

2차원 리샘플링에 기반한 광선추적법의 속도 향상 기법

(Speed Enhancement Technique for Ray Casting using 2D Resampling)

이 래 경[†] 임 인 성^{**}

(Rae-Kyoung Lee) (Insung Ihm)

요 약 볼륨 데이터에 대한 팔진트리와 같은 계층 자료구조를 사용하는 광선 추적법은 모든 광선이 계층구조를 순회하는 것으로 인한 중복된 계산을 포함하고 있으며, 좋은 화질의 영상을 얻기 위한 3차원 보간으로 인하여 많은 계산 비용을 요구한다. 본 논문은 볼륨 데이터의 계층구조에 대한 중복된 방문을 피하고, 오직 한 번만 계층구조를 방문하면서 효과적으로 광선의 리샘플링 지점을 결정하여 색상과 투명도를 구하는 볼륨 렌더링 알고리즘을 제안한다. 이 방법은 물체 순서로 광선 추적법을 수행하면서, 각 복셀 주위에서의 리샘플링 지점을 점진적으로 찾아가면서 각 슬라이스 상에서의 2차원 보간에 기반을 둔 리샘플링을 수행한다. 또한 물체 순서 렌더링에서는 조기 광선 종결과 같은 최적화 기법을 구현하기 힘든데, 영상공간에서의 동적 자료구조를 이용하여 이를 효과적으로 해결하였다. 본 논문이 제안한 방법은 구현하기 쉽고 속도 향상을 위하여 추가적으로 요구되는 메모리가 매우 적기 때문에 광선 추적법과 쉬어 와핑 방법 사이의 성능 차이를 메워주는 효과적인 방법으로 사용될 수 있을 것이다.

Abstract The standard volume ray-tracing, optimized with octree, needs to repeatedly traverse hierarchical structures for each ray that often leads to redundant computations. It also employs the expensive 3D interpolation for producing high quality images. In this paper, we present a new ray-casting method that efficiently computes shaded colors and opacities at resampling points by traversing octree only once. This method traverses volume data in object-order, finds resampling points on slices incrementally, and performs resampling based on 2D interpolation. While the early ray-termination, which is one of the most effective optimization techniques, is not easily combined with object-order methods, we solved this problem using a dynamic data structure in image space. Considering that our new method is easy to implement, and need little additional memory, it will be used as very effective volume method that fills the performance gap between ray-casting and shear-warping.

1. 서 론

볼륨 렌더링(volume rendering)은 볼륨 데이터를 이루는 복셀(voxel) 상에서 정의된 스칼라와 벡터 데이터로부터 3차원 그래픽스 기법을 적용하여 영상을 생성하

는 기술을 말한다. 볼륨 렌더링은 일반적으로 그 볼륨 데이터의 거대한 양으로 인해 렌더링 시에 많은 시간을 요구하게 된다[1]. 볼륨 렌더링 알고리즘은 영상 순서(image-order) 렌더링 방법과 물체 순서(object-order) 렌더링 방법, 그리고 이 두 가지의 혼합 형태의 방법 등으로 나눌 수가 있다. 영상 순서 렌더링은 영상 평면(image plane)의 각 픽셀을 향하여 광선을 쏘아서 일정한 간격으로 샘플링하여 영상을 생성해내는 알고리즘이다. 대표적인 영상 순서 렌더링 방법에는 광선 추적법(ray casting)[2,3]이 있다. 물체 순서 렌더링 방법은 볼륨 데이터를 순차적으로 순회하면서 복셀을 영상평면으로 투영하여 영상을 생성하는 알고리즘으로 스피래팅

· 본 연구는 보건 복지부 선도기술·의료공학 기술개발 사업 연구비 지원에 의한 것임

† 비 회 원 : 서강대학교 컴퓨터학과
lrk@grmanet.sogang.ac.kr

** 종신회원 : 서강대학교 컴퓨터학과 교수
ihm@ccs.sogang.ac.kr

논문접수 : 1999년 12월 23일

심사완료 : 2000년 6월 1일

(splatting)[4,5,6]이 이에 속하는데 이러한 방법은 효과적인 최적화 방법인 광선 샘플링의 조기 광선 종결을 구현하기 힘들다는 단점이 있다.

팔진트리(octree)나 피라미드(pyramid), k-d 트리(k-d tree) 등과 같은 계층구조를 갖는 자료구조들이 볼륨 데이터 내에 존재하는 공기와 같이 가시화 될 필요가 없는 부분에 대한 불필요한 계산을 피하기 위한 구조로 사용되었다[3,4,7]. 계층구조를 갖는 자료구조는 그 자료구조를 순회하는 작업에 추가적인 비용이 든다. 광선 추적법에서는 각 광선 마다 공기와 같은 영역에 대한 불필요한 샘플링을 피하기 위해 매번 팔진트리를 계층적으로 순회해야 하는 중복된 계산을 포함하게 된다. [8]에서는 계층구조의 순회에 드는 추가비용을 줄이기 위해 플랫폼 피라미드(flat pyramid)라고 하는 자료구조를 사용하였다.

혼합 기법인 쉬어 와핑(shear-warping)을 이용한 렌더링 알고리즘[9]은 영상 순서 렌더링과 물체 순서 렌더링의 장점을 가진 알고리즘이다. 이 방법은 볼륨 데이터를 일단 분석하여 가시화를 하고자 하는 복셀에 대한 런LENGTH 인코딩(run-length encoding)을 한 자료구조와 중간영상의 런LENGTH 인코딩을 구성하여 주사선에 기초한 자료구조(scanline based coherence data structure)를 사용하는데, 볼륨 데이터에서의 불필요한 영역에 대한 건너뛰고 중간영상에 대한 불투명영역에 대한 건너뛰기를 동시에 진행하여 빠르게 렌더링하게 된다.

