미지(opaque run)를 만들어 불필요한 복셀을 건너뛴다.

그러나 최근의 볼륨렌더링은 다음과 같은 세가지 요구 조건을 갖는다. 첫째, 분류작업이 매우 자주 일어난다. 둘째, 볼륨데이타 크기가 크다. 셋째, 대화적 수준의 렌더링 속도를 가져야 한다. 대화적 분류 환경에서 기존의 쉬어-왑은 구성시간이 오래 걸리는 RLE 사용을 할 수 없으므로 복셀의 밀도 값을 얻기 위해 볼륨데이타를 좌표를 통해 참조하고 빈 공간의 탐색은 옥트리와 같은 추가 자료구조를 이용한다. 이때, 메모리 공간 연관성 저하와 옥트리의 탐색 비용으로 성능이 크게 감소한다[5]. 볼륨 데이타의 크기가 크므로 성능의 감소폭은 더욱 커지며 따라서 대화적 수준의 렌더링이 어렵게 된다.

메모리 공간 연관성 문제를 확인하기 위해 그림 1을 제시하고 있다. 볼륨데이타는 메모리에 저장될 때 1차원 배열의 형태로 저장되므로 y 방향과 z 방향에서 인접한

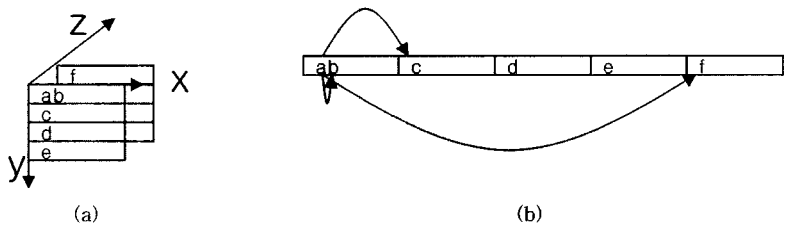



그림 1 공간과 메모리의 위치 차이

복셀은 메모리에서는 인접하지 않는다. 주시각축이 z 축인 경우에는 a와 b를 연속적으로 읽게 되므로 메모리 참조의 공간 연관성이 크지만, 다른 주시각축에 대해서는 a와 c, a와 f를 연속적으로 읽게되어 메모리 참조의 공간 연관성이 떨어지게 된다. 쉬어-왑은 주시각축에 대해 렌더링 순서가 다르며 그림 2의 (b), (c), (d)와 같이 주시각축이 각각 z, y, x 축인 경우, 렌더링은 중간 좌표계의 I 축과 같은 방향인 x, z, y 방향으로 진행한다. x 방향으로 진행하는 (b)경우를 제외하면, (c)나 (d)와 같이 중간 좌표계의 i 축에 대응되는 객체 좌표계의 축이 z 축이거나 y 축이 되는 경우 성능이 저하된다.

이와 같은 메모리 참조 문제를 해결하기 위해 메모리 시스템을 서브 블록(sub block), 슬라이스(slice), 8-방향 삽입(eight-way interleaved, skewed)의 방법을 사용하여 재구성하기도 한다[6]. 그러나 스캔라인을 기반으로 한 쉬어-왑 분해의 경우 CPU의 캐쉬 블록 크기가 작고 메모리 주소 계산에 필요한 연산이 많으며, 메모리 참조 효율을 높이기 위해 볼륨데이타를 세 벌 갖는 경우 메모리 소모량이 지나치게 커지게 된다. 또한 광선 템플릿(ray template)을 이용해 주시각축이 y, z 축인 경우 볼륨의 스캔라인 방향과 중간 이미지의 스캔라인 방향을 일치시키는 연구[13]도 있지만 주시각축이 x 축인 경우 불가능하다는 단점이 있다.

