Journal of Korea Multimedia Society Vol. 14, No. 3, March 2011(pp. 339-347)

# 카디널 보간을 이용한 효율적인 고화질 볼륨 가시화

계 희 원[†]


## 요 약

볼륨 가시화가 의료 데이터에 이용되면서 고화질 영상을 생성하고자 하는 요구가 계속되고 있다. 기존에 볼륨 가시화에 사용되던 선형 보간 필터는 상대적으로 빠른 속도로 좋은 화질의 영상을 생성하지만, 더 높은 화질의 영상을 생성하려면 고차 보간 필터가 필요하다. 고차 보간 필터를 사용하는 경우 성능 저하가 발생하는데, 재-샘플링의 연산 시간이 크게 증가하기 때문이다. 본 연구는 고차 보간 필터에 카디널 보간을 적용하여 고화질 볼륨 가시화를 수행한다. 그리고 빈공간 도약 기법을 적용 가능하게 하여 재-샘플링의 횟수를 감소시키고 효율적인 가시화를 수행한다. 구체적으로 볼륨 데이터를 일정 크기의 블록으로 나누고, 각 블록의 밀도값의 상계와 하계를 이용하여, 빈공간을 도약한다. 이 과정에서 본 연구는 각 블록의 밀도값의 상계와 하계를 계산하는 새로운 방법을 제안한다. 그 결과로서 빈공간 도약이 효율적으로 수행되어 고화질 볼륨 가시화 수행속도를 크게 향상되었다.


# Efficient High Quality Volume Visualization Using Cardinal Interpolation

Heewon Kye[†]


## ABSTRACT

As the volume visualization has been applied to render medical datasets, there has been a requirement to produce high quality images. Even though nice images can be generated by using previous linear filter, high order filter is required for better images. However, it takes much time for high order resampling, so that, overall rendering time is increased. In this paper, we perform high quality volume visualization using the cardinal interpolation. By enabling the empty space leaping which reduces the number of resampling, we achieve the efficient visualization. In detail, we divide the volume data into small blocks and leap empty blocks by referring the upper and lower bound value for each block. We propose a new method to estimate upper and lower bound value of for each block. As the result, we noticeably accelerated high quality volume visualization.


**Key words:** Direct Volume Rendering(직접볼륨가시화), Efficient Ray Casting(효율적인 광선추적법), Cardinal Spline Interpolation(카디널 보간)

## 1. 서 론

볼륨 가시화(volume visualization)는 삼차원 볼륨 데이터(volume data)를 읽어 영상을 생성하는 가시화 기법으로 과학적 가시화 또는 의료영상 시스템 등에 폭넓게 사용되는 방법이다. 볼륨 데이터는 일반

※ 교신저자(Corresponding Author) : 계희원, 주소 : 서울시 성북구 삼선동3가 389 한성대학교 연구관 911호(136-792), 전화 : 02)760-8014, FAX : 02)760-4347, E-mail : kuei@hansung.ac.kr
접수일 : 2011년 1월 20일, 수정일 : 2011년 2월 20일
완료일 : 2011년 3월 15일

[†] 정회원, 한성대학교 정보시스템공학과
※ 본 연구는 한성대학교 교내연구비 지원과제임 또한 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2010-0015641).

적으로 복셀(voxel)이라고 부르는 스칼라(sclar) 값의 삼차원 배열 형태로 구성되어 있으며[1], 배열의 크기가 매우 크기 때문에 볼륨 데이터는 수백MB에서 수GB에 이를 정도로 많은 메모리를 차지하게 된다.

볼륨 데이터를 가시화하려면, 빠른 시간에, 대량의 데이터를 처리하여, 고품질의 영상을 생성해야 한다. 그 때문에 가시화의 가속화 방법에 많은 연구가 이루어졌다[2-4]. 최근 고속의 하드웨어가 등장[5]하여 일반적인 크기의 범용 데이터를 가시화하는 것은 어렵지 않은 일이 되었으나, 이에 따라 더 높은 품질의 영상에 대한 요구가 발생하여 가속화 방법은 여전히 중요한 문제로 인식된다.

볼륨 가시화의 대표적인 방법인 광선 투사법(ray casting)[6]은 다음과 같은 과정을 통해 수행된다(그림 1). 영상을 구성하는 각 화소(pixel)에서 광선을 발사하여 육면체 형태의 볼륨 데이터를 관통하는 선분 영역을 구한다. 선분 영역에서 볼륨 데이터의 변화하는 밀도값(density value)을, 사용자가 정의한 함수를 통해 광학 성분(optical properties)으로 변환하고, 이를 적분하여 화소의 색상으로 결정한다. 이때 적분을 구하기 위해 근사적으로 수치해법을 이용하며, 광선의 일정 위치마다 이산적으로 밀도값을 구하는 방법을 사용한다.

이렇게 일정 위치에서 밀도값을 구하는 과정을 재-샘플링(re-sampling)이라고 하며, 주변 격자점에 위치한 복셀값을 가중 평균하여 계산하게 된다. 일반적으로 선형 보간(linear interpolation)을 이용하면 우수한 화질의 영상을 생성한다. 한편 더 높은 품질의 영상을 생성하기 위해 삼차 보간(cubic interpolation)을 사용하는 연구[7,8]가 이루어지고 있다.