본 논문에서는 팔진트리를 사용하는 광선 추적법에서 모든 광선이 팔진트리를 순회하는 중복된 계산을 피하면서 광선을 효율적으로 샘플링하는 알고리즘을 제안한다. 이 렌더링 방법은 팔진트리를 관찰자 시점에 따라 순서대로 한 번만 순회하고, 복셀 주변에서 슬라이스상에서의 2차원 샘플링의 지점을 계산하는, 물체 순서 광선 추적법이다. 특히 물체 순서 방법의 단점중의 하나인 가시성(visibility) 문제를 영상 공간에서의 동적 자료구조를 사용하여 해결하였다. 이 방법은 동적으로 물질을 선택하며 볼륨 데이터를 가시화하고자 할 때에도 추가적으로 많은 메모리를 사용하지 않으면서 빠르게 렌더링 할 수 있는 방법이다.

본 논문의 구성은 다음과 같다. 2장에서는 복셀 주변에서 샘플링 지점을 계산하는 방법과 이의 기반이 되는 2차원 샘플링을 설명한다. 3장에서는 2차원 샘플링에 기반한 물체 순서 광선추적법의 알고리즘을 설명한다. 4장에서는 실험환경과 실험 결과, 팔진트리를 사용한 광선 추적법과 쉬어 와핑 그리고 본 논문이 제안한 렌더링

방법에 대한 영상 생성 시간을 비교한다.

2. 물체 순서 광선 추적법 및 2차원 보간에 대한 소개

광선 추적법에서는 각각의 광선을 일정한 간격으로 3차원 샘플링을 하여 그 색상과 불투명도를 누적함으로써 영상을 얻게 된다. 본 논문에서 제안하는 광선 추적법에서 광선의 최초 샘플링지점은 볼륨 데이터와 최초로 만나는 첫 번째 슬라이스(slice)에서의 교차점이다. 샘플링 간격을 슬라이스들과 만나는 광선의 간격과 동일하게 하면 그 이후의 샘플링은 평행 투영의 경우 각 슬라이스 상에서 일어나게 된다. 이것은 색상과 투명도를 얻기 위한 2차원 샘플링을 가능하게 한다.

불필요한 데이터 영역에서의 샘플링을 피하기 위해서 볼륨 데이터에 팔진트리 구조를 추가하면 각각의 광선은 볼륨 데이터의 팔진트리를 순회하면서 샘플링지점을 구하게 된다[3]. 모든 광선이 팔진트리를 순회하며 샘플링 지점을 계산하기 위해서는 적지 않은 비용이 드는데, 이것을 해결하기 위해, 본 논문에서는 팔진트리를 한 번만 순회하며 볼륨 데이터 내에서 광선이 지나가는 샘플링 지점들을 찾아내는 물체 순서 렌더링 방법을 사용하였다.

2.1 영상 평면에서의 복셀의 위치를 결정하는 방법

물체 공간에서의 물체 순서 광선 추적법을 구현할 경우 주어진 복셀이 영상 평면 어디로 떨어지는가를 빠르게 계산을 하여야 한다. 영상평면으로 투영된 복셀의 좌표는 복셀의 좌표 값과 해당 투영 행렬의 곱으로 표현할 수 있다. 본 방법에서는 투영에 드는 비용을 줄이기 위하여 볼륨 데이터에 대한 관찰자 시점이 결정되었을 때, 볼륨 데이터가 존재하는 물체 공간에서의 기저 복셀 V_{000} 과 단위 벡터 \bar{X} , \bar{Y} , \bar{Z} 를 영상 평면에 투영을 한 값을 사용하여 투영점을 점진적으로(incrementally) 계산하도록 하였다.

볼륨 데이터의 임의의 복셀 V_{ijk} 는 기저 복셀을 기준으로 하여 다음과 같이 표현할 수 있다.

$$V_{ijk} = V_{000} + i \cdot \bar{X} + j \cdot \bar{Y} + k \cdot \bar{Z}$$

이 복셀이 영상 영역으로 투영된 지점은 기저 복셀이 투영된 지점 v_{000} 을 기준으로 하여 각 축의 단위 벡터를 영상 평면으로 투영하여 얻은 벡터 \bar{x} , \bar{y} , \bar{z} 를 확장하여 다음과 나타낼 수가 있고, 따라서 점진적으로 계산을 할 수가 있게 된다.

$$v_{ijk} = \begin{pmatrix} s_{ijk} \\ t_{ijk} \end{pmatrix} = v_{000} + i \cdot \bar{x} + j \cdot \bar{y} + k \cdot \bar{z}$$

영상 평면에서 필요한 값들을 미리 계산을 함으로써 렌더링 과정에서의 연산량을 줄이도록 하였는데 그림 1은 복셀 (1,3,2)이 투영되는 과정을 보여주고 있다.

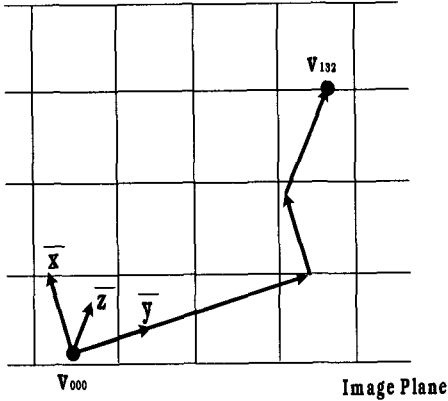


그림 1 영상평면으로 투영된 물체공간의 단위벡터로 구해지는 복셀 (1,3,2)의 영상 평면으로 투영된 위치

2.2 블록 데이터의 슬라이스에서 광선의 샘플링 지점을 결정하는 방법

복셀이 영상 평면으로 투영되는 위치는 일반적으로 정확한 광선의 지점에 위치하지 않고, 광선들 사이에 떨어지게 된다. 이 때, 현 복셀이 속한 슬라이스 내에서 복셀 주변의 샘플링지점을 구하기 위해서는 광선과 슬라이스간의 교차점을 계산하여야 한다. 이러한 계산을 효율적으로 수행하기 위하여 본 방법에서는 이미 복셀이 영상 평면으로 투영될 때 계산되어지는 정보를 이용하여 교차지점을 구하도록 하였다.