### 2.2 회전 변환을 추가한 제안 확장 모델

본 논문에서는 그림 2와 같이 x 축을 i 축과 평행하게 하도록 회전 변환을 추가한다. 쉬어 변환을 한 후에 x 축과 i 축이 평행하도록 k 축을 중심으로 적당한 각도의 회전 R(θ)을 한다. 이 결과는 식 (2)로 요약된다.

$$M_{view} = P \cdot S \cdot R \cdot (P \cdot S \cdot R)^{-1} \cdot M_{view} = P \cdot S \cdot R \cdot M'_{warp} \quad (M'_{warp} = (P \cdot S \cdot R)^{-1} \cdot M_{view}) \tag{2}$$

그림 2에서 중간 이미지 좌표계가 (a)와 같이 존재하는 경우 쉬어 S에 따른 변환은 주시각축에 따라 (b), (c), (d)와 같은 모습을 보인다. 여기에 적당한 회전 변환 R을 통해 (b), (c), (d)는 각각 (e), (f), (g)와 같이 x 축과 i 축이 평행한 형태가 된다. 그리고 렌더링은 2.1절과 같이 i 축 방향으로 진행한다. (e), (f), (g)의 경우 x 축과 i 축이 평행하므로 x, y, z의 어떠한 주시각축에서든 i 축을 진행할 때 볼륨은 항상 x 축을 진행한다. 따라서 인접한 복셀을 렌더링 할 때 메모리 위치도 인접하게 되어 2.1절에서 제시된 메모리 참조 연관성 문제가 해결된다.

회전각도는 그림 2에서 보는 것과 같이, 쉬어 변환 후의 x 축과 i 축 사이의 각이므로 주시각축이 z 축인 경우 0 또는 2π, y 축인 경우 π/2이다. 주시각축이 x 축인 경우 시각 벡터 Vview와 S의 원소 sx, sy을 이용하여, 식 (3)을 만족하는 회전각도 θ를 얻을 수 있다.

$$\cos\theta = s_x \cdot \frac{V_{view} \cdot x}{|V_{view} \cdot x|} \qquad \sin\theta = -s_y \cdot \frac{V_{view} \cdot x}{|V_{view} \cdot x|} \tag{3}$$

회전 변환을 사용한 확장 모델에서 발생하는 문제는 다음의 세 가지가 있다.

1. 전후향 합성 혼란: 쉬어-왑은 전후향 합성이므로 관찰자로부터 가까운 복셀부터 먼 순서대로 렌더링을 수행해야 먼 곳에 있는 물체가 가까운 곳의 물체를 가리는 문제가 발생하지 않는다. 따라서 그림 3에서 보는 것과 같이 주시각축 방향을 가장 밖에서 반복해야한다. 그러나 본 논문에서 제안한 방법은 x 축이 가장 내부에 위치해야 하므로 주시각축이 x 축일 때 반복 순서를 결정할 수 없게 되는 문제가 발생한다.
2. 홀 발생: 투영 과정이 객체기반 방식이므로 전향 매핑에서 홀이 발생할 수 있다. 쉬어-왑은 주시각축에

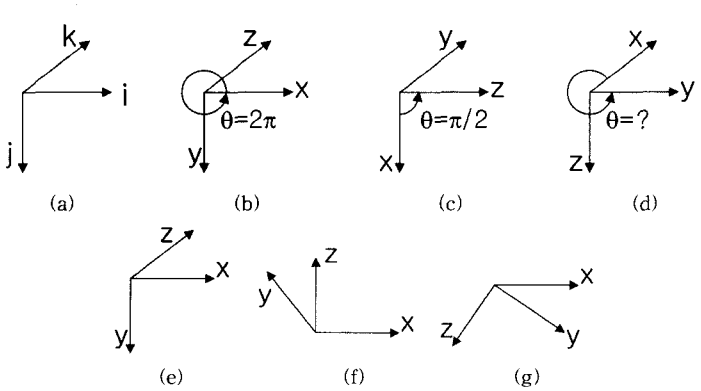



그림 2 중간 이미지가 회전 변환을 통한 결과

```
for z:=Zmin to Zmax do
  for y:=Ymin to Ymax do
    for x:=Xmin to Xmax do
      compose_voxel_to_image(x,y,z);
```