삼차 보간은 선형 보간에 비해 연산량이 대폭 증가한다. 선형 보간은 두 개의 점에 대한 가중합을 계산하며, 이차원 영상 데이터에 적용하면 한 번의 재-샘플링을 수행하기 위해 3(=2+1)회, 볼륨 데이터에 적용되면 그림 2(a)와 같이 7(=4+2+1)회의 선형 보간을 하게 된다. 한편 삼차 보간은 네 개의 점에 대한 가중합을 수행하며, 이차원 영상 데이터에 적용하면 한 번의 재-샘플링을 수행하기 위해 5(=4+1)회, 볼륨 데이터에 적용되면 그림 2(b)와 같이 21(=16+4+1)회의 삼차 보간을 해야 한다. 재-샘플링에 삼차 보간을 이용하면, 보간 횟수와 각 보간에 필요한 시간이 모두 크게 증가하므로 전체적인 가시화 시간이 크게 증가하게 된다. 따라서 가속화 알고리즘을 적용하는 것이 매우 중요하다.

볼륨 가시화의 대표적인 가속화 알고리즘은 빈공간 도약(empty space leaping)과 조기 광선종료(early ray termination)이다. 빈공간 도약 방법은 전처리(pre-processing)를 통해 볼륨 데이터 내부의 투명하다고 생각되는 지역의 위치 정보를 나무(tree) 등의 자료구조로 저장하여 놓고, 가시화 단계에서 투명한 지역의 재-샘플링을 생략하여 속도를 향상하는 방법이다[2]. 조기 광선종료는 하나의 광선에 대한 색상을, 재-샘플링을 반복하여 누적시켜 나가다가, 화소의 누적된 투명도가 불투명하다고 판단되면 더 이상의 재-샘플링을 중단하는 방법이다[2]. 그 이외에 하드웨어의 특성을 고려하여, 메모리의 참조 방법을 변경하거나[3,9] 자주 사용하는 메모리를 별도로 관리하여 캐시 효율을 향상시키거나[4], 그래픽스 하드웨어[5] 또는 볼륨 가시화 전용 하드웨어[10]를 사

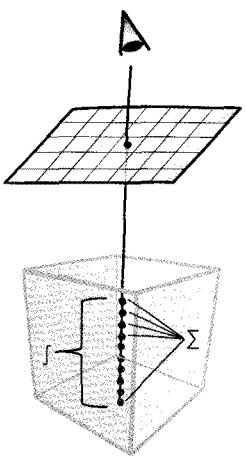



그림 1. 광선 투사법의 개요. 일정간격 위치마다 밀도값을 추정함

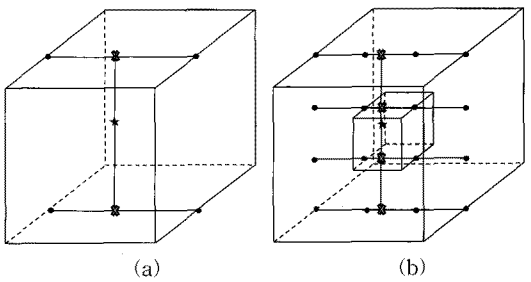



그림 2. 재-샘플링을 수행하는데 필요한 보간의 비교. (a) 선형 보간 (b) 삼차 보간

용할 수 있다.

본 연구에서는 빈공간 도약을 삼차 보간을 적용할 때 발생하는 문제점을 지적하고 이를 해결하는 방법을 제시한다. 이후 논문의 순서는 다음과 같다. 먼저 2장에서 삼차 보간을 위한 빈공간 도약 알고리즘을 설명하고, 3장에서 실험 결과를 보이며, 4장에서 결론을 맺는다.

## 2. 삼차 보간을 위한 빈공간 도약 알고리즘

### 2.1 빈공간 도약을 통한 가속화된 볼륨가시화

볼륨 데이터의 일부분이 투명한지 불투명한지의 여부는 사용자가 지정하는 불투명도 전이함수(opacity transfer function)에 따라 결정된다[6]. 사용자는 가능한 밀도값의 범위에 대해 투명도를 정의하는데, 예를 들어 인체의 골격계를 관찰하고 싶다면, 근육과 피하지방에 해당하는 밀도값에 불투명도 0을 지정한다. 이를 그래프로 표현하면, x축을 밀도값, y축을 불투명도로 표현하여 나타날 수 있다. 이후 가시화 과정에서 재-샘플링을 통해 획득한 밀도값에 불투명도를 적용하면 골격부분만 영상으로 출력되는 식이다.

한편, 볼륨 데이터를 균일한 크기의 블록으로 나누고, 전처리 단계에서 각 블록에 포함된 모든 밀도 범위에 대한 최댓값과 최솟값을 계산할 수 있다. 만약, 블록의 [최솟값, 최댓값] 영역이 불투명도 전이함수에서 모두 0 (투명함)으로 판단되면, 블록의 어느 위치에서 재-샘플링을 수행하여도 투명한 것이 자명하므로, 투명한 블록을 가시화에서 건너뛰어 성능을 향상시킬 수 있는데, 이를 빈공간 도약 기법이라고 한다. 예를 들면, 그림 3에서 블록 [$m_1$, $M_1$]은 투명하게 판단되어 생략되며, [$m_2$, $M_2$]같은 경우 불투명도 전이함수와 겹치는 영역이 있어 불투명한 것으로 판단한다.