편의상 현재의 관찰자 시점에서 z축에 수직인 방향으로 xy 평면에 평행한 슬라이스들을 앞에서 뒤로 가면서 처리한다고 가정하자. 또한 영상 공간에서의 격자점이 화소의 중점을 나타낸다고, 즉 광선이 격자점을 통과한다고 가정하자. (i, j, k) 번째 복셀이 영상 평면으로 투영된 지점이 (s_{ijk}, t_{ijk})'라고 할 때, 영상 평면에서의 바로 왼쪽 아래에 있는 기준 화소, 즉 광선이 출발하는 지점까지의 거리는 다음과 같이 구해진다.

$$s' = s_{ijk} - (int)s_{ijk}$$

$$t' = t_{ijk} - (int)t_{ijk}$$

영상 공간에서의 s, t 축 각 방향으로의 단위 벡터를 \bar{s} 와 \bar{t} 라 할 때 위의 광선에 해당하는 샘플링 지점

은 \bar{s} 와 \bar{t} 를 물체 공간의 슬라이스로 역투영하여 얻어지는 벡터 $\bar{S} = (\bar{S}_x, \bar{S}_y, 0)'$ 와 $\bar{T} = (\bar{T}_x, \bar{T}_y, 0)'$ 를 이용하여 구할 수가 있다(그림 2). 복셀 V_{ijk}에 대응되는 영상 평면의 좌표 ((int)s_{ijk}, (int)t_{ijk})에 해당하는 광선의 k 번째 슬라이스 내에서의 샘플링 지점은 다음과 같이 구할 수 있다.

$$R_{ijk} = V_{ijk} - s' \cdot \bar{S} - t' \cdot \bar{T}$$

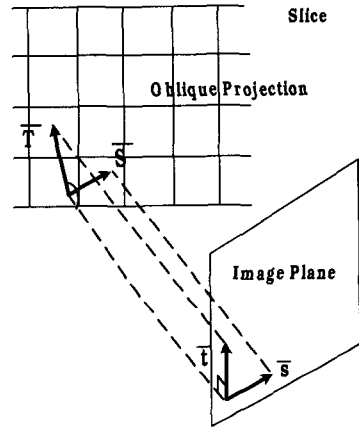


그림 2 영상 평면에서의 광선의 시작 위치를 결정하기 위한 광선 단위 벡터의 물체공간으로의 사선투영

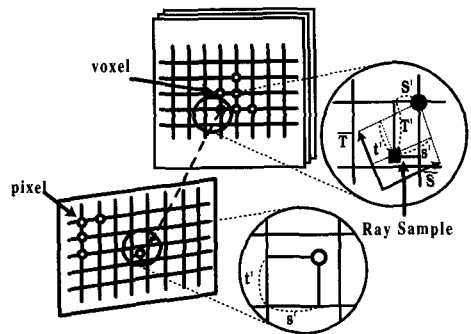


그림 3 샘플링 지점의 결정

이 계산은 \bar{S} 와 \bar{T} 를 미리 계산을 해놓을 경우 슬라이스와 광선의 교차점을 구하는 계산에 비해 간단하고, 계산 비용도 적으며, 또한 같은 슬라이스 내에 존재하는 주변 광선 샘플의 위치도 \bar{S} 와 \bar{T} 를 이용하여 점진적으로 구할 수 있게 된다.

그림 3은 한 복셀이 투영된 영상 평면에서, 광선이

시작되는 픽셀과 투영된 복셀 사이의 거리를 알아내고, 그것을 이용하여 특정 슬라이스에서 올바른 광선 샘플링 지점을 계산하는 과정을 보여준다. S' 와 T' 를 k 번째 슬라이스 상에서 (i, j, k) 복셀의 기준 샘플링 지점까지의 거리라 할 경우 $S' = s' \cdot \overline{S}_x + t' \cdot \overline{T}_x$ 와 $T' = s' \cdot \overline{S}_y + t' \cdot \overline{T}_y$ 와 같이 계산될 수 있다(그림 3).

주어진 복셀에 대하여 그것에 대한 기준 샘플 지점이 구해지면 그 복셀이 포함된 슬라이스에서 2차원 보간에 의해 색상과 불투명도 값을 구할 수 있다. 2차원 보간은 샘플 주위의 4개 복셀의 밀도 값과 그 안에서 위치한 비율을 가지고 계산되어진다.

이 샘플을 시작으로 이 복셀이 영향을 미치는 주변의 샘플 지점은 역 투영된 벡터 \overline{S} 와 \overline{T} 를 이용하여 점진적으로 구할 수가 있다. 즉 예를 들어 한 광선의 샘플링 지점에 대한 처리가 끝난 후 영상 공간에서 s 축 방향, 즉 가로 방향으로 바로 옆의 광선에 대한 샘플링 지점은 기존의 좌표에 \overline{S} 를 더함으로써 쉽게 계산될 수가 있다(그림 4). 한 복셀의 주변에서 처리해야 하는 샘플의 개수는 볼륨의 해상도와 렌더링을 하려하는 영상의 해상도에 의해 결정된다.

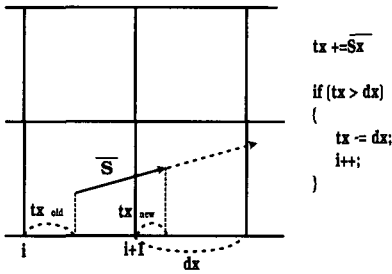


그림 4 다음 샘플링 지점의 결정과 그 때의 복셀의 인덱스와 셀 내의 위치

3. 2차원 샘플링에 기반한 물체 순서 광선 추적법 알고리즘

3.1 물체 순서 광선 추적법 알고리즘

본 논문에서 제안하는 물체 순서 광선 추적법 알고리즘은 볼륨 데이터에 대한 최소-최대 팔진트리 (min-max octree)의 노드들을 관찰자 시점을 기준으로 하여 앞에서 뒤로 가며 방문을 하면서 계산을 하게 된다(그림 5). 우선 관찰자 시점이 결정이 되었을 때 $-X, X, -Y, Y, -Z, Z$ 등 좌표축에 평행한 6개의