그림 3 주시각축이 z 축인 경우 반복 순서(z → y → x)

수직인 평면의 복셀과 중간 이미지 영상의 픽셀이 1:1로 대응하기 때문에 홀이 발생하지 않지만, 본 논문에서는 회전을 통해 1:1 대응이 성립하지 않게 되므로 회전 각도가 $n\pi/2$로 표현되지 않는 주시각축 x 축에서 홀이 발생한다.

3. 계산 증가: 투영 좌표 변환이 회전 변환으로 복잡해지므로 계산이 증가하여 성능 저하의 원인이 된다.

## 3. 확장 모델의 문제 해결

### 3.1 전후향 합성 보장

투영과정은 빠른 광선 종료법을 적용하기 위해 전후향 슬라이스 순서(near-to-far slice order)로 진행한다. 쉬어-왑 분해의 경우 슬라이스 단위로 렌더링을 수행하고, 각 슬라이스에서의 샘플은 중간 이미지의 픽셀에 1:1로 대응되므로 슬라이스의 순서만 고려한다. 그러나, 슬라이스 단위로 렌더링을 하지 않는 경우 전후향 합성을 보장하기 위해 처리 순서가 달라져야 하며[15], 본 논문에서는 x 축 스캔라인을 단위로 렌더링을 수행하므로 슬라이스, 스캔라인, 복셀의 처리 순서를 모두 고려하였다. 각 주시각축에 따라 시각 벡터 Vview의 방향과 sx, sy의 부호를 고려하면 모두 24가지의 경우가 존재하며 이를 정리하면 표 1과 같은 결과를 얻는다.

### 3.2 최근접 화소 보간에서 전향 매핑의 홀 제거를 위한 축소 변환

쉬어-왑 분해의 경우 이중 선형 보간(bilinear interpolation)으로 샘플을 얻어 중간 이미지에 투영하는 방식이며 슬라이스에 대해 보간 가중치(sampling weight)가 일정하기 때문에 효율적인 방법이다. 반면 본 논문에서는 회전 변환을 하므로 보간 가중치가 일정치 않다. 각 샘플에 대해 보간 가중치를 계산하면 렌더링 시간이 오래 걸리므로, 각 복셀을 전향 매핑을 이용해 중간 이미지에 매핑 하는 것이 효율적이다. 그러나 각 복셀의 쉬어-회전 변환한 결과에 가장 근접한 정수 좌표로 중간 이미지에 매핑하면, 전향 매핑의 최근접 화소 보간(nearest sampling)이므로 홀이 발생하게 된다. 홀의 원인은 회전 변환이므로 주시각축이 y, z와 같이 $n\pi/2$형태의 회전에서는 홀이 발생하지 않으며, 주시각축이 x 축인 경우만 발생하게 된다. 이와 같은 홀 발생을 제거하기 위해 본 논문에서는 중간 이미지 크기를 축소시켜 홀의 발생을 제거하는 축소 변환을 추가하는 방법을 제안한다.

임의의 아핀(affine) 변환에서는 전향 매핑에서 홀을 제거하기 어렵지만, 회전 변환은 중간 이미지의 투영 간격을 좁혀 홀 발생을 막을 수 있다. 예를 들어 회전 각도가 $\pi/4$인 경우 홀이 가장 많이 발생하며, 결과 이미지를 $1/\sqrt{2}$의 비율로 축소시키면 홀이 발생하지 않는다. 이를 회전 변환에서 샘플간의 거리 변화를 이용하여 일반화시키면 크기 변환 행렬 $SC(\sqrt{s_x^2+s_y^2}\,/\,|s_x+s_y|)$을 얻는다. 크기 변환을 포함한 결과를 정리하면 식 (2)와 비슷하게 식 (4)로 표현할 수 있으며 이로써 홀이 제거된다.

