

분석적 광선 추적법을 이용한 체적시각화 시스템

박 현 우[†] · 백 두 원^{††} · 정 문 렬^{†††}

요 약

본 논문에서는 체적 시각화 과정을 이론적으로 고찰하여 시각화 모델을 제시하고 그 모델로 유도된 관계로부터 미분 방정식을 이용하여 분석적 체적 시각화 해법을 구하였다. 이 분석적 방법을 체적 시각화의 대표적인 방법인 Levoy의 이산적 광선 추적법과 비교하여 본 연구에서 제시한 방법의 특수한 형태가 Levoy의 이산적 방법임을 보였다. 그리고 체적 데이터를 시각화하기 위해서는 사용자가 시각화하기를 원하는 부위를 선택하고 이 부분만을 추출하는 영역 분할 작업이 필요하다. 본 논문에서는 영역확장법에 기반을 둔 효율적인 3차원 영역 분할 기법을 개발하여 위의 분석적인 방법을 이용하여 3차원 체적 데이터의 시각화를 위한 시스템을 구현하였다. 그리고 본 접근법에 대한 의의와 유용함에 대한 가설적인 결론을 구현된 시스템을 이용한 실험에 근거하여 유도하였다. Levoy의 이산적인 방법과 분석적인 방법을 같은 데이터에 대해 3차원 영역 분할 수행 후 적용한 실험은 분석적인 방법이 이산적인 방법에 비해 렌더링된 이미지의 질이 더 좋음을 보여준다.

Volume Visualization System Using an Analytical Ray Casting

Hyun-Woo Park[†] · Doo-Won Paik^{††} · Moon-Ryul Jung^{†††}

ABSTRACT

When volume data is visualized by the ray casting method, the color value of each pixel in the image is obtained by composing the color contributions of the sample points that lie on the ray cast from the pixel point. In most ray tracing methods including Levoy's classical method, the color composition is formulated as a summation of the color contributions of the discrete sample points. However, the more precise color composition is formulated as differential equations over the color contributions of the continuous sample points. The discrete formulation is used, because analytical solutions to the continuous formulations are hard to find. In this paper, however, we have discovered a semi-analytical solution to the continuous formulation of a typical ray tracing of volume data. We have applied both Levoy's method and ours to the same set of data, and compared the visual quality of both results. The comparison shows that our method produces a more fine-grained visualization of volume data.

1. 서 론

데이터 시각화의 목적은 데이터가 가지고 있는 정보를 사용자에게 효율적으로 보여주는 것이다. 시각화의 중요성은 특히 의료 영상 데이터를 포함하는 3차원 체적 데이터의 경우 더욱 부각된다. 3차원 체적 데이터

는 CT나 MRI 등으로부터 얻어지는 체적 원소(volume element, voxel)들이 이루는 3차원 밀도 분포(3D density field)를 말한다. 이 밀도 분포는 추상적인 물리량을 나타내고 있기 때문에 그 자체로는 가시적이지 않다. 밀도 분포를 시각화시킨다는 말은 밀도 분포 영상을 생성할 수 있는 어떤 가시적 물체나 현상으로 바꾸어 표현하고 이로부터 영상을 만들어 낸다는 것을 의미한다. 즉 어떤 가시적 중간 표현 방식(intermediate representation)이 요구된다. 체적 시각화 방법은 여러

† 준 회원 : 숭실대학교 대학원 컴퓨터공학과
 †† 정 회원 : 숭실대학교 컴퓨터공학부 교수
 ††† 정 회원 : 서강대학교 영상대학원 디지털영상학과 교수
 논문접수 : 1998년 6월 23일, 심사완료 : 2000년 1월 9일

가지가 있으나[1, 2, 6, 7, 8, 10, 12] 보통 이 가지적 중간 표현 방식에 따라 크게 표면 기반 기법(surface-based techniques)과 반투명 체적 기법(semi-transparent volume techniques)으로 나뉜다. 표면 기반 기법에서는 관심 부위를 나타내는 불투명한 표면을 삼각형이나 일반 다각형의 집합으로 표현한 후, 이 다각형 면을 적당한 음영 계산법을 이용하여 렌더링 한다. 반면 반투명 체적 기법은 체적 데이터를 가시적인 반투명 응고체(gel)로 간주하고, 이로부터 하나 또는 다수의 관심 부위를 주변 부위 또는 물체를 배경으로 하여 디스플레이 하는 것이다. 이 방법은 체적 데이터 내의 모든 체적 원소를 다 고려하여 렌더링 한다는 점에서 표면 기반 기법과 근본적으로 다르다. 반투명체 기법중 대표적 방법은 광선투사법(ray casting)을 이용한 방법으로 direct volume rendering으로 불린다. 이 방법은 3차원 볼륨의 가시화를 위해 어떠한 중간 단계의 기하학적인 구조도 만들지 않고 각각의 복셀(voxel)을 직접 처리하는 방법으로 이미지의 각 픽셀로부터 반투명한 3차원 체적으로 평행 광선을 투사한다. 광원으로부터 나온 빛은 물질에 도달하여 반사하는데 이미지 상의 각 픽셀에서의 색깔은 이 픽셀 방향으로 반사되는 모든 빛의 합이다. 이미지의 픽셀에서 발사되는 광선 상에 있는 물질의 각 점에서 그 픽셀 방향으로 반사되는 빛은 그 점과 픽셀 사이의 반투명 물질을 투과하면서 그 강도를 어느 정도 잃어버린 후 픽셀에 도달한다. 이것을 고려하여, 물질의 한 점이 픽셀의 색에 미치는 영향을 계산할 때, 그 점에서 반사된 색의 강도와 그 점과 픽셀 사이의 누적 투명도의 곱으로 표시된다. Levoy의 방법[3, 4]을 포함한 대부분의 일반적인 광선 추적법에서는 광선이 지나가는 선상에서 등간격으로 추출한 각 점에서의 색을 합하여 구하는 이산적인 방법을 사용한다. 그러나 가장 자연스럽고, 정확한 색의 계산은 이산적인 방법이 아니라 연속적인 방법으로 미분 방정식의 해를 구하는 문제로 귀결 될 수 있고 이 미분방정식의 해는 적분 형태로 나타내어진다. 이산적인 방법이 사용되는 이유는 연속 방정식의 분석적인 해를 구하기 어렵기 때문이다. 하지만 본 논문에서는 전통적인 광선 추적법에서 연속 방정식의 분석적인 해를 근사시켜 구하는 분석적인 방법을 개발하였다. 즉, 이산적 알고리즘으로부터 누적 불투명도와 색의 계산을 위한 미분 방정식을 미분 불투명도(differential opacity)를 이용하여 유도하고 이 미분방정식의 해인 적분 값을 분석적