방향 중 투영 방향과 가장 평행한 방향을 결정하고 이후의 계산에서 방문하는 팔진트리 노드와 교차하는 슬라이스 세그먼트(segment)는 바로 이 방향으로 처리가 되게 된다. 그림 5의 알고리즘에서 기술된 바와 같이 우선 팔진트리의 루트 노드에서부터 시작하여 노드들을 재귀적으로 방문하게 된다. 우선 한 노드에 대하여 그 노드의 최소-최대값과 렌더링을 하려는 물체의 밀도의 최소-최대값을 비교하여 그 범위가 겹치지 않는 경우 그 노드에 대한 처리를 중단하고 다음 노드로 넘어가게 된다. 이렇게 함으로써 볼륨 데이터의 물체 공간에서의 응집성(coherence)을 쉽게 이용할 수 있게 된다. 물체 순서 광선 추적법의 경우 기존의 영상 순서 광선 추적법에서와는 달리 모든 광선에 대하여 팔진트리를 방문할 필요가 없이 전체 렌더링에 대하여 팔진트리가 표현하는 볼륨 데이터의 계층구조를 한번만 탐색하게 됨으로써 이에 대한 계산을 절약할 수가 있다. 만약 팔진트리 노드 방문 시 미리 설정한 레벨까지 내려 왔다면 계산 효율을 위하여 그 노드에 대응되는 슬라이스 세그먼트들을 앞에서 뒤로 가면서 처리하게 된다. 각 슬라이스 세그먼트에서는 2절에서 기술한 바와 같이 슬라이스 내에서 점진적으로 리샘플링 지점을 계산하여 2차원 보간을 통해 색상과 불투명도를 구한 후 이미지 평면에 합

```

OSRayCasting(node n, level l)
{
  if (The node n's min-max interval doesn't overlap
      that of voxels of interest) return;
  if (l is the lowest level in the min-max octree) {
    for each slice, clipped by n, in the front to back
      order {
        Resample rays corresponding to transparent pixels;
        for each resample {
          Shade and accumulate its color and opacity into
            the image plane;
          Update the opaqueness information for the image
            plane; /* Line (A) */
        }
      }
  }
  else {
    Project n into the image plane;
    Check if the projected region is opaque or not;
    /* Line (B) */
    if (it is not opaque) {
      Determine the traversal order of the eight
        subnodes;
      for each subnode sn in the front to back order
        OSRayCasting(sn, l+1);
    }
  }
}
    
```

그림 5 물체 순서 광선 추적법

성을 통하여 누적을 하게 된다. 광선 추적법을 물체 순서 방법으로 구현을 할 경우 영상 순서에서의 광선 추적법의 가장 효과적인 최적화 방법중의 하나인 조기 광선 종결을 구현하기 힘들게 된다는 단점이 있다. 이러한 문제점을 해결하기 위하여 본 논문에서는 팔진트리 순회 시 각 관심이 있는 데이터를 포함하는 노드들을 영상 평면으로 투영하여, 투영된 영역에 내재하는 화소들이 이미 불투명도가 1 또는 미리 정한 값(예를 들어 0.95) 이상이 되어 더 이상 리샘플링할 필요가 없다면 그 노드에 대한 불필요한 계산을 피할 수 있도록 하였다(그림 5, Line (B)). 그렇지 않은 경우 재귀적으로 그 노드의 자식 노드들을 앞에서 뒤 순서로 방문하게 된다. 위에서 기술한 물체 순서 광선 추적법에서 조기 광선 종결 기법과 같은 효과를 내는 기법을 효율적으로 구현하기 위해서는 광선에 대한 화소의 불투명도가 충분히 커졌을 경우 그러한 정보를 효과적으로 저장하여야 한다(그림 5, Line (A)).

그림 6에서 불투명데이터의 일정한 영역을 나타내는 각 노드에서 표현된 번호는 방문순서를 나타내는 것이다. 0*, 1*, 2*, 3* 순으로 노드들이 방문되고, 0* 에서는 다시 00, 01, 02, 03 순으로 방문된다. 00, 01, 02는 모두 비어있는 영역이기 때문에 광선을 리샘플링 할 필요가 없고, 03에서는 광선의 리샘플링 지점을 슬라이스 위에서 구하게 된다.

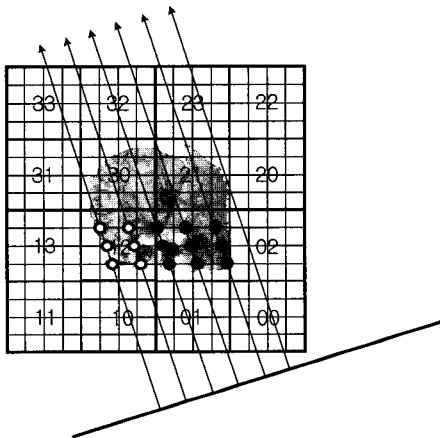


그림 6 팔진트리의 노드 방문과 슬라이스에서의 샘플링

3.2 범위 트리(Range Tree)를 이용한 광선 샘플링의 조기 종결 방법

3.1절에 기술한 물체 순서 알고리즘을 효과적으로 구

현하기 위해서는 그림 5의 Line (A)와 Line (B)에 해당하는 부분을 효율적으로 구현을 하여야 한다. 즉 2차원 영상의 화소의 불투명도 값이 동적으로 변화하는 상황에서 주어진 직사각형 영역에 대하여 그 안의 모든 화소들의 불투명도 값이 주어진 값보다 큰 지를 빠르게 검사할 수 있어야 한다. 또한 셰이딩이 된 칼라 값의 누적의 결과, 그 화소가 충분히 불투명해졌을 때 그러한 사실을 효율적으로 저장하여야 한다. [10]에서는 물체 순서 알고리즘인 스플래팅 방법에 대하여 범위 트리를 사용하여 효율적으로 렌더링 속도를 향상시켰다. 범위 트리(range tree)는 계산기하학에서 잘 알려진 자료구조로서 동적으로 변하는 데이터에 대하여 임의의 공간에 대한 검색을 효율적 할 수 있게 해주는 구조이다[11]. 본 논문에서는 이러한 범위 트리를 물체 순서 광선 추적법에 적용하여 그 속도를 향상시키도록 하였다.

$n \times n$ 해상도를 갖는 영상에서 2차원 범위 트리는 하나의 1차 트리(primary tree)와 $2n-1$ 개의 2차 트리(secondary tree)로 구성할 수 있다. 1차 트리는 Y축에 대한 구간을, 2차 트리는 X축에 대한 구간을 정의하며, 각각 2진 트리로 구성된다. 본 논문에서 사용하는 범위 트리의 각 노드는 이미 충분히 불투명해져 색상과 불투명도의 누적이 필요하지 않은 불투명한 지역을 이루는 자식 노드의 개수를 나타내도록 구성되어 있다.