$$\begin{aligned} &M_{view} \\ &= P\cdot S\cdot R\cdot SC\cdot (P\cdot S\cdot R\cdot SC)^{-1}\cdot M_{view} \\ &= P\cdot S\cdot R\cdot SC\cdot M''_{warp} \\ &(M''_{warp} = (P\cdot S\cdot R\cdot SC)^{-1}\cdot M_{view}) \end{aligned} \tag{4}$$

### 3.3 템플릿을 이용한 투영 연산 최소화

전향 매핑의 회전 변환, 축소 변환은 계산을 증가시키는 단점이 있다. 본 논문에서는 템플릿 기반의 참조 테이블을 사용해 문제를 해결한다. 투영 연산은 식 (4)에서 보듯이 전치, 쉬어, 회전, 축소 변환의 곱으로 수행된다. 평행 투영을 가정하는 경우 아핀 맵(affine map)이 되므로 비율 보존(ratio preserving)이 되고 객체 좌표의 각 좌표축의 벡터가 중간 이미지 좌표로 매핑 되었을 때 선형성을 보장한다[14]. 본 논문에서는 x, y, z의 각 좌표축이 중간 이미지에서의 변위에 해당하는 템플릿 참조 테이블 $T_x$, $T_y$, $T_z$를 $M_{affine}$의 열벡터를 이용해서 식 (5)와 같이 생성한다.

표 1 주시각축에 대한 렌더링 순서와 증감. sgn(x)= + (x>=0), - (x<0)

| 주시각축 | z | y | x |
|---|---|---|---|
| 렌더링 순서 | z → y → x | y → z → x | z → y → x |
| 회전각도 | 0 (2π) | π/2 | θ |
| z증감 | sgn($V_{view}.z$) | + | sgn($-s_y \cdot V_{view}.x$) |
| y증감 | + | sgn($V_{view}.y$) | sgn($-s_x \cdot V_{view}.x$) |
| x증감 | + | + | sgn($V_{view}.x$) |

$$\begin{bmatrix} i \\ j \\ k \\ 1 \end{bmatrix} = M_{affine} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$
$$= M_{affine} \begin{bmatrix} x \\ 0 \\ 0 \\ 1 \end{bmatrix} + M_{affine} \begin{bmatrix} 0 \\ y \\ 0 \\ 1 \end{bmatrix} + M_{affine} \begin{bmatrix} 0 \\ 0 \\ z \\ 1 \end{bmatrix} \quad (5)$$
$$= T_x[x] + T_y[y] + T_z[z]$$
(단, $M_{affine} = P \cdot S \cdot R \cdot SC$)

이때, 복셀의 투영은 x, y, z로부터 i, j를 얻는 과정이며 식 (6)으로 간단히 할 수 있다. 연속적인 투영 과정에서 복셀은 x 좌표만이 변경되므로 새로운 i, j를 얻기 위해 식 (6)의 $T_x[x].x$만 갱신한다. 따라서 $T_x$에 대한 참조연산 한번과 덧셈 한번으로 새로운 투영 좌표가 결정되어 연산을 최소화한다.

$$i = T_x[x].x + T_y[y].x + T_z[z].x$$
$$j = T_y[y].y + T_z[z].y \quad (6)$$
$$(\because Tx[x].y = 0)$$

## 4. 최적화

분류 작업이 자주 일어나는 경우 쉬어-왑은 일반적으로 옥트리를 구성하고 렌더링을 수행하지만 스캔라인 기반의 렌더링 특징을 고려하여 본 논문에서는 스캔라인 최소-최대 테이블을 생성한다[15].