인 방법으로 구하였다. 부정적분이 알려진 함수로 존재하지 않으므로 이 적분값 역시 근사값으로 밖에 구할 수 없지만 본 연구에서는 근사치를 구하는 영역을 국지화 시킬 수 있는 관계를 찾아내어 전체가 아닌 그 작은 주변에서만 근사하고 나머지의 전 영역에서는 정확한 값을 구하여 근사치의 오차가 적어지도록 했다. 분석적인 방법은 무수히 많은 표본을 추출한 Levoy의 방법과 동일하고 따라서 이산적인 방법보다 더 자연스럽고, 정확한 이미지를 생성하리라 여겨진다. 본 논문에서는 Levoy의 이산적인 방법과 분석적인 방법을 같은 데이터에 대해 적용하여 결과를 비교했고 이 실험은 위의 예상이 옳음을 보여준다. 완성된 체적 데이터 시각화 시스템에는 사용자가 시각화하기를 원하는 부위를 선택하고 이 부분만을 추출하는 영역 분할 기능이 필요하다. 삼차원 물체의 영역의 구분은 그 밀도 함수값으로 나타내어지므로 사용자는 밀도함수값을 지정하여야 한다. 하지만 원하는 부위는 보통 단일 밀도 값으로만 이루어져 있지 않고 또한 지정한 밀도 값을 갖는 복셀이 원하는 부위가 아닐 수도 있다. 따라서 이 영역 분할은 많은 장시간의 수작업을 필요로 한다. 본 논문에서는 사용자가 시각화하기를 원하는 부위는 서로 연결된 영역으로 이루어져 있다는 사실에 주목하여 영역확장법에 기반으로한 3차원 영역 분할 기법을 개발하여 구현했다. 체적 데이터는 크기가 커서 많은 양의 메모리와 처리 시간을 요구하므로 효율성에 중점을 두었다.

2. 3차원 영역 분할

영역 분할이란 주어진 데이터에서 의미 있는 부분만을 추출해 내는 것을 말한다. 3차원 체적 데이터를 시각화하고자 할 경우 사용자가 보고자 하는 부분만을 정확하게 시각화하기 위해서는 이와 같은 영역 분할 작업이 필요하다. 만약 영역 분할 작업 없이 시각화하기를 원하는 밀도 값에 불투명도를 높게 할당하여 영역을 구분하는 방법을 사용한다면, 사용자에게 관심이 없는 영역이지만 비슷한 밀도 값을 갖는 다른 영역도 함께 시각화 과정에 포함되어 원하지 않는 결과를 가져올 수 있다. 예를 들어 사람 뇌의 내부에 일정한 밀도 값을 갖는 환부가 있어서 이를 시각화하기를 원하는 경우, 환부의 밀도 값이 두개골의 밀도 값과 비슷하다면 실제로 보여지는 부분은 환부가 아니라 그 주

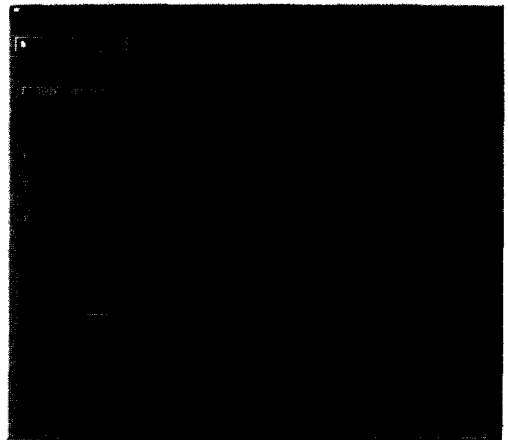
의를 둘러싸고 있는 두개골이 된다. 이와 같은 문제는 사용자가 원하는 부분만을 정확하게 추출해 내는 영역 분할을 통해서 해결할 수 있다.

영역 분할 방법에는 여러 가지가 있는데, 3차원 의료 데이터를 시각화하는데 있어 사용자가 시각화하기를 원하는 부위는 서로 연결되어 있다는 사실에 주목하여 본 논문에서는 영역확장법에 기반한 3차원 영역 분할 기법을 개발하여 사용하였다. 영역확장법은 사용자가 선택한 시작점(seed point)을 중심으로 그 주변에 같은 밀도 값을 갖는 점들을 영역에 포함시키면서 점단위로 재귀적으로 확장해 나가는 방법이다. 이 방법을 본 논문에서 구현한 선단위 영역 확장 법과 비교하기 위해서 점단위 영역확장법이라 부르기로 한다. 2차원 점단위 영역확장법을 3차원으로 확장시키는 것은 새로운 축에 대한 영역 확장 부분을 추가함으로써 간단하게 구현할 수 있다. 하지만 점단위로 같은 영역인지를 판단하는 비교를 하기 때문에 중복되는 비교 횟수가 많아져 수행 속도가 떨어진다. 또한 3차원 영역 분할 정보를 저장하기 위해서는 많은 양의 메모리를 요구하게 된다. 이러한 문제점을 피하기 위해서 본 논문에서는 선단위로 정보를 저장하고 확장하는 선단위 영역확장법을 사용하였다. 선단위 영역확장법도 기본적으로는 점단위 영역확장법과 방법은 같지만 선분 단위로 정보를 저장함으로써 점단위로 할 때보다 영역 판단을 하는 중복 비교 횟수를 줄일 수 있다. 그리고 3차원 체적 데이터에 대한 영역 분할 정보를 저장하는데 있어서도 불필요한 영역을 제외하고 필요한 영역에 대한 정보만을 압축하여 저장함으로써 메모리 사용량을 줄일 수 있다.

2.1 영역 선택과 분할

영역 분할을 하기 위해서는 먼저 사용자 자신이 보고자 하는 영역을 선택해야 한다. 영역이란 같은 밀도 값을 갖는 연속된 부분이라고 정의할 수 있다. 하지만 3차원 체적 데이터에서 한 영역은 보통 단일 밀도 값으로 이루어져 있지 않기 때문에 영역 분할을 하기 위해서는 영역에 대한 충분한 정보가 필요하다. 본 논문에서 구현한 시스템에서는 영역에 대한 충분한 정보를 얻기 위해 사용자가 보고자 하는 영역 내의 일정 부분을 선택하게 하였다. (그림 1)에서와 같이 뇌를 시각화하기 위해 보고자 하는 부분 안에 사각형을 위치시켜 영역을 선택했으며 이러한 선택은 여러 곳에 지정할

수 있다. 사용자가 선택한 사각형 내의 점들의 밀도 값은 영역 분할 수행시 어떤 점이 원하는 영역에 속하는가를 판단하기 위한 정보로 사용한다. 즉 선택한 사각형 내의 모든 픽셀들의 밀도 값을 이용해 영역 판단에 기준으로 사용할 평균값 m 과 분산 σ 을 구하고, 해당 픽셀과 주변에 이웃한 8점의 평균 밀도 값이 $[m-t\sigma, m+t\sigma]$ 의 범위 내에 있을 경우 원하는 영역 안에 포함된다고 판단한다. 영역 선택이 끝나면 영역 확장을 시작할 시작점을 선택하고 2차원 영역 분할을 시작한다.