범위 트리에서의 투영 영역에 대한 정확한 불투명도 검사는 투영된 영역을 구성하는 범위 트리에서의 정규 지역에 대응되는 노드들을 찾고 이것의 값을 검사함으로써 이루어진다. 그림 7은 4×4 해상도의 영상을 범위 트리로 구성한 예를 보여주고 있다. 이 그림의 굵은 선으로 둘러싸인 3×3 에 해당하는 투영영역은 2×2 영역

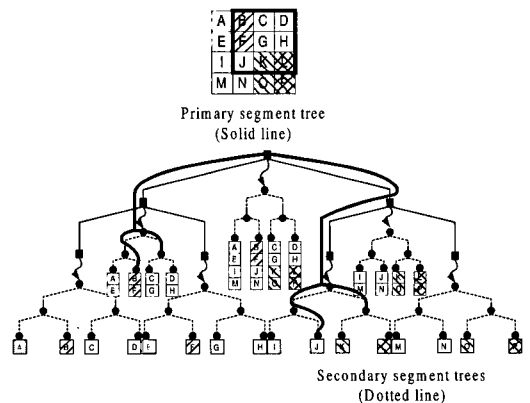


그림 7 범위 트리의 예

(CDGH) 1개와 2×1영역(KL) 1개, 1×2영역(BF) 1개, 1×1영역(J) 1개 등 4개의 정규지역(standard rectangle)들로 구성되어 있다. 따라서 이 영역 전체가 불투명한지를 검사하기 위하여 그 안의 9개의 화소 각각의 불투명도를 검사할 필요가 없이 범위 트리의 4개의 노드를 검사함으로써 효과적으로 계산을 할 수가 있다. 그림 8은 7x7 투영영역에 대한 검사를 49개의 화소가 아닌 9개의 범위 트리 노드를 검색함으로써 수행할 수 있음을 보여주고 있다.

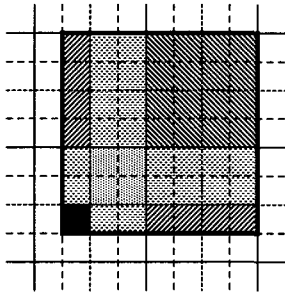


그림 8 7X7 투영영역에 대한 검사시 범위 트리에서 9개의 노드검색

영상평면의 불투명도 값을 범위 트리로 구성할 경우 투영영역은 다양한 크기의 정규지역으로 구성되어 있다. 각 정규지역은 범위 트리의 노드로서 존재하므로 트리 순회를 통해 이들 정규지역들을 찾아내고 불투명도 검사를 해야 한다. 본 논문에서는 팔진트리의 노드를 처리할 때 그것을 투영한 영역의 바운딩 박스를 구하여 팔진트리 노드에 대한 투영영역으로 사용하였다. $n \times n$ 의 영상해상도에 대한 범위 트리에서 특정한 팔진트리 노드의 투영영역의 크기가 $m \times m$ 일 때, 범위 트리에서 임의의 영역에 대한 불투명도 검사는 1차 트리에서 Y축에 대한 정규구간을 찾고 다시 그 정규구간내의 2차 트리에서 X축에 대한 정규구간을 찾아야 하므로 $O(\log^2 m)$ 에 수행할 수 있다. 이는 m^2 의 화소의 불투명도를 검색하는 것보다 더 빠른 시간에 할 수 있음을 알 수 있다.

한편 불투명 데이터를 리샘플링하여 색상과 불투명도를 영상 버퍼에 누적할 때, 화소의 불투명도에 대한 정보에 변화가 생겼을 경우 범위 트리의 해당 노드들에 대한 값도 효율적으로 갱신을 하여야 한다. 팔진트리의 탐색 중 미리 지정한 레벨까지 내려왔을 경우 해당 노드의 복셀 영역 (예를 들어 2x2x2 또는 4x4x4)에 대하여 2차원 리샘플링 및 색상 합성을 하게 되고 이때 필요에 따라 범위 트리를 갱신하여야 한다. 현재 만들고 있는

영상의 해상도가 $n \times n$ 이고 노드의 투영영역의 크기가 $m \times m$ 이라 하자. 이 노드 처리 결과 불투명해지는 화소의 개수가 p 개라 할 경우 이 노드처리를 위한 갱신 비용은 $O(p \cdot \log n \cdot \log m + \log^2 n)$ 이 됨을 증명할 수 있다[10]. 이 경우 p 와 m 은 일반적으로 작은 값이고 이 복잡도는 최악의 경우에 대한 것으로서 실제 평균 검색 비용은 훨씬 적게 든다. 한 화소가 불투명해졌을 경우 그 화소와 그 화소를 포함하여 불투명해지는 정규 지역에 대한 노드들을 리프 노드에서 루트 노드 방향으로 올라가며 노드 값들을 갱신하게 된다. 따라서 위의 복잡도에서 $\log n$ 은 바로 최악의 경우 1차 트리의 루트까지 가야하는 (이 경우는 영상의 모든 화소의 값들이 불투명해짐을 의미) 상황을 나타내지만 그러한 경우는 거의 없고 일반적으로 $\log n$ 보다 적은 값이므로 평균 복잡도는 훨씬 적게 된다.

불투명한 영역에 떨어지는 팔진트리의 노드에 대한 리샘플링은 영상평면의 스캔라인을 따라 수행을 하게 되는데 이 영역에서 불투명한 화소를 빠르게 건너뛰기 위하여, 각 스캔라인에 대한 간단한 런LENGTH 인코딩(run-length encoding)을 사용하여 그 효율을 높였다.