### 4.1 최소-최대 테이블

빠른 렌더링을 위한 전처리는 대부분의 시스템에서 필수적이다. 전처리에서는 옥트리, 최소-최대 테이블, RLE, 누적 영역 테이블(summed area table) 등의 기법을 사용할 수 있으며[5] 이들의 선택과 적용은 렌더링 방식에 좌우된다. 빠른 분류를 위한 스캔라인 단위의 렌더링을 하는 본 논문에서는 스캔라인 최소-최대 테이블을 적용하였다. 옥트리 최소-최대 테이블 기법이 $2^3$, $4^3$, $8^3$ 등의 정방형 블록의 최소-최대 값을 저장하는 방식인데 비해 스캔라인 최소-최대 테이블은 16×1×1과 같이 긴 막대 모양 블록의 최소-최대 값을 저장한다.

스캔라인 최소-최대 테이블은 각 렌더링 주시각축에 대해 16×1×1, 1×16×1, 1×1×16의 3벌을 유지해야 하지만[15], 본 논문에서는 x 방향의 렌더링을 하므로 1벌의 스캔라인 최소-최대 테이블만으로 충분하다. 따라서 기존 방식에 비해 메모리 소모량과 구성 시간이 1/3으로 줄어든다. 옥트리와 마찬가지로 스캔라인 최소-최대 테이블도 계층적 구조로 만들 수 있으며 본 논문에서는 16×1×1, scanline width×1×1의 두 단계의 계층만을 두어서 실험하였다.

각 블록에 대해 최소-최대 값을 저장해야 하므로 추가적인 메모리 소모량은 원 볼륨의 크기와 비교하면 $\frac{2}{16} + \frac{2}{scanline\ width} \approx \frac{1}{8}$($\because scanline\ width \gg 16$) 이며 옥트리에서 리프 노드(leaf node)가 $2^3$인 경우(2/7), $4^3$인 경우(1/28)에 비해 중간정도의 추가 메모리 소모량을 보인다.

## 5. 실 험

본 실험은 1GB의 메모리를 가진 1.7GHz의 Pentium 4 개인용 컴퓨터에서 수행하였으며, 실험에 사용한 데이타는 표 2와 같다. Head는 스텐포드 대학교의 VolPack 라이브러리[16]에서 공개적으로 제공하는 것이고, Chest는 서울대학교 진단방사선과에서 제공한 컴퓨터단층촬영영상(CT: Computed Tomography)으로 대용량 데이타 환경을 실험하기 위하여 사용했다. 각 복셀은 2바이트의 밀도 값(density)을 갖도록 하고 전처리로 법선벡터를 구해 2바이트로 양자화 하였다. 따라서 각 복셀에 대해 4바이트의 저장 공간을 필요로 한다.

표 2 실험에 사용한 볼륨 데이타

| 볼륨 데이타 | 영상크기 | 슬라이스의 수 | 저장 공간(4bytes/복셀) |
|---|---|---|---|
| Head | 256×256 | 225 | 59.0MB |
| Chest | 512×512 | 200 | 209.7MB |

본 실험에서는 제안 모델의 성능을 알아보기 위해, 기존 알고리즘과 제안 알고리즘 간의 렌더링 속도의 차이를 분석한다. 그리고 속도 차이 발생의 근거로, 메모리 참조 공간 연관성으로 인한 렌더링시 캐쉬 미스 횟수를 분석한다. 또한, 기존 알고리즘과의 화질 비교 결과를 보이고 2장에서 제시한 제안 모델의 세가지 문제점을 해결하지 않는 경우의 결과를 보인다.

제안 알고리즘의 성능을 분석하기 위해 렌더링 속도를 측정한 결과를 표 3에 제시하였다. 빈 공간 탐색을 위한 방법으로 쉬어-왑과 제안 알고리즘에 대해 최대-최소 테이블을 사용하지 않거나 옥트리 또는 스캔라인 최소-최대 테이블을 적용할 수 있다. 옥트리의 리프 노드는 $4^3$의 크기를 갖도록 하였다[5].