(그림 1) 영역 선택

영역 분할된 결과는 (그림 6)의 (a)와 같이 나타난다. 만약 2차원 영역 분할의 결과가 원하는 영역이 아닐 경우 t 값을 적당히 조절하여 포함될 영역의 범위를 설정할 수 있다. 그런 후에 2차원 영역 분할 결과에 만족하면 3차원 영역 분할을 하게 된다.

2.2 기본적인 영역확장법

2차원에서 점단위 영역확장법은 주어진 시작점(Seed Point)을 중심으로 이웃한 모든 방향으로 확장해 가면서 같은 영역이라고 판단되면 그 이웃한 점을 집합에 포함시키고 더 이상 포함시킬 이웃한 점이 없을 때까지 재귀적으로 수행하는 방법이다. 이웃한 점은 사용자가 선택한 점 주변에 y 축으로 상·하, x 축으로 좌·우에 있는 연결된 4점이나 대각선 방향에 있는 점들까지 포함한 8개의 점으로 정의할 수 있는데 본 논문에서는 이웃한 4점을 사용하였다. 2차원 영역확장법을 3차원

으로 확장할 경우 인접한 슬라이스에 대하여 이웃한 점을 확장하는 부분만 추가하여 간단히 구현할 수 있다(그림 2).

```

FUNCTION NAME GrowRegion(x, y, z, Image, Data)
INPUT seed point x, y, z
         Image[Zmax][Ymax][Xmax]
         Data[Zmax][Ymax][Xmax]
         (0 ≤ x < Xmax, 0 ≤ y < Ymax, 0 ≤ z < Zmax, integer value)
OUTPUT Region Information Data
-----
step 1 if Data[z][y][x] is CHECKED then return.
step 2 if Image[z][y][x] is not the region then return.
step 3 Data[z][y][x] := CHECKED.
step 4 if x>0 then GrowRegion(x-1, y, z, Image, Data).
step 5 if y>0 then GrowRegion(x, y-1, z, Image, Data).
step 6 if z>0 then GrowRegion(x, y, z-1, Image, Data).
step 7 if x<Xmax-1 then GrowRegion(x+1, y, z, Image, Data).
step 8 if y<Ymax-1 then GrowRegion(x, y+1, z, Image, Data).
step 9 if z<Zmax-1 then GrowRegion(x, y, z+1, Image, Data).
step 10 STOP
    
```

(그림 2) 점단위 영역 확장 알고리즘

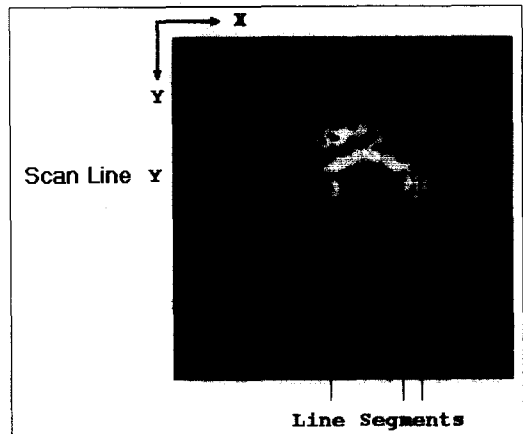
이 영역확장법을 픽셀 단위로 재귀적인 호출을 통해 처리하므로 점단위 영역확장법이라고 할 수 있는데, 이 방법은 각 픽셀에서 이미 영역에 속해 있는 점들에 대해 영역 검사를 하는 중복 비교를 많이 하게 된다. 이 방법을 3차원으로 확장할 경우 중복되는 비교 횟수가 많아져서 수행 속도가 떨어질 수 있다. 또한 분할된 정보는 저장되어야 하는데, 각 픽셀이 원하는 영역에 속하는지를 한 비트로 표현할 경우에 선택되지 않은 영역에도 저장 공간이 할당되어야 하므로 많은 양의 메모리가 필요하다. 메모리 사용량을 줄이기 위해 RLE(Run-Length Encoding)로 압축하여 저장하는 방법 등을 사용할 수 있지만 이 방법은 각 점이 영역에 포함되는지를 찾기 위해서 압축을 해제해야 하는 문제가 있다. 본 시스템에서는 위의 영역확장법이 불필요한 중복 비교가 많은 점과 비효율적인 분할 정보 저장 방법을 가지고 있다는 점에 착안해서, 이를 극복할 수 있는 효율적인 영역확장법을 개발하였는데 이를 위의 점단위 영역확장법과 구분하여 선단위 영역확장법이라 칭한다.

2.3 선단위 영역확장법

2차원 영상에서 한 점은 (x, y)로 표현 할 수 있다.

점단위 영역확장법에서는 이와 같은 점들이 영역을 확장하는데 필요한 판단의 최소 단위가 된다. 그러나 선단위 영역확장법에서는 영역검사를할 때 각각의 점들을 비교하지 않고 선분 단위로 비교한다. 여기서 하나의 선분은 (그림 3)에서와 같이 2차원 영상의 한 주사선(Scan Line) 상에서 연속적으로 같은 영역이라고 판단되는 점들의 좌측 경계값 X_l 과 우측 경계값 X_r 으로 정의할 수 있는데 영역에 포함되는 선분들은 한 주사선상에 여러개 있을 수 있으므로 이러한 한 주사선, Y_i 상에 있는 선분들의 집합을 주사선 집합이라고 정의하면 주사선 집합은

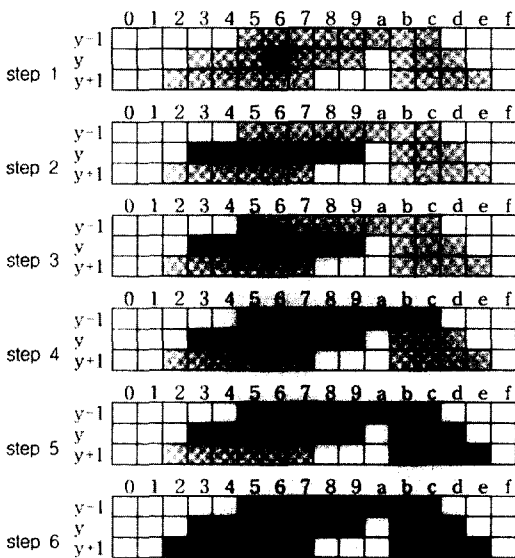
$$\{Y_i, (X_l^1, X_r^1), (X_l^2, X_r^2), \dots, (X_l^n, X_r^n)\}$$



(그림 3) 주사선과 선분

로 표현할 수 있다. 선분 단위 영역확장법은 영역 분할 정보를 이와 같이 선분 단위로 저장함으로써 각각의 점들에 대한 영역 분할 정보를 저장할 때보다 메모리 사용량을 줄이고 영역에 포함되는 각각의 점들을 비교하지 않고 두 경계값 내에 있는 각 점에 대해서는 영역 검사를 하지 않음으로써 계산량을 줄이는 방법이다. (그림 4)는 2차원에서 선분 단위로 영역을 확장해 나가는 과정을 나타낸 그림이다. 그림에서 회색으로 칠해진 부분이 사용자가 보고자 하는 영역이고 검은 부분은 본 알고리즘이 각 단계에서 현재까지 찾은 영역을 나타낸다. 이 그림을 통해 선분 단위로 영역을 확장하는 과정과 비교 횟수 줄이는 과정을 설명하면 다음과 같다.