4. 실험결과

본 논문에서는 기존의 팔진트리를 이용한 광선 추적법 알고리즘과 2차원 샘플링에 기반을 둔 물체 순서 광선 추적법 알고리즘을 구현하고 그 렌더링 성능을 비교하였다. 실험은 MIPS 195MHz R10000 CPU와 256MBytes 주기억장치를 가진 SGI 워크스테이션 상에서 수행하였다. 표1의 실험데이터의 BB는 사람머리에 대한 MRI 데이터로 256×256×167의 해상도를 갖고 BH는 CT 데이터로서 256×256×225의 해상도를 갖는다(그림 10). BONESKIN의 경우 BH 데이터에 대하여 뼈는 불투명하게 그리고 피부는 투명하게 렌더링을 하도록 하였다. 256×256×110의 해상도를 갖는 ENGINE은 엔진에 대한 레이저 스캔 데이터이다. 이 세 가지 데이터는 모두 미국 노스캐롤라이나 대학교(UNC)에서 만든 실험 데이터이다. VHBONE과 VHSKIN은 미국의 NLM (National Library of Medicine)에서 만든 Visible Man 데이터를 전처리한 것으로서 587×341×1878의 해상도를 갖는다[12]. VHBONE은 Visible Man 데이터의 불투명한 뼈만을 가시화 하는 경우이고 VHSKIN은 불투명한 피부만을 가시화 하는 경우이다. Visible Man 데이터는 크기가 방대하여, 데이터를 일정한 블록으로 나누어 블록 단위의 렌더링을 하고, 부분적으로 렌더링이 된 부분 영상을 하나의 영상 버퍼로 모

아 나타내었다.

표 1 팔진트리를 사용한 광선 추적법과 2차원 샘플링에 기반을 둔 광선 추적법의 영상 생성을 위한 렌더링 시간의 비교

데이터	해상도	이미지 해상도	팔진트리-3D	2D 리샘플링	스피드업	쉬어 와핑
BB	256X256X167	256X256	6.92	0.75	9.23	0.51
"	"	512X512	25.51	1.72	14.83	0.55
BH	256X256X225	256X256	14.49	2.17	6.68	0.78
"	"	512X512	49.80	6.34	7.86	0.82
BONESKIN	256X256X225	256X256	21.82	4.65	4.70	2.02
"	"	512X512	67.44	15.94	4.23	2.12
ENGINE	256X256X110	256X256	9.87	2.28	4.34	1.33
"	"	512X512	34.70	8.53	4.07	1.40
VHBONE	587X341X1878	200X680	31.46	7.20	4.37	-
VHSKIN	587X341X1878	200X680	53.34	21.42	2.49	-

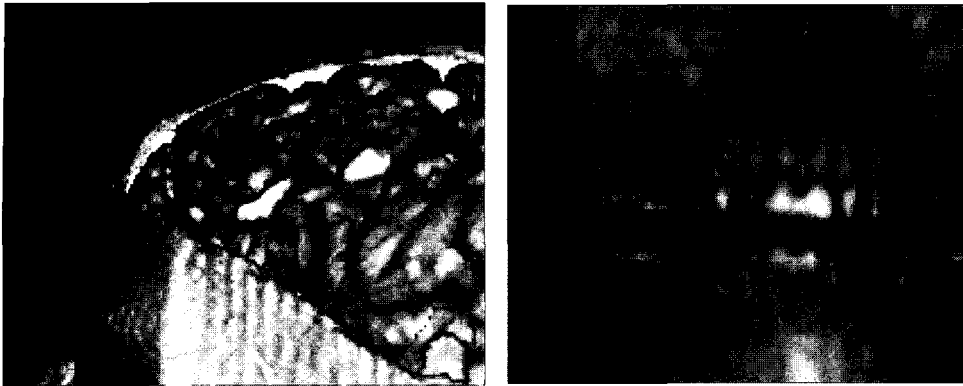
표1에서 팔진트리-3D 열과 2D 리샘플링 열은 각각 기존의 팔진트리와 초기 광선 종결 기법을 적용한 광선 추적법과 본 논문에서 제안한 2차원 샘플링에 기반을 둔 물체 순서 광선 추적법을 사용하여 렌더링한 평균 시간을 나타낸다. 또한 쉬어 와핑 열은 쉬어 와핑 렌더링 방법을 사용한 경우의 평균 렌더링 시간으로 위의 두 방법과의 성능 비교를 위하여 실험을 하였다. UNC 데이터의 경우, 같은 데이터에 대해 256×256과 512×512의 크기를 갖는 영상을 생성하여 실험하였으며 VHBONE과 VHSKIN의 경우 데이터의 방대함으로 인하여 실험을 하지 못하였다. 표1에서 스피드업 열은 팔진트리-3D의 시간을 2D 리샘플링의 시간으로 나눈 값으로서 기존의 광선 추적법에 대한 2차원 샘플링에 기반을 둔 물체 순서 광선 추적법의 속도향상을 나타내는 값이다. 이 실험 결과에서 볼 수 있듯이 모든 데이터에 대하여 좋은 속도향상을 보이고 있다. 영상의 질 또한 3차원 리샘플링을 하는 광선 추적법에 못지 않은 좋은 질의 영상을 생성한다.

본 기법과 직접 블룸 렌더링 방법 중 가장 빠른 방법으로 알려진 쉬어 와핑[9]과의 비교를 위하여 같은 조건에서 쉬어 와핑 방법을 이용하여 영상을 생성하여 보았다. 쉬어 와핑의 경우 팔진트리를 사용하여 빠른 분류를 한 후 쉬어 와핑 기법을 적용하는 방법에 대한 실험을 하였다. 이는 본 논문이 제안한 방법과 같이 물체에 대한 분류가 동적으로 이루어질 수 있다는 조건을 맞추기 위함이다. 쉬어 와핑의 경우 256×256 해상도에 대하여 평균 2-3배 정도 빠르게 영상을 생성한다(표 1). 또한 쉬어 와핑 방법의 특성상 렌더링 시간이 영상의 해상도가 아닌 블룸 데이터의 해상도에 비례하므로 512×512 해상도를 갖는 영상을 생성할 때는 광선 추적법