한편, 블록의 최댓값을 계산하는 과정에서 오차가 있어, 블록의 최댓값이 그림 4(a)와 같이 실제 최댓값보다 작게 측정되었다고 가정하면, 불투명한 블록들의 일부가 투명하게 오인 되어 부당하게 가시화에서 제외될 수 있다. 이 경우는 화질 손상이 일어나게 되므로 발생해서는 안 된다. 반대로, 계산한 최댓값이 그림 4(b)와 같이 실제 최댓값보다 크게 측정되었다고 가정하면, 투명한 블록들의 일부가 불투명한 블록

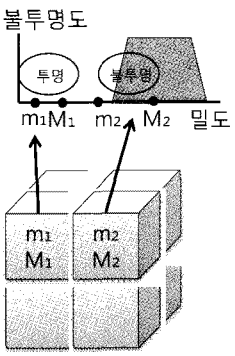



그림 3. 볼륨 데이터를 구성하는 블록의 투명도 검사법

으로 오인되어 가시화를 건너뛰지 못하고 재-샘플링을 수행한다. 재-샘플링 결과는 결국 투명하게 판단되어 화질 손상은 없는 대신 속도 향상 정도가 일부 줄어들게 된다. 따라서 실제 최댓값을 정확하게 구하기가 매우 어려운 경우, 실제 최댓값보다 약간 큰 근사값을 구할 수 있다면 이를 이용해 빈공간 도약을 수행할 수 있다.

즉, 빈공간 도약을 수행하려면 꼭 블록의 최댓값을 구할 필요는 없고 적당한 상계(upper bound)를 구하면 되나, 빈공간 도약의 빈도를 높여 성능을 향상시키기 위해 최소상계(least upper bound)인 상한(supremum) 또는 최댓값에 근접한 값을 구하는 것이 바람직하다. 이는 최솟값에 대해서도 마찬가지로, 블록의 최솟값에 가까운 하계 (lower bound)를 구할 수 있다면 빈공간 도약을 수행할 수 있다. 블록의 최댓값 또는 상계를 구하는 과정은 최솟값 또는 하계를 구하는 과정과 거의 동일하므로, 이후 블록의 상계를 구하는 과정에 대해서만 주로 다루도록 한다.

### 2.2 삼차 보간에서 블록의 상계 획득 문제

먼저, 본 연구에서는 삼차 보간을 함수로서, 일반적으로 많이 사용하는 카디널 곡선(cardinal spline)

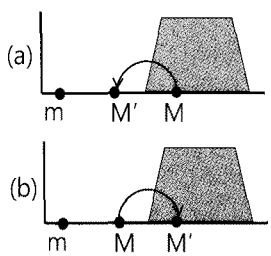



그림 4. 블록의 최댓값 계산에 오차가 있는 경우의 처리방법

[11]을 사용하였다. 카디널 곡선은 제어점을 모두 지나고 부드러운 형태의 보간을 생성하는 장점이 있다. 카디널 곡선은 두 점 $P_i$, $P_{i+1}$ 사이의 구간을 보간하기 위해 네 점 $P_{i-1}$, $P_i$, $P_{i+1}$, $P_{i+2}$가 필요하고 다음과 같은 (식 1)과 같이 표현된다.

$$c(t)=\begin{bmatrix}t^3 & t^2 & t & 1\end{bmatrix}\frac{1}{2}\begin{bmatrix}-1 & 3 & -3 & 1\\ 2 & -5 & 4 & -1\\ -1 & 0 & 1 & 0\\ 0 & 2 & 0 & 0\end{bmatrix}\begin{bmatrix}P_{i-1}\\ P_i\\ P_{i+1}\\ P_{i+2}\end{bmatrix} \quad (1)$$

$(\text{단}, 0\le t\le 1)$

블록의 최댓값을 계산하는 것은 선형 보간에서는 자명한 일이다. 블록을 구성하는 모든 복셀값을 비교하여 최댓값을 구하면, 그것이 바로 블록의 최댓값이 된다. 그림 5(a)와 같이 선형 보간으로 얻은 재-샘플링 값은 복셀의 밀도값보다 클 수 없기 때문이다. 그러나 삼차 보간을 하게 되면, 최댓값을 구하는 것은 어려운 문제이다. 그림 5(b)와 같이 보간으로 생성된 재-샘플링 값은 복셀의 밀도값보다 클 수 있고 최댓값이 존재하는 위치 또한 복셀이 위치하는 격자점이 아니므로 구간을 검사해야 하는 어려움이 있다.

하나의 블록은 수 만개의 셀(cell, 공간적으로 인접하는 $8(=2^3)$개의 복셀을 꼭지점으로 하는 육면체와 그 내부 영역)로 구성되므로, 하나의 블록을 구성하는 모든 셀에 대해 최댓값을 각각 구하고 그 크기를 비교하여, 가장 큰 값을 블록의 최댓값으로 구할 수 있다.