최소-최대 테이블을 적용하지 않은 (a), (d)의 결과를 보면 제안 알고리즘을 사용한 (d)의 경우 (a)보다 5배 이상 빠른 결과를 보인다. 이는 모든 복셀에 대해 투명도 검사를 하므로 대부분의 복셀을 읽어들여야 하기 때문이다. 옥트리를 사용한 (b)와 (e)의 결과를 비교하여도 제안 알고리즘의 성능이 더 좋음을 알 수 있다. 스캔라인 최소-최대 테이블을 사용한 (c)와 (f)를 비교하면 Head에서는 2배, Chest에서는 3배 가량의 성능 향상을 얻는다. 실험 결과를 보면 옥트리를 이용한 경우보다 스

표 3 쉬어-왑과 제안 알고리즘의 수행 시간 비교(단위: ms)

| | 쉬어-왑 | | | 제안 알고리즘 | | |
|---|---|---|---|---|---|---|
| | 최소-최대 테이블 없음(a) | 옥트리 최소-최대 테이블(b) | 스캔라인 최소-최대 테이블(c) | 최소-최대 테이블 없음(d) | 옥트리 최소-최대 테이블(e) | 스캔라인 최소-최대 테이블(f) |
| Head | 1493 | 219 | 161 | 281 | 174 | 78 |
| Chest | 3904 | 931 | 637 | 625 | 588 | 217 |

캔라인 최소-최대 테이블을 이용한 경우에 성능 향상 폭이 더 큰데, 그 이유는 스캔라인 최소-최대 테이블이 메모리 참조 연관성을 더 많이 이용하는 자료구조이기 때문이다.

메모리 참조 공간 연관성을 비교하기 위해 소프트웨어 분석 도구인 Intel의 vTune 소프트웨어를 이용해 복셀의 밀도 값을 읽는데 발생하는 캐쉬 미스 횟수를 조사하였다. 실험 데이타는 Head를 사용했고, 양쪽 모두 스캔라인 최소-최대 테이블을 사용하였다. 복셀의 값을 읽는 부분은 11개 인스트럭션(instruction)으로 이루어져 있으며 이 부분을 1000회 반복 실행하여 얻은 결과를 표 4에 제시하고 있다. 즉, 실제 렌더링 과정에서 1000개의 복셀을 렌더링 할 때 복셀의 밀도 값을 얻는 과정을 시뮬레이션 하여 캐쉬 미스의 횟수를 조사한 결과이다.

표 4 복셀 밀도값 참조에 대한 캐쉬 미스 횟수 분석

| | 쉬어-왑 | 제안 알고리즘 |
|---|---|---|
| 반복 횟수 | 1000 | 1000 |
| 인스트럭션 개수 | 11000 | 11000 |
| 명령어 캐쉬 미스 회수 | 2 | 2 |
| 데이타 캐쉬 미스 회수 | 955 | 159 |
| 총 수행 시간 (Total cycles) | 86191 | 18576 |
| CPI (Clock Per Instruction) | 7.84 | 1.69 |

인스트럭션의 수와 반복 횟수는 같지만, 데이타 캐쉬 미스의 발생 횟수를 살펴보면 쉬어-왑(955회)에 비해 제안 알고리즘(159회)이 매우 적음을 알 수 있다. 1000개의 복셀의 참조 결과이므로, 쉬어-왑이 거의 매 복셀 참조마다(95.5%) 데이타 캐쉬 미스가 발생하는데 비해 제안 알고리즘에서는 적은 비율(15.9%)로 캐쉬 미스가 발생하고 있다. 이것은 제안 알고리즘에서 메모리 참조의 공간 연관성이 훨씬 높다는 것을 의미한다. 캐쉬 미스의 발생을 줄이면 대기 시간이 줄어들게 되므로 총 수행시간과 CPI를 향상시키는 효과를 얻는다. 이 결과, 속도의 차이는 4.6(=7.84/1.69)배로 표 3에서 Head의 (c)와 (f)에서 보인 성능 차이 2배에 비해 더 크다. 앞의 실험에서 Head에 비해 Chest에서 더 높은 성능 향상을 얻은 것과 마찬가지로, 렌더링의 복셀 참조 횟수가 많아질수록 더 큰 속도 향상을 기대할 수 있다는 것을 알 수 있다.