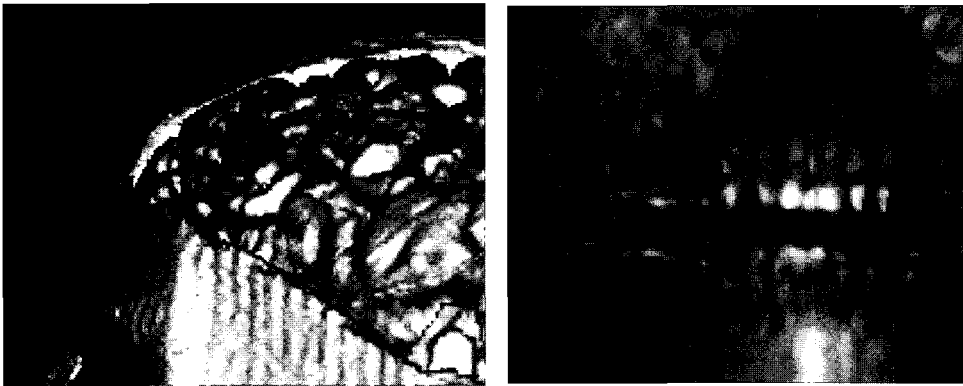
보다 훨씬 빠름을 알 수 있었다. 반면에 해상도가 커짐에 따라 영상의 질에 있어서는 기존의 광선 추적법과 본 논문이 제안한 방법에 비해서 떨어지게 된다. 쉬어 와핑을 이용한 렌더링 방법에서는 블룸 데이터의 슬라이스에 위치한 4개의 복셀 사이에서 한 번의 리샘플링만을 수행하여 중간 영상을 만들고, 이에 대하여 최종 영상을 생성하기 위하여 와핑을 하는 과정에서 중간 영상 공간에서 생성하려는 최종 영상의 해상도에 따라 다시 리샘플링을 하게 된다. 따라서 최종 해상도에 맞게 블룸 데이터가 존재하는 물체공간에서 직접 리샘플링을 하는 광선 추적법보다 영상의 질이 떨어지게 된다. 이 현상은 최종적으로 생성되는 영상의 크기가 블룸데이터의 해상도에 비해 커질 경우 더욱 극명해진다.

원래의 광선 추적법과 본 논문이 제안한 2차원 샘플링에 기반을 둔 광선 추적법은 영상의 해상도에 따라 블룸 데이터 내에서 세밀하게 리샘플링이 되는 것에 비해서, 쉬어 와핑을 이용한 렌더링 방법은 중간 영상의 생성 시 블룸의 해상도에만 의존하여 샘플링 한 후, 와핑 과정을 통해서 최종영상의 해상도에 맞는 영상을 생성해 내므로, 여전히 블룸 데이터 내에서 이루어지는 샘플링의 개수는 최종 영상의 해상도에 미치지 못하게 되므로 영상의 질이 떨어지게 되는 것이다. 그림 9의 (a)~(c)는 세 가지 방법으로 BB 데이터와 BH 데이터에 대하여 512×512 영상을 생성하여 그 부분 영상을 비교하고 있다. 전체적으로 본 논문에서 제안한 기법은 기존의 광선 추적법과 상당히 유사한 영상을 생성하는 반면 쉬어 와핑을 사용할 경우 그 만큼 선명하지 못한, 약간 부에지는(blurring) 느낌을 주는 영상을 생성함을 알 수 있다. 특히 오른쪽의 BH 데이터의 경우 오른쪽 뺨과 입 부분(그림의 왼쪽 가운데)을 자세히 보면 본 논문에서 제안하는 방법에서는 원래의 광선 추적법으로 생성한 영상과 같이 상당히 정확하게 렌더링이 되지만, 쉬어 와핑의 경우 영상이 부에짐에 따라 그 부분에서의 자세한 구조를 정확하게 표현을 하지 못하게 된다. 이러한 현상은 쉬어 와핑의 방법 자체로부터 기인한 것으로 인하여 해상도가 커질수록 더 심각한 문제가 발생하게 된다.

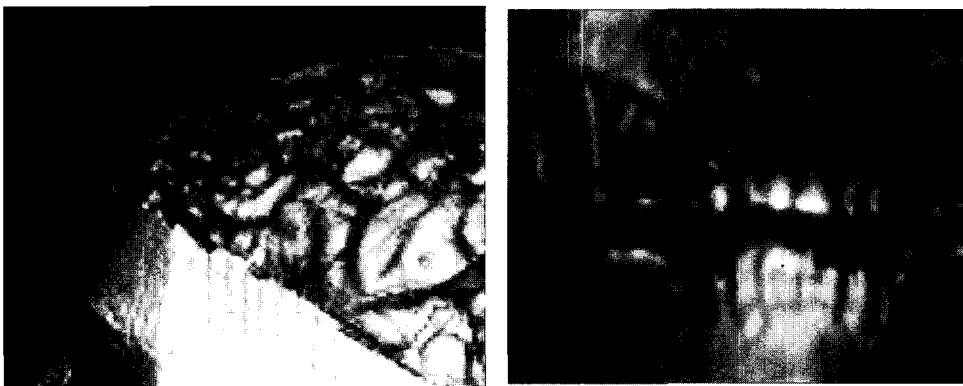
본 논문에서 제안한 2차원 샘플링에 기반을 둔 광선 추적법은 기존의 광선 추적법과 쉬어 와핑 기법 사이의 성능 차이를 메울 수 있는 방법이라 할 수 있다. 즉 쉬어 와핑 기법 보다는 느리나 더 우수한 영상을 생성하는 광선 추적법으로서, 쉬어 와핑 기법 보다 구현하기가 쉽고 또한 전자와는 달리 2차원 범위 트리 표현에 필요한 적은 양의 메모리만 추가적으로 필요로 하기 때문에



(a) 광선 추적법으로 생성한 영상들



(b) 2차원 샘플링에 기반한 광선 추적법으로 생성한 영상들

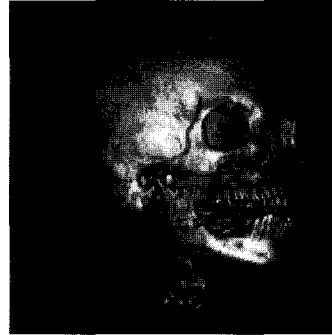


(c) 쉬어 와핑 방법으로 생성한 영상들

그림 9 광선 추적법, 2차원 샘플링에 기반한 광선 추적법 그리고 쉬어 와핑을 이용한 렌더링 방법으로 생성된 512×512 영상에서의 부분 영상의 비교