셀의 최댓값을 구하기 위해, 해석적 방법을 사용할 수 있다. 즉, 공간좌표를 정의역으로 함수를 구성하고, 구간에서 최댓값을 구하는 식을 계산하는 방법을 고려할 수 있다. 해석적 방법은 이론적으로 정확한 값을 얻을 수 있다는 장점이 있지만 계산이 복잡하여 실용적으로 사용하기 어렵다. 실제로 하나의 셀에 대해 해석적 방법을 적용하면 $x$, $y$, $z$의 세 축에 대하여 각각 삼차함수가 구성되므로 일반적으로 (식 2)와 같이 64개의 항을 가진 삼변수 삼차함수가 되어 최댓값을 구하기 매우 어렵다. 게다가 하나의 볼륨 데이터를 구성하는 수 억개에 달하는 셀에 대해 이 계산을 수행하는 것은 오랜 시간이 걸리며, 컴퓨터로 계산할 때 발생하는 수치적 오차를 감안하면 현실적인 해결책이 아니다.

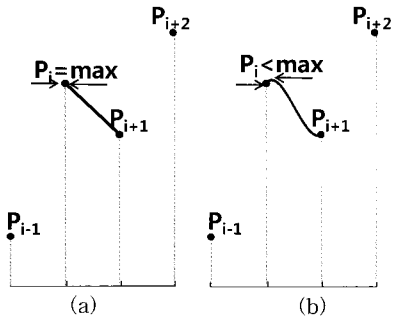



그림 5. 선형 보간과 삼차 보간의 비교

$$f(x,y,z)=a_{3,3,3}x^3y^3z^3+a_{3,3,2}x^3y^3z^2 +a_{3,3,1}x^3y^3z+\ldots+a_{0,0,1}z+a_{0,0,0} \quad (2)$$

또는 각 셀 내부에 대해 극단적으로 조밀하게 수퍼-샘플링(super sampling)을 수행하여, 셀에 대한 최댓값의 근사값을 구할 수도 있다. 그러나 이 방법은 수퍼-샘플링을 했다고 하더라도 수퍼-샘플링 위치 사이에 존재하는 최댓값은 찾지 못하는 경우가 있으며, 수퍼-샘플링 또한 오랜 시간이 걸리는 작업이므로 적합하지 않은 방법이다.

이같이 정확한 최댓값을 구하기 어려운 경우, 2.1절에 논의한 바와 같이 가능한 한 작은 상계를 효과적으로 구하는 방법이 필요하다. 예를 들어, 삼차 보간을 수행하는 네 점의 위치 $P_{i-1}$, $P_i$, $P_{i+1}$, $P_{i+2}$에서 밀도의 최댓값 $M$과 최솟값 $m$을 알고 있다면, 네 점으로 생성할 수 있는 곡선의 최댓값은 $M+(M-m)/8$보다 작다고 알려져 있다[12]. 이 방법은 간편하게 상계를 구할 수 있으나, 곡선의 개형과 관계없이 최악의 경우를 가정하고 상계를 구하기 때문에 상대적으로 효율이 나쁜 문제가 있다.

### 2.3 베지어 보간을 이용한 상계 획득

본 연구에서는 카디널 보간을 베지어(Bezier) 보간으로 변형하여 되도록 작은 상계를 구하는 방법을 제안한다. 이 절에서는 편의를 위해 일반성을 잃지 않고, 삼차원 볼륨 데이터 대신 일차원 데이터를 이용하여 설명한다. 따라서 곡선을 구성하는 제어점(복셀)은 일차원 공간의 좌표인 x값과 해당 위치에서의 밀도값인 y값으로 구성되어 있다. 먼저 베지어 곡선은 제어점 $B_{i,0}$, $B_{i,1}$, $B_{i,2}$, $B_{i,3}$에 대해 (식 3)과 같이

표현되는 삼차 곡선이다[11].

$$B(t)=\begin{bmatrix}t^3 & t^2 & t & 1\end{bmatrix}\begin{bmatrix}-1 & 3 & -3 & 1\\ 3 & -6 & 3 & 0\\ -3 & 3 & 0 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}B_{i\_0}\\ B_{i\_1}\\ B_{i\_2}\\ B_{i\_3}\end{bmatrix} \quad (3)$$

본 연구는 주어진 카디널 곡선의 제어점(복셀)으로부터 (식 4)와 같은 변환을 통해 베지어 곡선의 제어점을 생성하고자 한다.

$$\begin{aligned}B_{i\_0}&=P_i\\ B_{i\_1}&=P_i+(P_{i+1}-P_{i-1})/6\\ B_{i\_2}&=P_{i+1}-(P_{i+2}-P_i)/6\\ B_{i\_3}&=P_{i+1}\end{aligned} \quad (4)$$

이때 생성된 베지어 곡선은 (식 5)와 같이 표현된다.

$$\begin{aligned}B(t)&=\begin{bmatrix}t^3 & t^2 & t & 1\end{bmatrix}\begin{bmatrix}-1 & 3 & -3 & 1\\ 3 & -6 & 3 & 0\\ -3 & 3 & 0 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}P_i\\ P_i+(P_{i+1}-P_{i-1})/6\\ P_{i+1}-(P_{i+2}-P_i)/6\\ P_{i+1}\end{bmatrix}\\ &=\begin{bmatrix}t^3 & t^2 & t & 1\end{bmatrix}\begin{bmatrix}-0.5P_{i-1}+1.5P_i-1.5P_{i+1}+0.5P_{i+2}\\ P_{i-1}-2.5P_i+2P_{i+1}-0.5P_{i+2}\\ -0.5P_{i-1}+0.5P_{i+1}\\ P_{i+1}\end{bmatrix}\\ &=\begin{bmatrix}t^3 & t^2 & t & 1\end{bmatrix}\frac{1}{2}\begin{bmatrix}-1 & 3 & -3 & 1\\ 2 & -5 & 4 & -1\\ -1 & 0 & 1 & 0\\ 0 & 2 & 0 & 0\end{bmatrix}\begin{bmatrix}P_{i-1}\\ P_i\\ P_{i-1}\\ P_{i+2}\end{bmatrix}=c(t)\end{aligned} \quad (5)$$