- step 1 : 사용자가 보고자 하는 영역의 한 점(Seed Point)을 선택한다.
- step 2 : 선택된 점으로부터 좌·우로 연속되는 같은 영역을 찾으면 (y, (3,9))의 결과를 얻는다.
- step 3 : 새로 만들어진 영역 정보인 선분 (3,9)를 이용해 위·아래에 있는 주사선을 검사한다. 먼저 위에 있는 주사선 y-1에서 x좌표 3부터 9까지를 검사한다. x좌표가 3,4인 경우는 영역에 속하지 않으므로 제외시키고 5번에서는 같은 영역이므로 이 점을 시작점으로 삼아 step 1,2의 과정을 반복한다.
- step 4 : 새로운 영역 정보 (y-1, (5,c))를 얻는다. 다음으로 주사선 y와 y-1의 점 6번을 비교해야 하는데 새로 만들어진 영역 정보 (5,c)에서 우측 경계값 c를 이용해서 나머지 비교, 즉 y와 y-1의 x좌표 6부터 9까지의 비교는 생략할 수 있다. 이 부분에서 속도 향상이 이루어지게 된다.
- step 5 : 이 단계에서는 더 이상 위쪽으로 영역 확장 할 수 없으므로 (y-1, (5,c))와 아래 주사선을 검사한다. 이때 주사선 y의 x좌표 5부터 검사를 하는데, 이미 영역(y, (3,9))에 포함되어 있으므로 x좌표 9까지는 검사하지 않고 a부터 검사를 한다. a는 영역에 포함되지 않으므로 제외하고 b를 시작점으로 해서 step 1을 실행한다. 마찬가지로 방법으로 아래쪽으로 영역 분할 계속한다.
- step 6 : 마지막 (y+1, (2,7))까지 영역 분할 수행후 종료한다.



(그림 4) 선단위 영역 확장 과정

2차원 선단위 영역확장법을 3차원으로 확장할 경우 인접한 슬라이스에 대하여 이웃한 주사선의 영역확장 부분을 추가하여 구현할 수 있다. 3차원으로 확장한 선단위 영역분할 알고리즘은 (그림 5)와 같다. 3차원 영역분할시 현재 영상에서는 사용자가 선택한 영역에 포함되어 있지 않아서 다른 영역이라고 판단되지만 실제로는 서로 연결되어 있는 영역이 있을 수 있다. 이와 같은 영역들은 3차원 영역 확장 과정에서 재귀적인 호출을 통해 결국 사용자가 선택한 영역에 포함되게 된다. (그림 6)은 선분 단위 영역 분할법을 사용하여 동일한 슬라이스에 대해 2차원 영역 분할과 3차원 영역 분할 한 결과인데, 2차원 영역 분할 결과 (a)에는 포함되지 않았던 영역이 3차원 영역 분할 결과 (b)에서는 포함되어 있는 것을 볼 수 있다.

FUNCTION NAME *GrowRegion(y, z, Data)*

INPUT y, z

current scan line information Data

OUTPUT all scan line information Data

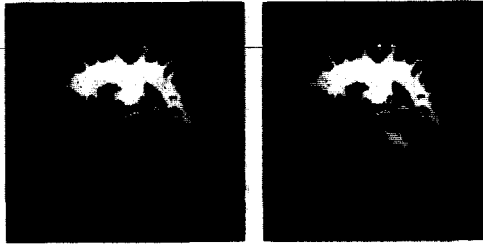
- step 1 *sl := Data.GetScanLine(y, z)*
- step 2 *if sl is empty then return*
- step 3 *if y > 0 and MergeScanLine(y-1, z, sl) = true then GrowRegion(y-1, z, Data).*
- step 5 *if z > 0 and MergeScanLine(y, z-1, sl) = true then GrowRegion(y, z-1, Data).*
- step 6 *if y < Ymax-1 and MergeScanLine(y+1, z, sl) = true then GrowRegion(y+1, z, Data).*
- step 7 *if z < Zmax-1 and MergeScanLine(y, z+1, sl) = true then GrowRegion(y, z+1, Data).*
- step 8 STOP

FUNCTION NAME *GrowRegion(y, z, SL)*

OUTPUT if new region is found then return true

- step 1 *bFlag := false*
- step 2 *LS := SL.getFirstLS()*
- step 3 *while LS is not empty do begin*
 - x := LS.left*
- step 4 *loop*
 - jump := IsChecked(x, y, z)*
 - // acceleration*
 - x := x+jump*
 - if x ≥ LS.right then exit loop*
 - newLS := MakeLineSegment(x, y, z)*
 - if newLS is not empty then*
 - bFlag := true*
 - // acceleration*
 - x := newLS.right*
 - x := x+1*
- end loop*
- LS := SL.getNextLS()*
- end while*
- step 5 *return bFlag*
- step 6 STOP

(그림 5) 선단위 영역 확장 알고리즘

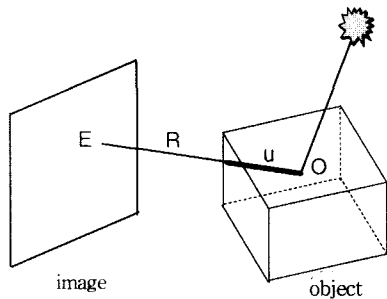


(a) 2차원 영역 분할 (b) 3차원 영역 분할
(그림 6) 영역 분할 수행 결과

3. Levoy의 이산적 광선 추적법

3.1 광선 추적법(ray casting)