결과 영상을 비교하기 위해 Head와 Chest의 영상을 각각 그림 4와 그림 5에 제시하였다. 주시각축이 y, z 축인 경우 완전히 동일한 결과가 나오므로 비교에서 제외하였고, 영상의 비교는 주시각축이 x 축인 경우로 한정하였다. 그림 4의 Head는 (a)의 제안 알고리즘과 (b)의 쉬어-왑 결과가 거의 동일함을 보인다. 그림 5의 Chest에서도 (a)의 쉬어-왑과 (b)의 제안 알고리즘의 결과가 거의 유사한 것으로 나타난다.

(a) (b)

그림 4 Head의 렌더링 결과

(a)

(b)

그림 5 Chest의 렌더링 결과

본 논문의 제안 알고리즘에서 발생하는 문제를 3장과 같은 방법으로 해결하지 않는 경우, 이미지의 손상이 일어나거나 성능이 저하된다. 그림 6은 각각 전후향 합성을 보장하지 않은 경우(a)와 홀의 생성을 방치했을 경우(b) 생성되는 결과이다. 그리고 계산 증가를 위한 템플릿을 사용하지 않을 경우 Head에서의 렌더링 속도는 116ms로 표 3의 (f)와 비교하면 33% 가량의 속도 저하가 발생한다.

(a) (b)

그림 6 전후향 합성 혼란과 홀 발생

## 6. 결 론

본 논문에서는 대화적 분류가 요구되는 볼륨렌더링에서 볼륨과 이미지의 메모리 참조를 모두 스캔라인 방향으로 하는 새로운 모델을 제안하였다. 쉬어-왑을 기반으로 하여 회전 변환을 추가한 제안 모델은 수학적으로 견고하며 개념적으로 명료하다. 제안 모델에서 발생하는 전후향 합성, 홀 발생, 계산량 증가 문제는 렌더링 순서 변경, 축소 변환 적용, 템플릿 사용으로 해결하였다.

제안 알고리즘은 메모리 참조의 공간 연관성을 이용하여 캐쉬 적중률을 높이고 있다. 6배 정도로 높아진 캐쉬 적중률은 전체 성능을 높이는데 기여하여, 최소-최대 테이블을 사용하지 않는 실험 결과에서 기존의 알고리즘에 비해 5배 이상 빠른 속도를 보인다. 빈 공간 탐색을 위한 최소-최대 테이블을 사용한 경우에도 여전히 우수한 성능을 보이며, 특히 스캔라인 최소-최대 테이블과 같이 메모리 참조의 공간 연관성을 많이 이용하는 자료구조를 도입하면 기존 알고리즘에 비해 2배 이상의 속도 향상을 이룰 수 있다. 또한 실험 결과에 나타난 것처럼 렌더링할 복셀이 많은 데이타일수록 성능 향상의 폭이 크므로, 본 논문은 대용량 볼륨의 대화적 속도 렌더링을 필요로 하는 최근의 요구에 부합한다. 향후 과제로는 홀의 발생을 더 효율적이며 좋은 화질을 갖는 방법으로 제거하는 방법을 연구하고자 한다.

## 참 고 문 헌


[1] Arie Kaufman and Daniel Cohen and Roni Yagel, Volume Graphics, *IEEE Computer*, 26(7), pp. 51-64 (July 1993).