(식 5)와 (식 1)에 의해 카디널 보간은 베지어 보간으로 표현할 수 있다는 것이 증명된다. 즉, (식 4)와 같이 베지어의 제어점 $B_{i\_0}$, $B_{i\_1}$, $B_{i\_2}$, $B_{i_3}$을 생성하면 전체 보간은 카디널 보간 대신 베지어 보간을 통해 계산할 수 있다.

이 결과를 예를 들면, 그림 6(a)은 제어점 $P_0$, $P_1$, $P_2$, $P_3$로부터 생성한 카디널 보간 결과 곡선이며, 그림 6(b)은 카디널 보간 제어점으로부터 (식 5)를 통해 베지어 제어점 $B_{1\_0}$, $B_{1\_1}$, $B_{1\_2}$, $B_{1_3}$을 생성하고, 베지어 보간을 통해 얻은 곡선이다. 서로 다른 방법으로 생성된 두 곡선은 완전히 일치한다.

본 연구는 카디널 보간의 상계를 얻는 것이 목적인데, 본 연구는 카디널 보간과 동일한 베지어 보간의 상계를 구하는 방법을 제안한다. 베지어 곡선의 기저함수는 구간에서 모두 양수이므로 볼록포 성질(convex hull property)을 갖는다. 따라서 곡선은 (식 3)에서 제어점 $B_{i\_0}$, $B_{i\_1}$, $B_{i\_2}$, $B_{i_3}$을 외곽으로 둘러싼 사각형 내부에 존재하게 된다. 그리고 제어점 $B_{i\_0}$, $B_{i\_1}$, $B_{i\_2}$, $B_{i_3}$중 최댓값은 곡선의 상계가 된다. 예를 들어 그림 6(b)에서 $B_{1_1}$지점의 높이가 곡선으로 그린 함수의 상계인 $M'$가 되며 실제 곡선의 최댓값과 차이가 많이 나지 않음을 알 수 있다.

한편 기존 연구[12]에 따르면 주어진 제어점에서, 카디널 곡선의 최댓값을 구하기 위해, 발생할 수 있는 가장 넓은 범위(최악의 경우)를 가정하여 그림 6(c)와 같이 $M''=M+(M-m)/8$을 상계로 결정한다 (단, $M$는 복셀들의 최대밀도, $m$은 복셀들의 최소밀도). 이 경우, 그림 6(c)의 $M''$보다 제안 방법인 그림 6(b)의 $M'$가 곡선의 최댓값을 더 정확하게 계산하였

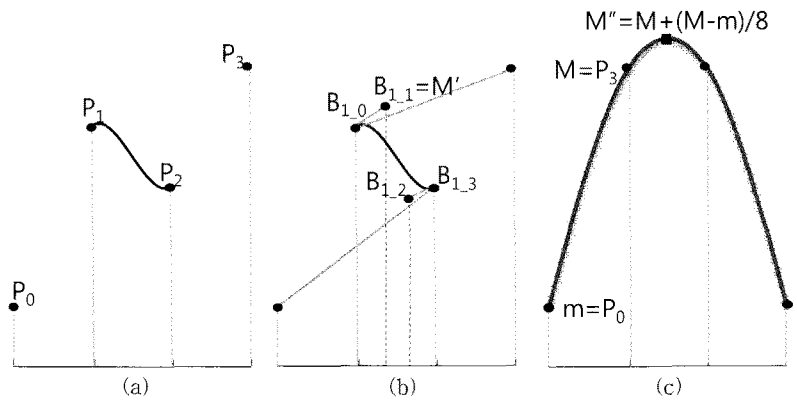



그림 6. 보간 방법과 상계의 비교. P0, P1, P2, P3로부터 카디널 보간 결과(a)와 B1_0, B1_1, B1_2, B1_3로부터 베지어 보간 결과(b)는 동일하다. 같은 조건에서 (c)와 같이 기존 방법[12]을 이용하면 지나치게 큰 상계 M"를 얻게 되어 비효율적이다.

음을 알 수 있다. 다시 말하면, 베지어 보간의 제어점을 활용하면 곡선의 개형을 더 정확하게 알 수 있어, 더 작은 상계를 얻을 수 있다.

지금까지의 결과를 삼차원 공간에 대해 적용하면, 각 셀에 대해 $x$, $y$, $z$ 방향으로 각각 네 개의 베지어 제어점을 생성해야 한다. 하나의 셀에 대해 $64(=4^3)$개의 베지어 제어점을 생성하고, 64개의 밀도값 중 최댓값을 구해 셀의 상계로 결정한다. 한편 64개의 밀도값 중 최솟값은 자명하게 셀의 하계가 된다. 이를 이용하여 삼차 보간 필터 사용시, 화질저하 없이 빈공간 도약을 이용한 가속화를 수행할 수 있다.