광선 추적법(ray casting)이란 빛이 광원에서 출발하여 물체에 도달해서 반사하여 망막에 맺히는 자연현상을 시뮬레이션 하는 방법이다. 실제로 자연현상에서는 망막(이미지)의 한 점에 도달하는 빛은 광원에서 출발한 무수히 많은 수의 빛이 무한히 많은 물체 상의 점에 도달하여 각 점의 특성에 따라 반사하여 각기 다른 방향으로부터 도달하여 합하여지므로 이 현상을 정확히 시뮬레이션 하는 것은 불가능하다. 따라서 더 단순한 모델이 필요한데 보통 광선 추적법에서는 이미지 상의 각 점(픽셀, pixel)에 들어오는 빛 중 단 한 방향의 빛만을 고려한다. 그리고 모든 점에 대해 이미지 평면 상에 도달하는 빛은 평행하며 이미지 평면과 수직을 이룬다고 가정한다(orthographic parallel projection). 즉, 이미지 상의 각 픽셀의 색을 결정할 때 이러한 조건을 만족하는 빛만을 고려한다. 광원에서 나온 빛은 이 가상의 광선 상에 있는 물질의 한 점에 도달하여 그 점에서의 물질의 성질에 따라 반사한다. 이 중 픽셀 E 방

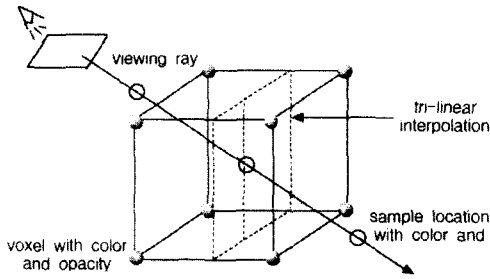


(그림 7) 색상의 계산

향으로 반사된 빛만이 픽셀 E의 색을 결정하는데 영향을 미치는데 다시 이 점에서 E 방향으로 반사된 빛은 점 O와 E 사이에 있는 반투명 물질을 통과하면서 강도가 변하여 E에 맺혀진다. 광선 R 상에 있는 임의의 점에서 이 값을 계산하여 모든 R 상의 점 O에 대하여 합함으로써 E의 색이 결정된다(그림 7).

3.2 불투명도의 정의

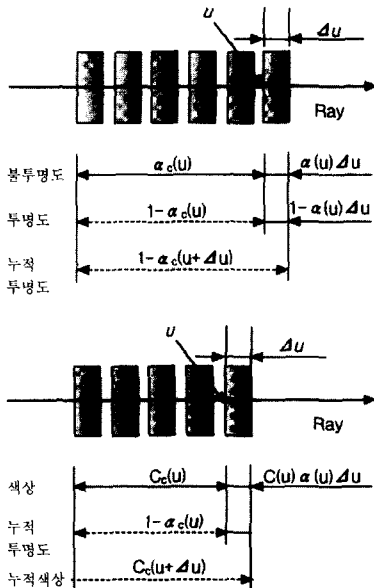
위와 같이 물질의 한 점이 픽셀의 색에 미치는 영향을 계산할 때 그 점과 픽셀 사이의 누적 투명도를 결정하여야만 하고 이를 위하여 각 체적 원소에 불투명도를 할당해 주어야 한다. 또한 이 문제는 사용자가 보고자 하는 것을 선택하기 위하여 체적 원소의 불투명도를 조절해야 하므로 매우 중요한 문제이다. 어떤 물질의 불투명도는 그 물질로 들어온 빛이 그 물질을 통과함에 따라 얼마나 약해지는가와 관계가 있다. 불투명도의 크기는 빛이 얼마만큼의 물질을 통과하느냐, 즉 물질의 “길이”에 따라 결정된다. 따라서 어떤 물질의 “단위 길이 불투명도”를 정의할 수 있다. 주어진 한 점에서 광선의 진행 방향으로 단위 길이의 물질이 채워져 있다고 할 때 그 점에서의 단위 길이 불투명도는 빛이 그 단위 길이의 물질을 통과할 때 약화되는 비율로 정의된다. 사용자가 특정한 밀도를 가지는 체적 원소만을 보기를 원할 때 원하는 물질의 표면을 그 주변 물질을 배경 삼아 보여주기 위해서는 특정한 밀도를 가지는 체적 원소에는 불투명도를 높게 할당하고 그 밀도와 차이가 큰 밀도를 가지는 체적 원소에는 낮은 불투명도를 할당하며(즉 투명하다고 가정함) 그 주어진 밀도와 비슷한 밀도를 가지는 체적 원소에 대하여서는 그 중간 값을 할당하는 방법을 사용한다. Levoy는 다음 세 가지의 기준에 의거하여 중간 불투명도를 할당하였다. 첫째, 체적 원소의 밀도가 주어진 밀도 값에서 멀어짐에 비례하여 그 체적 원소의 불투명도가 작아진다. 둘째, 체적 원소의 그래디언트(gradient)의 절대값에 반비례하여 체적 원소의 불투명도가 작아진다. 즉 체적 원소의 밀도 변화가 클수록 불투명도는 가파르게 떨어진다. 셋째, 원하는 표면 주위의 보고자 하는 배경의 두께를 사용자가 정하여 그 두께보다 멀리 떨어져 있는 체적 원소는 투명하게 한다. 그리고 체적 원소가 아닌 다른 점에서의 불투명도는 그 점을 에워싸고 있는 여덟 개의 체적 원소들의 불투명도로부터 3차 선형 근사식으로(trilinear interpolation) 결정한다(그림 8).



(그림 8) 셀을 지나는 광선

3.3 누적 불투명도와 색상의 계산

한 픽셀에서 발사된 광선은 직선이므로 임의의 변수 u 로 매개될 수 있다. 즉 광선 상의 임의의 점은 그 픽셀과 그 점과의 거리 u 의 함수로 표시할 수 있다. 따라서 매개변수 u 는 광선 상의 한 점을 나타낸다. $\alpha_c(u)$ 를 픽셀과 광선 상의 점 u 사이의 누적 불투명도라 하고 Δu 를 체적 원소의 불투명도를 정의하는데 사용된 단위 길이라고 하면 픽셀에서 점 u 까지의 투명도는 $1 - \alpha_c(u)$ 로 나타내어진다. 그때 픽셀에서 점 $u + \Delta u$ 까지의 투명도 $1 - \alpha_c(u + \Delta u)$ 는 u 와 $u + \Delta u$ 사이의 투명도인 $1 - \alpha(u)\Delta u$ 와 u 까지의 투명도 $1 - \alpha_c(u)$ 의 곱으로 나타내진다(그림 9).