[2] Robert A. Drebin and Loren Carpenter and Pat Hanrahan, Volume Rendering, *Computer Graphics (Proceedings of (SIGGRAPH 88)*, 22(4), pp. 65-74 (August 1988).

[3] Marc Levoy, Efficient Ray Tracing of Volume Data, *ACM Transactions on Graphics*, 9(3), pp. 245-261 (July 1990).

[4] Sarah F. Gibson, Using Distnace Maps for Accurate Surface Representation in Sampled Volumes, *Volume Visualization Symposium*, pp. 23-30 (October 1998).

[5] Philippe Lacrout and Marc Levoy, Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation, *Proceedings of SIGGRAPH 94, Computer Graphics Proceedings*, Annual Conference Series, pp. 451-458 (July 1994).

[6] Harvey Ray and Hanspeter Pfister and Deborah Silver and Tood A. Cook, Ray Casting Architectures for Volume Visualization, *IEEE Transactions on Visualization and Computer Graphics*, 5(3), pp. 210-223 (July - September 1999).

[7] Brian Cabra and Nancy Cam and Jim Foran, Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware, *1994 Symposium on Volume Visualization*, pp. 91-98 (October 1994).

[8] Marc Levoy, Volume Rendering Display of Surfaces from Volume Data, *IEEE Computer Graphics & Application*, 8(3), pp. 29-37 (May 1998).

[9] Roni Yagel and Arie Kaufman, Template-Based Volume Rendering, *Computer Graphics Forum (Eurographics '92)*, 11(3), pp. 153-167 (September 1992).

[10] Kim, T. Y. and Shin, Y. H., Fast Volume Rendering with Interactive Classification, *The Journal of Computers and Graphics*, Vol. 25, No. 5, 2001.

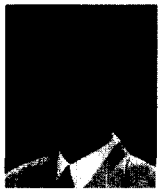
[11] Jayaram K. Udupa and Dewey Odhner, Shell Rendering, *IEEE Computer Graphics and Applications* (November 1993).

[12] Intel Corporation, *Intel Architecture Optimization (Reference Manual)*, http://www.intel.com/developer, 1999.

[13] Cheol-Hi Lee and Yun-Mo Koo and Yeong Gil Shin, Template-based rendering of run-length encoded volumes, *Proceedings of the 5th Pacific Conference on Computer Graphics and Applications (Pacific Graphics '97)*. 1997

[14] G. Farin, *Curves and Surfaces for CAGD*, 4th ed.,

Academic Press, Boston, 1997.
[15] 김태영, 비균등 볼륨 데이타를 위한 대화형 분류 및 렌더링, 박사학위논문, 서울대학교, 2001.
[16] Lacroute, P., VolPack Software Distribution, URL http://graphics.stanford.edu/software/volpack, 1995.
[17] R. Lee, and I. Ihm, On Enhancing the Speed of Splatting Using Both Object- and Image-Space Coherence, *Graphical Models(formerly GMIP)*, Vol. 62, No. 4, pp. 263-282, July 2000.



계 희 원

1995년 3월~1999년 2월 서울대학교 전산과학전공(학사). 1999년 3월~2001년 2월 서울대학교 전기컴퓨터공학부(석사) 2001년 3월~현재 서울대학교 전기컴퓨터공학부 박사과정. 현재 (주)인피니트테크놀로지 연구원. 관심분야는 컴퓨터 그래픽스, 볼륨 가시화, 하드웨어 가속화 렌더링

신 영 길

1978년~1981년 서울대학교 계산통계학과(학사). 1982년~1984년 서울대학교 계산통계학과(석사). 1984년~1989년 University of Southern California(Ph.D) 1990년 2월~1992년 2월 경북대학교 전임강사. 1992년 3월~현재 서울대학교 전기컴퓨터공학부 교수. 관심분야는 컴퓨터 그래픽스, 볼륨 가시화, 영상처리, 실시간 렌더링