### 2.4 전체 알고리즘

전체 알고리즘은 그림 7과 같이 전처리 단계과 가시화 단계의 둘로 나뉜다. 먼저 전처리 단계에서는 볼륨 데이터를 얻어 각 셀에 대해 베지어 제어점을 생성한다. 제어점에 대한 크기 비교를 통해 각 셀의 상한과 하한을 구한다. 블록을 구성하는 각 셀의 상한끼리 비교하여 블록의 최댓값을 결정하고 하한끼리 비교하여 블록의 최솟값을 결정한다. 블록의 최댓값, 최솟값은 별도의 공간에 저장된다.

가시화 단계에서는 일반적인 광선 추적법[2]을 적용한다. 영상을 구성하는 각 화소에 대해 광선을 발사하고, 광선이 진행하는 과정에서 현재 위치를 포함하는 블록의 최댓값, 최솟값을 읽어 투명도를 검사한다. 투명한 블록은 재-샘플링을 생략하고 현재 블록 끝까지 도약하여 가속화가 일어나며, 불투명한 블록은 재-샘플링, 투명도 계산, 색상 계산, 누적 과정을 거쳐 광선의 색을 변경한다. 그리고 현재 위치를 광선 방향으로 약간 움직이는 과정을 반복하게 된다. 이때 조기광선 종료와 같은 가속화 알고리즘도 추가적으로 적용하는데 문제가 없다.

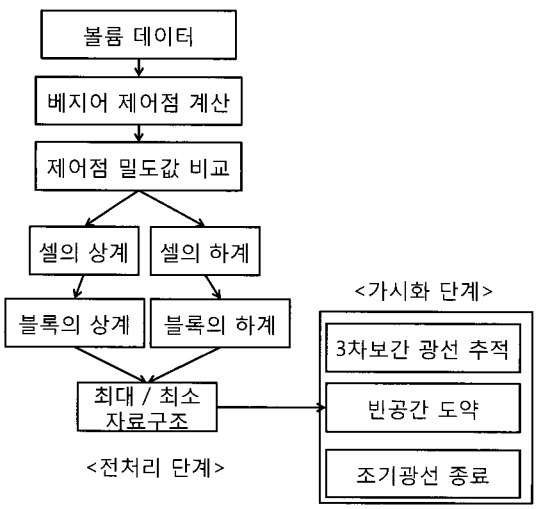



그림 7. 본 연구에서 제안한 알고리즘의 흐름도.

## 3. 실 험

이번 장에서는 제안 방법을 사용하여 결과 영상을 보이고 가시화 속도의 향상 정도를 확인한다. 구현은 Windows XP 운영체제에서 Microsoft Visual studio로 C++을 이용하였으며, 성능 측정을 위한 실험은 특별한 그래픽스 하드웨어를 사용하지 않고, Intel Core2Duo CPU와 1GB의 메모리를 장착한 개인용 컴퓨터에서 수행되었다. 실험에 사용한 데이터는 표 1에 제시되었다. 각 데이터는 인체의 컴퓨터 단층 촬영 데이터이며 각 복셀은 2바이트를 차지한다.

삼차 보간을 사용한 본 연구의 화질을 보이기 위해, 선형 보간을 사용한 가시화 결과와의 비교를 그림 8 (Head), 그림 9 (Chest), 그림 10 (Legs)에 보였다. 본 연구의 삼차 보간 방법이, 세밀한 부분의 곡선을 잘 보여주어, 선형 보간에 비해 두드러진 화질 향상을 보인다. 한편, 빈공간 도약 가속화를 적용하지 못하는 기존의 삼차 보간을 이용한 가시화 방법과 제안 방법을 비교하면 둘은 완전히 동일한 영상을 생성하게 된다. 그 근거로서 2.1절에서 빈공간 도약시 블록의 상한 또는 상계를 사용하는 한 화질 손상이 없음을 이론적으로 보였고, 2.3절에서 제안한 베지어 제어점이 곡선의 상한이 됨을 증명하였다. 두 기법이 동일한 영상을 생성하므로, 화질의 비교는 하지 않았다.

가시화 속도 향상을 분석하기 위해, 제안 기법과 기존의 기법의 가시화 시간을 측정하여 표 2에 비교하였다. 빈공간 도약을 수행하지 않은 삼차 보간 방법(도약없음), 그림 6(c)에 보인 것과 같이, 최댓값을

표 1. 실험에 사용한 볼륨 데이터

| 데이터 이름 | 크 기 | 용량(MB) |
|---|---|---|
| Head | 512×512×300 | 150 |
| Chest | 512×512×300 | 150 |
| Legs | 512×512×200 | 100 |

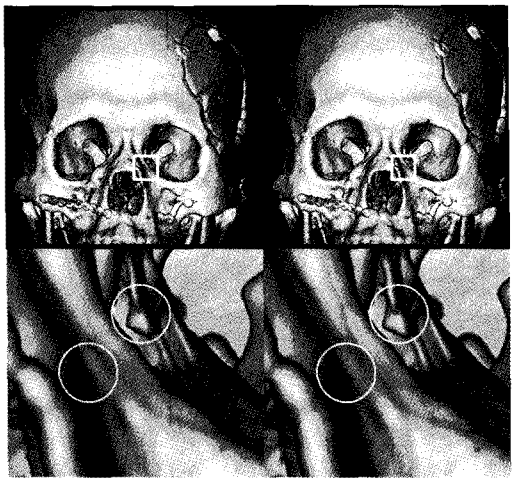


그림 8. Head의 가시화 결과 영상. 왼쪽이 선형 보간이고 오른쪽이 삼차 보간이다. 위의 영역 일부를 확대하여 아래쪽에 도시하였다.