(그림 9) 색상과 누적 불투명도

즉 $1 - \alpha_c(u + \Delta u) = (1 - \alpha(u)\Delta u)(1 - \alpha_c(u))$ 로 나타내지며 다음과 같은 관계를 얻을 수 있다.

$$1 - \alpha_c(u + \Delta u) = (1 - \alpha_c(u))(1 - \alpha(u)\Delta u) \quad (1)$$

$$\alpha_c(u + \Delta u) = \alpha_c(u) + \alpha(u)(1 - \alpha_c(u))\Delta u$$

마찬가지로 누적 색깔 $C_c(u)$ 는 Phong Lighting model을 사용하여 다음과 같이 계산될 수 있다. $C(u)$ 는 점 u 에서의 픽셀 방향으로의 색이고 점 u 를 포함하는 단위 길이 물질의 단위길이당 불투명도가 $\alpha(u)$ 이므로 점 u 에서 실제로 픽셀 방향으로 반사되는 색의 강도는 $C(u)\alpha(u)\Delta u$ 가 된다. 이 빛은 u 와 픽셀 사이를 통과하면서 약화되므로 다음과 같은 관계가 성립한다[9].

$$C_c(u + \Delta u) = C_c(u) + (1 - \alpha_c(u))C(u)\alpha(u)\Delta u \quad (2)$$

Levoy[4]의 이산적인 방법은 이 Δu 를 1로 한 결과이다.

4. 분석적 광선 추적법

4.1 분석적 방법을 이용한 모델링

누적 불투명도와 색상을 계산하는 미분 방정식을 유도하는 방법은 여러 가지가 있다[1, 2, 6, 7, 8, 10, 12]. 본 논문에서는, Levoy[4]가 제안한 이산적인 방법으로부터 미분 방정식을 유도했다. 위에서 구한 식 (1), (2)에서 단위 길이 Δu 를 무한하게 작게 잡으면 (1), (2)는 지금까지의 이산적 관계가 아닌 연속적 관계가 된다. 하지만 이때는 “단위 길이 불투명도” $\alpha(u)$ 와는 다른 새로운 불투명도 밀도 함수가 정의 되어야 한다. 주어진 한 점 u 를 포함하는 작은 주변에서의 불투명도 밀도 함수 $\beta(s)$ 는 빛이 점 s 를 통과하면서 흡수되는 정도를 말한다. 이는 미분 불투명도(differential opacity)라 불리기도 하는데[12], $\alpha(u)$ 와 $\beta(s)$ 와의 관계는 4.3절에 설명될 것이다. Δu 를 무한히 작게 수렴시키면 식 (1), (2)로부터 다음 관계를 얻는다

$$\lim_{\Delta u \rightarrow 0} \left[\frac{\alpha_c(u + \Delta u) - \alpha_c(u)}{\Delta u} \right] = \alpha(u)(1 - \alpha_c(u))$$

$$\lim_{\Delta u \rightarrow 0} \left[\frac{C_c(u + \Delta u) - C_c(u)}{\Delta u} \right] = (1 - \alpha(u))C(u)\alpha(u)$$

위의 식으로부터 다음과 같은 연속적인 공식을 얻는다.

$$da_c(u)/du = \beta(u)(1 - a_c(u)) \quad (3)$$

$$dC_c(u)/du = (1 - a_c(u))C(u)\beta(u) \quad (4)$$

(3)의 미분방정식을 경계 조건 $a_c(0) = 0$ 으로 풀면

$$a_c(u) = 1 - e^{-\int_0^u \beta(s)ds} \quad (5)$$

을 얻는다. 마찬가지로 (6)의 미분방정식의 해를 구하면 다음과 같은 식을 얻는다.

$$C_c(u) = \int_0^u C(u)\beta(u)(1 - a_c(u)) du \quad (6)$$

식 (6)에서 $1 - a_c(u) = e^{-\int_0^u \beta(s)ds}$ 이므로 다음을 구할 수 있다.

$$C_c(u) = \int_0^u C(u)\beta(u) e^{-\int_0^u \beta(s)ds} du \quad (7)$$

식 (7)에서 $C(u)$ 과 $\beta(u)$ 는 u 를 에워싸고 있는 여덟 개의 체적 원소들의 불투명도로부터 3차 선형근사식으로(trilinear interpolation) 결정한다. 따라서 $C(u)$ 와 $\beta(u)$ 는 u 의 3차 다항식이 된다. 위의 부정 적분은 우리에게 알려져 있는 함수로 나타내 지지 않으므로 본 논문에서는 $e^{-\int_0^u \beta(s)ds}$ 를 다항식으로 근사시켰다. 이 경우 $C(u)\beta(u) e^{-\int_0^u \beta(s)ds}$ 는 다항식이 되므로 적분이 가능하게 된다.

4.2 분석적 해법

픽셀에서 발사된 광선은 많은 셀을 통과하는데 $\beta(u)$ 는 점 u 가 속해 있는 셀을 둘러싸고 있는 여덟개 체적 요소의 β 값에 의하여 3차 선형근사된다. 따라서 우리는 광선을 분할하여 선분들인 $[0, u_0], [u_0, u_1], \dots, [u_{n-1}, u_n]$ 을 정의할 수 있는데, 여기서 각 선분 $[u_{i-1}, u_i]$ 는 한 셀에 속한다. 그러면 $\int_0^u \beta(s)ds = \sum_{i=1}^n \int_{u_{i-1}}^{u_i} \beta(s)ds$ 로 나타낼 수 있다. 이때 $\beta(s)$ 는 s 가 u_{i-1} 에서 u_i 사이에 변할 때 오직 한 셀에서만 정의된다. β 는 그 셀에서 3차 다항식으로 나타내어지므로 부정 적분이 존재하게 되고, 그 부정 적분은 모든 셀에 대하여 일정한 형태를 갖는다. 따라서 미리 계산된 부정 적분의 계수를 이용하여 쉽게 계산될 수 있다. 즉, β_{li} ($i=1, 2, \dots, 8$)을 1 번째 셀에 인접한 8개 voxel

의 미분 불투명도라고 하고, $\beta_i(x,y,z)$ 을 1 번째 셀내의 미분 불투명도라 하면, $\beta(x,y,z)$ 를 원점으로 이동시킨 $\beta_{Orr}(x,y,z)$ 은 다음과 같이 표현된다.

$$\begin{aligned} &(1-x)(1-y)(1-z)\beta_{l,1} + x(1-y)(1-z)\beta_{l,2} + \\ &(1-x)y(1-z)\beta_{l,3} + xy(1-z)\beta_{l,4} + \\ &(1-x)(1-y)z\beta_{l,5} + x(1-y)z\beta_{l,6} + \\ &(1-x)yz\beta_{l,7} + xyz\beta_{l,8} \end{aligned}$$

따라서, $\int_{u_{i-1}}^{u_i} \beta(s)ds$ 는 β_{li} 들의 합으로 표시될 수 있다.