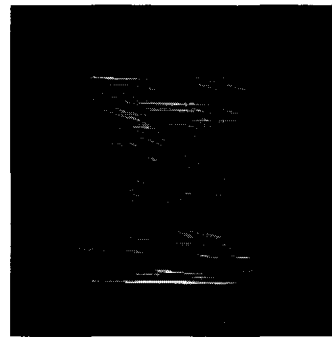
(a) MRI Human Brain (BB)



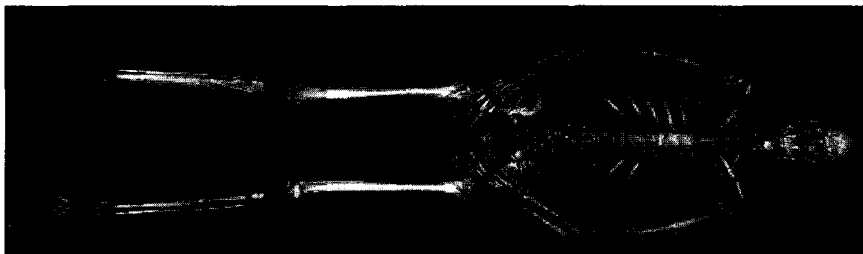
(b) CT Human Head (BH)



(c) CT Human Head with Semitransparent Skin (SKIN)(ENGINE)



(d) CT Engine



(e) Visible Human Data with Bone (VHBONE)



(f) Visible Human Data with Skin and Bone (VHSKIN)

그림 10 2차원 리샘플링에 기반을 둔 광선추적법으로 생성한 영상들

상황에 따라 효과적으로 사용이 될 수 있을 것이다.

5. 결론

본 논문은 기존의 광선 추적법이 갖는 중복된 계층구조의 순회와 3차원 리샘플링의 부담을 줄이기 위해 2차원 리샘플링에 기반을 둔 물체 순서 광선 추적법을 제안하였다. 물체 순서 광선 추적법은 계층구조를 오직 한번만 방문하면서 관심이 있는 복셀의 주변에서 리샘플링 지점을 결정하는 방식이다. 2차원 리샘플링은 3차원 리샘플링에 비하여 더 좋은 영상을 생성하지는 않으나, 3차원 리샘플링에 못지 않은 비교적 좋은 영상을 생성하며, 물체 순서 광선 추적법을 더욱 효율적으로 수행할 수 있게 한다. 또한 본 논문에서는 물체 순서 볼륨 렌더링에서 구현하기 힘든 광선 샘플링의 초기 광선 종결을 효율적으로 해결하기 위해 영상공간에서의 자료구조 이용을 제안하였다. 본 논문이 제안한 방법은 임의의 물체에 대한 가시화나 임의 방향 관찰에 대해서도 효과적으로 수행이 되며, 추가적으로 많은 메모리를 사용하지 않으면서도 효과적으로 렌더링 할 수 있는 방법이라 할 수 있다. 리샘플링의 비용을 줄이기 위해 가장 가까운 복셀에서의 색상 값과 투명도 값을 쓸 수도 있는데, 이러한 방법은 더 빠른 영상 생성을 가능하게 할 것이다.

참고 문헌

- [1] Todd Elvins, "A Survey of Algorithms of Volume Visualization," *Computer Graphics*, Vol. 26, No. 3, pp. 191-201, August 1992
- [2] M. Levoy, "Display of Surfaces from Volume Data," *IEEE Computer Graphics and Applications*, Vol. 8, No. 5, pp. 29-37, May 1988.
- [3] M. Levoy, "Efficient Ray Tracing of Volume Data," *ACM Transactions on Graphics*, Vol. 9, No. 3, pp. 245-261, July 1990.
- [4] D. Laur and P. Hanrahan, "Hierarchical Splattings : A Progressive Refinement Algorithm for Volume Rendering," *Computer Graphics*, Vol. 25, No. 4, pp. 285-288, 1991.
- [5] L. Westover, "Footprint Evaluation for Volume Rendering," *Computer Graphics*, Vol. 24, No. 4, pp. 367-376, 1990.
- [6] I. Ihm and R. Lee, "On Enhancing the Speed of Splattings with Indexing," *In Proceedings of IEEE Visualization 95*, pp. 27-34, Atlanta, GA, October 1995.
- [7] R. S. Subramanian and D. S. Fussell, "Applying Space Subdivision Techniques to Volume Rendering," *In Proceedings of Visualization '90*, pp. 150-159, 1990.
- [8] D. Cohen and Z. Sheffer, "Proximity Clouds - An Acceleration Technique for 3D Grid Traversal," *The Visual Computer*, Vol. 11, pp. 27-38, 1994.
- [9] P. Lacroute and M. Levoy, "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation," *In Computer Graphics (SIGGRAPH 94 Proceedings)*, pp. 451-458, 1994.
- [10] R. Lee and I. Ihm, "On Enhancing the Speed of Splattings Using Both Object- and Image-Space Coherence," *Graphical Models (formerly Graphical Models and Image Processing)*, Vol. 62, No. 4, pp. 263-282, July 2000.
- [11] F. P. Preparata and M. Ian Shamos, *Computational Geometry : Introduction*, Springer-Verlag New York Inc., 1995.
- [12] NLM, http://www.nlm.nih.gov/research/visible/visible_human.html, 1997.



이래경

1993년 2월 덕성여자대학교 전산학과 학사. 1995년 8월 서강대학교 컴퓨터학과 석사. 1995년 9월 ~ 현재 서강대학교 컴퓨터학과 박사과정 재학중. 관심분야는 컴퓨터 그래픽스, Volume Visualization, Volume Rendering, ImageBased Rendering, Collaboration, Multi-Resolution Rendering 등.



임인성

1985년 서울대학교 계산통계학과 졸업(이학사). 1987년 Rutgers University 전자계산학과 졸업(이학석사). 1991년 Purdue University 전자계산학과 졸업(이학박사). 1992년 ~ 1993년 Purdue University 전자계산학과 연구원. 1999년 ~ 2000 Univ. of Texas at Austin TICAM 방문연구원. 1993년 ~ 현재 서강대학교 컴퓨터학과 부교수. 관심분야는 컴퓨터 그래픽스, Scientific Visualization, High Performance Computing 등임.