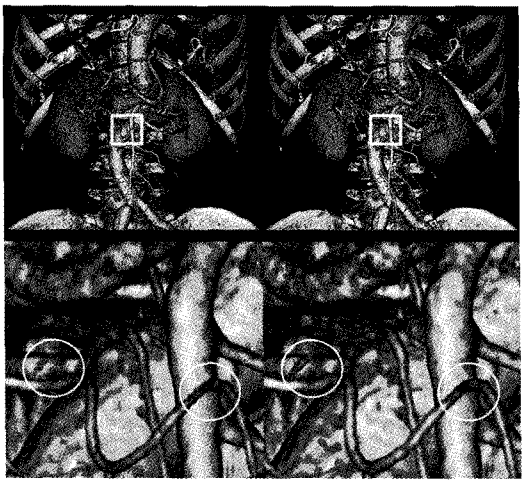


그림 9. Chest의 가시화 결과 영상. 왼쪽이 선형 보간이고 오른쪽이 삼차 보간이다. 위의 영역 일부를 확대하여 아래쪽에 도시하였다.

표 2. 각 실험 데이터에 대한 제안 방법의 성능 향상 결과 (시간단위: 초)

| 알고리즘 | Head | Chest | Leg |
|---|---|---|---|
| 도약 없음 | 10.85 | 23.63 | 39.72 |
| 개선된 최소-최대 블록[12] | 1.57 | 2.35 | 4.09 |
| 제안 알고리즘 | 1.20 | 1.66 | 2.43 |
| 선형 보간 | 0.47 | 0.71 | 1.03 |

결정하는 기존 연구[12]에 근거한 삼차 보간 방법(개선된 최소-최대 블록), 본 연구의 제안방법(제안 알

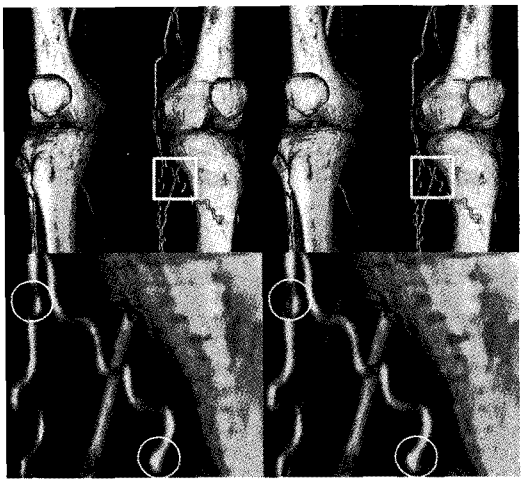


그림 10. Legs의 가시화 결과 영상. 왼쪽이 선형 보간이고 오른쪽이 삼차 보간이다. 위의 영역 일부를 확대하여 아래쪽에 도시하였다.

고리즘)의 속도를 비교하였다. 비교한 세 방법은 모두 동일한 화질의 영상을 생성하게 되므로, 속도의 비교가 곧 종합적인 성능을 비교하는 기준이 된다. 추가로, 화질에서 대등하지 않으나 상대적인 속도 비교를 위해 선형 보간을 이용한 속도를 측정하여 비교하였다. 시간 측정 결과는 각 데이터를 회전시켜 측정한 시간의 평균값이다.

제안 알고리즘은 도약을 하지 않는 방법에 비해 10배 이상으로 성능이 향상되었으며, 개선된 최소-최대 블록[12] 방법과 비교하여도 1.31배(=1.57/1.20) ~ 1.68배(=4.09/2.43)의 두드러진 가속 효과를 얻었다. 한편, 선형 보간 방법과 제안 알고리즘을 비교하면 제안 알고리즘이 2.5배 정도 느린 것으로 측정되었으며, 삼차 보간이 선형 보간에 비해 매우 복잡한 것을 감안하면 적절한 것으로 파악된다.

한편 성능 향상 정도와 데이터의 상관 관계를 살펴보면, 상대적으로 Head 데이터에서 성능 향상이 적고 Legs 데이터에서 성능 향상이 크게 나타난다. 그 이유는 그림 8에서 확인할 수 있듯이, Head 데이터가 볼륨의 외곽을 크게 감싸는 형태의 불투명한 데이터이기 때문으로 파악된다. 이 경우, 빈공간 도약이 많이 이루어지지 않고, 조기광선 종료 가속화의 영향을 많이 받기 때문에 빈공간 도약을 통한 성능 향상 폭이 적어지게 된다. 반대로, 상대적으로 복잡한 형태의 혈관 등의 구조를 가시화 하는 경우(Legs 데이터), 본 알고리즘은 더욱 효과가 크게 나타나게

된다.