즉,

$$\int_{u_{i-1}}^{u_i} \beta(s)ds = \int_0^{u_i - u_{i-1}} \beta_{Orr}(s)ds = \sum_{i=1}^8 B_i \beta_{l,i}$$

이다. 여기서 광선 γ , $(x(u), y(u), z(u))$ 는 $(p_1 + uq_1, p_2 + uq_2, p_3 + uq_3)$ 로 표현되고 B_i 는 p_i, q_i , 그리고 $u_i - u_{i-1}$ 의 함수로 표현할 수 있다. 따라서 한 광선 γ 에 대해서 q_i' 가 고정되고 B_i 는 p_i' 와 $u_i - u_{i-1}$ 를 이용하여 쉽게 계산할 수 있다. 한 픽셀에서 발사된 광선 γ 가 통과하는 투명하지 않은 마지막 점을 u_n 이라 할 때, 그 픽셀의 밝기는

$$C_c(u_n) = \int_0^{u_n} C(u)\beta(u) e^{-\int_0^u \beta(s)ds} du$$

이다. 일반적으로 $C_c(u_n)$ 에 대한 닫힌 형태의 해를 구할 수 없으므로, 우리는 다음과 같은 방법으로 근사적인 해를 구하였다. 우선 다음과 같이 부분 적분을 이용하여 식을 간단히 하였다.

$$\begin{aligned} C_c(u_n) &= -C(u_n) e^{-\int_0^{u_n} \beta(s)ds} + C(u_0) \\ &+ \int_{u_0}^{u_n} C'(u) e^{-\int_0^u \beta(s)ds} du \end{aligned}$$

그리고 $C(u)$ 와 $\beta(u)$ 는 광선이 지나는 복셀에 따라서 다르게 정의되므로 이를 나누어 쓰면 다음의 결과를 얻는다.

$$\begin{aligned} &\int_{u_0}^{u_n} C'(u) e^{-\int_0^u \beta(s)ds} du \\ &= \sum_{i=1}^n \int_{u_{i-1}}^{u_i} C'(u) e^{-\int_0^u \beta(s)ds} du \\ &= \sum_{i=1}^n \int_{u_{i-1}}^{u_i} C'(u) e^{-\sum_{j=1}^{i-1} \int_{u_{j-1}}^{u_j} \beta_{l,j}(s)ds - \int_{u_{i-1}}^{u_i} \beta_{l,i}(s)ds} du \\ &= \sum_{i=1}^n e^{-\sum_{j=1}^{i-1} \int_{u_{j-1}}^{u_j} \beta_{l,j}(s)ds} \int_{u_{i-1}}^{u_i} C'(u) e^{-\int_{u_{i-1}}^u \beta_{l,i}(s)ds} du \end{aligned}$$

여기서 $e^{-\int_u^{u_1} \beta(s) ds}$ 를 Taylor expansion으로 $(u_{l-1} + u)/2$ 주위에서 근사시키면 $\int_{u_{l-1}}^u C'(u) e^{-\int_u^{u_1} \beta(s) ds} du$ 의 값을 구할 수 있다.

4.3 미분 불투명도의 계산

누적 불투명도와 색상을 계산하기 위한 연속 방정식을 사용하려면 주어진 점에서의 미분 불투명도가 필요하다. 그러나 각 점에서의 미분 불투명도 $\beta(u)$ 를 명확하게 지정하기가 쉽지 않다. 본 논문에서는 미분 불투명도 $\beta(u)$ 를 단위 길이 불투명도 $\alpha(u)$ 로부터 식 (1)를 이용해서 구한다. $\alpha(u)$ 는 0에서부터 u 까지의 누적 불투명도이고, $\alpha_c(0; u)$ 로 표시할 수 있다. 0이 아닌 지점으로부터 시작하는 점을 선택하면 다음과 같은 식을 얻을 수 있다.

$$\alpha_c(u-0.5; u+0.5) = 1 - e^{-\int_{u-0.5}^{u+0.5} \beta(s) ds}$$

여기서 1은 단위 길이를 의미한다. 위에서 정의한 단위 길이 불투명도 $\alpha(u)$ 에 의해

$$\alpha_c(u-0.5, u+0.5) = \alpha(u)$$

라고 할 수 있다. 따라서 $\alpha(u) = 1 - e^{-\int_{u-0.5}^{u+0.5} \beta(s) ds}$ 라고 할 수 있다. 주어진 어떠한 단위 길이 불투명도 $\alpha(u)$ 에 대해서도 위에 식을 만족하는 수많은 미분 불투명도 분포 함수 $\beta(u)$ 가 존재한다. 사실 $\alpha(u)$ 자체로는 $\beta(u)$ 에 대한 어떤 정보도 주지 못한다. 그래서 $\beta(u)$ 가 간격 $[u-0.5; u+0.5]$ 의 간격 내에서는 상수 β 와 같다고 생각할 수 있다. 그렇다면

$$\alpha(u) = 1 - e^{-\int_{u-0.5}^{u+0.5} \beta(s) ds} = 1 - e^{-\beta(s)}, \quad s \in [u-0.5, u+0.5]$$

로 표현할 수 있고

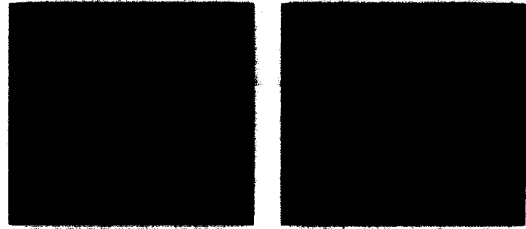
$$\beta(s) = \log \frac{1}{1 - \alpha(u)}, \quad s \in [u-0.5, u+0.5]$$

를 구할 수 있다. 따라서 점 u 에서 주어진 단위 길이 불투명도 $\alpha(u)$ 를 알면 $[u-0.5, u+0.5]$ 에서 정의되는 미분 불투명도 $\beta(s)$ 를 구할 수 있다.

5. 실험 결과

본 논문에서 제안된 방법을 Levoy의 방법과 함께 구

현하여 비교하였다. 구현 환경은 펜티엄 200 MHz CPU와 64M RAM을 사용하는 Windows 95 환경이고 실험에 사용한 데이터는 사람 머리를 MRI로 촬영한 $256 \times 256 \times 109 \times 8\text{bit}$ 크기의 자료이다. 본 논문에서는 뇌의



(a) 체적 원소당 5개의 표본 (b) 체적 원소당 2개의 표본

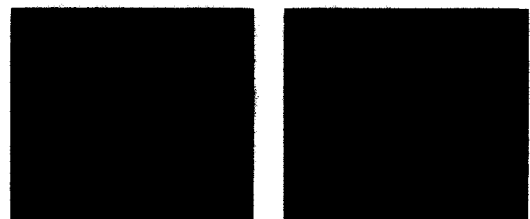


(c) 체적 원소당 1개의 표본 (d) 분석적 방법

(그림 10) 안쪽 뇌



(a) 체적 원소당 5개의 표본 (b) 체적 원소당 2개의 표본



(c) 체적 원소당 1개의 표본 (d) 분석적 방법

(그림 11) 대뇌 피질

안쪽 부분과, 대너 피질 부분을 2절의 방법으로 영역 분할한 후 렌더링을 하였으며, Levoy의 방법에서는 표본 추출 횟수를 다양하게 하여 결과를 얻었다. (그림 10), (그림 11)의 (a), (b), (c)는 Levoy의 방법을 사용해서 렌더링한 결과이며 샘플링은 각각 5개, 2개, 1개이다. 그리고 (d)는 본 논문에서 제시한 분석적 방법으로 렌더링한 결과이다. Levoy가 주장했듯이[5], 체적 원소 길이당 3개 이상의 표본 추출은 이미지에 별로 커다란 향상을 가져오지 못하였다. Levoy의 방법과 비교해서 분석적인 방법의 속도가 느렸으나 우리가 관심이 있는 것은 시간이 아니고 결과의 질이다. 이 결과에서 볼 수 있듯이 Levoy의 이산적인 방법보다는 분석적인 방법을 사용한 렌더링 이미지가 명암을 더 잘 표현한다.