제안 알고리즘의 가시화 시간이 1초 이상 걸리는 것으로 측정되어, 제안 알고리즘은 실시간 가시화 속도에 미치지 못한다. 그러나 선형 보간 방법과 비교하면 화질 향상을 고려하였을 때, 충분히 사용할 수 있는 수준이다. 그리고 본 실험의 결과가 보급형의 범용 CPU를 이용한 측정 결과임을 감안하였을 때, DirectX[13] 등의 그래픽스 라이브러리나 GPGPU [14]와 같은 그래픽스 병렬화 도구, 또는 CPU에서 지원하는 단일연산 복수데이터(single instruction multiple data) 명령[15] 등을 사용한다면 실시간 처리에 근접한 속도를 얻을 수 있을 것으로 보인다.

마지막으로 블록의 상계와 하계를 얻는데 필요한 전처리 시간을 측정한 결과를 표 3에 제시하였다. 베지어 제어점을 생성하는 것은 데이터의 내용에 독립적이므로, 전처리 시간은 데이터 크기에 거의 비례한다. 한편, 전처리가 각 데이터에 대해 한 번만 수행되는 것을 감안 하더라도, 상대적으로 오랜 시간이 걸리는 것으로 측정되었다. 추후 가속화 기법을 적용할 수 있을 것으로 파악된다.

표 3. 전처리에 소요되는 시간 측정

| 데이터 이름 | 크기 | 시간 (초) |
|---|---|---|
| Head | 512×512×300 | 208.5 |
| Chest | 512×512×300 | 215.4 |
| Leg | 512×512×200 | 127.8 |

## 4. 결론 및 향후과제

본 연구는 삼차 보간을 사용하는 고화질 볼륨가시화에 적용하는 빈공간 도약 기법을 제안하였다. 블록의 최대, 최소를 구하기 어려운 점에 착안하여 상계와 하계를 이용한 가속 기법을 제안하였고, 구체적으로 카디널 보간을 베지어 보간으로 변형하고, 베지어 보간의 볼록포 성질을 이용하여 상계와 하계를 효율적으로 얻을 수 있었다. 그 결과로 삼차 보간 필터를 사용하는 고화질 볼륨가시화의 속도를 대폭 향상시켰다. 본 연구에서 제안하는 가속화 알고리즘은 조기 광선 종료 등의 다른 가속화 알고리즘을 적용하는데 문제가 없다.

한편, 본 연구는 CPU에서 실행되어 절대적인 가시화 속도는 느린 편인데, 그래픽스 하드웨어를 이용하여 구현하면 실용적인 성능을 확인할 수 있을 것으로 기대한다. 또한 전처리 과정에서 베지어곡선의 제어점 생성에 많은 시간이 소모되는데, 향후 과제로서 계산량을 줄이고 시간을 절약하는 방법을 연구하려 한다.

## 참 고 문 헌

[1] K. Engel, M. Hadwiger, J. M. Kniss, C. Rezk-Salama, and D. Weiskopf, *Real-Time Volume Graphics*, Wellesley, Massachusetts, 2006.

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

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

[4] G. Knittel, "The Ultravis System," IEEE/ACM SIGGRAPH Volume visualization and graphics symposium, pp. 71-78, 2000.

[5] W. Li, K. Mueller, and A. Kaufman, "Empty Space Skipping and Occlusion Clipping For Texture-Based Volume Rendering," In Proc. of IEEE Visualization Conference (2003), pp. 317- 324, 2003.

[6] M. Levoy, "Volume Rendering Display of Surfaces from Volume Data," *IEEE Computer Graphics and Application*, Vol.8, pp. 29-37, 1988.

[7] K. Bjorke, "High-Quality Filtering," *GPU Gems*, R. Fernando, Ed., Upper Saddle River, NJ: Addison-Wesley, pp. 391-415, 2004.

[8] C. Sigg and M. Hadwiger, "Fast third-order texture filtering," *GPU Gems 2*, M. Pharr, Ed., Addison Wesley, Los Alamitos, CA, pp. 313-329, 2005.

[9] 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.

[10] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler, "The VolumePro Real-time Ray-Casting System," SIGGRAPH '99, pp. 251-260, 1999.

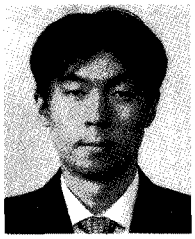
[11] G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design 4th Ed.*, Academic Press, San Diego, CA, 1997.

[12] B. Lee, J. Yun, J. Seo, B. Shim, Y. Shin, and B. Kim, "Fast High-Quality Volume Ray-Casting with Virtual Samplings," *IEEE TVCG*, Vol.16, No.6, pp. 1525-1532, 2010.

[13] DirectX, Available online (http://www.microsoft.com/directx)

[14] CUDA, Available online (http://gpgpu.org)

[15] 계희원, "단일 명령 복수 데이터 연산과 순차적 메모리 참조를 이용한 효율적인 최대 휘소 투영 볼륨 가시화," 한국멀티미디어학회 논문지, 제 12권, 4호, pp. 512-520, 2009.



계 희 원

1999년 2월 서울대학교 전산과학과 학사

2001년 2월 서울대학교 전기컴퓨터공학부 석사

2005년 8월 서울대학교 전기컴퓨터공학부 박사

2006년 1월~2007년 3월 서울대학교 컴퓨터연구소 연구원

2007년 9월~현재 한성대학교 정보시스템공학과 조교수

관심분야 : 볼륨 가시화, 실시간 렌더링, 대용량 영상처리