6. 결 론

본 논문에서는 체적 시각화 과정을 이론적으로 고찰하여 시각화 모델을 제시하고 그 모델로 유도된 관계로부터 미분 방정식을 이용하여 분석적 체적 시각화 해법을 구하였다. 그리고 영역확장법을 기반으로 한 효율적인 3차원 영역 분할 개발하여 위의 분석적인 방법을 이용하여 3차원 체적 데이터의 시각화를 위한 시스템을 구현하였다. Levoy의 이산적인 방법과 분석적인 방법을 같은 데이터에 대해 3차원 영역 분할 수행 후 적용하여 비교하였다. 실험에서는 분석적인 방법이 이산적인 방법에 비해 더 좋은 이미지를 생성했다. 그 이유는 Levoy의 방법에서는 유한 개의 추출 점들에 대해 이산적인 컬러 값들의 함으로 구한 반면 분석적인 방법은 이산적인 방법에서 무한개의 추출 점을 사용한 것과 동일한 효과를 갖기 때문이라 생각된다. Levoy는 표본 추출 빈도에 주목하여 체적 원소 길이당 3개 이상의 샘플링은 이미지에 별로 커다란 향상을 가져오지 못한다고 주장하였고 우리의 실험도 같은 결론을 얻었으나 이것은 어디까지나 잠정적인 실험 결과일 뿐, 그래야 할 이유가 밝혀진 것은 아니다. 직관적으로는 더욱 많은 표본을 추출하면 더 좋은 결과가 나올 것이라고 추정하는 것이 당연하다. 그러나 이산적인 방법에서 무한개의 표본을 선택하는 것은 불가능하기 때문에 본 논문에서 제안하는 분석적인 방법이 필요하다고 할 수 있다. 또한 실험에 사용된 3차원 체적 데이터가 연속적이지 않다는 사실에 주목할 필요가 있

다. 이산적인 체적 데이터에는 이산적인 방법만으로도 충분하기 때문이다. 이는 연속적인 체적 데이터에서는 분석적인 방법이 더 적합하다는 주장이 설득력을 갖게 하지만 위의 주장들은 더 깊은 이론적 고찰을 필요로 하고 이는 향후의 연구 과제이다.

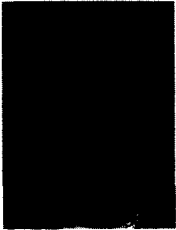
참 고 문 헌

- [1] James Kajiya and Brian P. Von Herzen, "Ray Tracing Volume Densities," *Computer graphics*, Vol.10, No.3, pp.165-174, July 1984.
- [2] "The Rendering Equation," *Computer Graphics*, Vol.20, No.4, pp.143-150, August 1986.
- [3] Marc Levoy, "Display of surfaces from volume data," *IEEE Computer Graphics and Applications*, 8(3):29-37, March 1988.
- [4] Marc Levoy, "Efficient Ray Tracing of Volume Data," *ACM Transactions on Graphics*, Vol.9 No.3, pp.245-261, July 1990.
- [5] Marc Levoy, "Ray Tracing of Volume Data," *Course Notes*, SIGGRAPH 1991.
- [6] Nelson Max, "Light Diffusion through Clouds and Haze," *Computer Vision, Graphics, and Image Processing*, 33:280-292, 1986.
- [7] Nelson Max, "Optical Models for Direct Volume Rendering," *IEEE Transactions on Visualization and Computer Graphics*, Vol.1, No.2, pp.99-108, June 1995.
- [8] Tomoyuki Nishita, Yasuhiro Miyajaki, and Eihachiro Nakame, "A Shading Model for Atmospheric Scattering considering Luminous Intensity Distribution of Light Sources," *Computer Graphics*, Vol.21, No.4, pp.303-310, July 1987.
- [9] Thomas Porter and Tom Duff, "Compositing Digital Images" *Computer Graphics*, 18(3):253-260, July 1984.
- [10] Paolo Sabella, "A Rendering Algorithm for Visualizing 3D Scalar Fields," *Computer Graphics*, 22(4): 51-58, July 1988.
- [11] Craig Upson and Michael Keeler, "The V-buffer: Visible Volume Rendering," *Computer Graphics*, 22(4):L59-64, July 1988.
- [12] Jane Wilhelms and Allen V. Gelder, "A Coherent Projection Approach for Direct Volume Rendering," *Computer Graphics*, Vol.15, No.4, pp.275-284, July 1991.



박 현 우

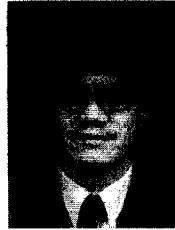
e-mail : gusn@multi.soongsil.ac.kr
1997년 한국외국어대학교 물리학과
졸업(학사)
2000년 숭실대학교 컴퓨터공학과
졸업(석사)



백 두 원

e-mail : dpaik@computing.soongsil.ac.kr
1983년 서울대 수학과 졸업(학사)
1990년 미국 University of Min-
nesota 전산학과 졸업(석사)
1991년 미국 University of Min-
nesota 전산학과 졸업(박사)

1992년~1994년 AT&T Bell 연구소 연구원
1995년~현재 숭실대학교 컴퓨터학부 조교수
관심분야 : 컴퓨터 그래픽스, 애니메이션, 알고리즘, CAD



정 문 렬

e-mail : moon@family.sogang.ac.kr
1980년 서울대 계산통계학과 졸업
(학사)
1982년 한국과학기술원 전산학과
졸업(석사)

1992년 미국 Univ. of Pennsylvania 전산학과 졸업(박사)
1982년~1986년 국방과학연구소 연구원
1992년~1994년 일본구주공업대학 조교수
1994년~1999년 숭실대학교 컴퓨터학부 부교수
2000년~현재 서강대학교 영상대학원 디지털영상학과
부교수

관심분야 : 컴퓨터 그래픽 애니메이션, 인간과 컴퓨터
상호작용, 가상현실