

# 방대한 볼륨 데이터의 인터랙티브 가시화를 위한 웨이블릿 기반 3차원 압축 기법과 네비게이션 시스템의 개발 Development of Wavelet-Based 3D Compression Scheme and Navigation System for Interactive Visualization of Very Large Volume Data

박 상 훈      임 인 성

Sanghun Park    Insung Ihm

서강대학교 전자계산학과

Department of Computer Science

Sogang University

## 요 약

방대한 볼륨 데이터의 가시화는 상당히 많은 계산 비용과 메모리 공간을 필요로 하는 작업이다. 본 논문에서는 방대한 볼륨 데이터를 효과적으로 압축할 수 있는 웨이블릿에 기반을 둔 3차원적인 압축 기법을 제안하고, 이 기법으로 압축된 데이터를 인터랙티브하게 가시화할 수 있는 네비게이션 시스템의 구현에 대해 설명한다. 본 압축 기법은 방대한 볼륨 데이터를 효과적으로 압축함으로써, 64~128MByte 정도의 제한된 메인 메모리를 갖는 범용의 PC나 워크스테이션을 이용해 수 GByte에 이르는 볼륨 데이터를 효과적으로 가시화하는 것을 목표로 한다. 따라서, 개발된 압축 스킴은 데이터 전체에 대한 복원없이 가시화가 필요한 부분의 데이터만을 빠르게 복원할 수 있는 랜덤 액세스가 가능하도록 설계되었으며, 적은 계산 비용으로 복원이 가능하다는 장점을 갖는다. 개발된 네비게이션 시스템은 가시화 속도 향상을 위한 여러가지 효과적인 렌더링 모듈과 사용자 인터페이스 모듈로 구성되며, 현재 성능 향상을 위한 연구가 계속 진행되고 있다.

## 1 서론

볼륨 가시화는 상당히 많은 계산량과 메모리 공간을 필요로 하는 작업이다. 따라서 지금까지의 대부분의 연구들은 256x256x256 정도의 비교적 작은 데이터를 대상으로 수행되어 왔다. 그러나 최근에 자연과학, 공학, 의학 등의 분야에서 방대한 크기를 갖는 여러 가지 데이터들이 산출되고 있으며, 이러한 데이터에 대한 가시화 요구가 증가되고 있다. 특히, 미국의 NLM (National Library of Medicine)에서는 인체 내부에 대한 심도 있는 연구를 위해 남녀 각각의 시신에 대해 CT, MRI, RGB 이미지 데이터를 생성하는 Visible Human Project를 수행하였다 [11]. 이 프로젝트를 통해 만들어진 남자 데이터 (Visible Man)와 여자 데이터 (Visible Woman) 각각의 크기는 약 15GByte와 40GByte이다. 본 논문에서 실험을 위해 사용한 Visible Man의 CT 데이터의 경우 512x512x1878의 볼륨 해상도를 갖고 각 복셀(voxel)은 12bits로 표현된다.

기존의 기법들을 이용하여 이러한 데이터를 가시

화하기 위해서는 전체 데이터를 로드(load)할 수 있을 정도로 큰 메인 메모리와 매우 빠른 CPU를 갖는 컴퓨터를 사용해야만 한다. 즉, 고성능의 병렬 컴퓨터를 사용하지 않고 이러한 방대한 크기의 데이터를 기존의 가시화 기법으로 가시화 하는 것은 거의 불가능하다[9,13,14]. 본 논문은 방대한 데이터의 가시화를 64~128MByte 정도의 메모리를 갖는 PC나 워크스테이션에서 수행할 수 있도록 만들어 주는 효과적인 압축 기법을 제시한다[5,6,12]. 이것은 방대한 볼륨 데이터에 대해 웨이블릿을 이용한 3차원적인 압축을 수행하고, 압축된 전체 볼륨 데이터를 64~128MByte의 메인 메모리로 로드하여 현재 계산을 위해 필요한 부분만을 복원하고 가시화를 수행하는 구조를 갖는다. 이러한 기법의 성공적인 수행을 위해서는 높은 압축율을 가지면서 전체 데이터를 복원하지 않고, 추가적인 메모리의 사용 없이 필요한 부분의 복셀값만을 빠르게 복원할 수 있는 효과적인 압축 스킴의 개발이 필수적이다.

Muraki는 볼륨 데이터에 대해 3차원적인 웨이블

릿 압축을 처음으로 적용하였다[8]. 그러나 그의 연구 결과에는 압축된 데이터를 가시화할 때 사용되는 실제 복원 데이터의 크기에 대한 언급이 없고, 랜덤 액세스나 복원 속도에 대한 고려도 없다. 또한 복원된 데이터에 의해 가시화된 이미지의 화질도 그리 좋지 않다. 그 밖에도 기존의 압축 기법들을 응용해 볼륨 데이터를 압축하고 가시화하는 몇 가지 시도가 있었으나 압축율도 그리 높지 않고, 복원 화질도 그리 좋지 않은 결과를 나타낸다[3,10].

## 2 웨이블릿 기반 압축

### 2.1 3D 웨이블릿 변환

웨이블릿이란 함수를 계층적으로 표현하는 수학적 도구로서, 함수를 다해상도(multi-resolution)로 표현할 수 있고 데이터 압축을 위한 수학적 기반을 제공한다[1,2,15,17]. 웨이블릿 변환에서 사용되는 여러 기저(basis)들 중에서 Haar 기저는 가장 간단하고 계산량이 적다는 장점을 갖고 있다. 즉, 변환 과정이 덧셈, 뺄셈 그리고 쉬프트(shift) 연산만으로 단순하게 수행될 수 있기 때문에 속도가 빠르다. 따라서 본 논문과 같이 빠른 랜덤 액세스를 필요로 하는 응용이나 많은 데이터를 실시간으로 복원해야 하는 응용에서 상당히 적합한 기저라고 할 수 있다.

웨이블릿 변환을 통해 주어진 데이터를 반복적으로 분해(decomposition)한 후, 사용된 전체 계수들 중에서 절댓값(magnitude)이 어떤 한계값(threshold) 이하인 계수들을 잘라내고 (0으로 치환하고), 남은 계수들만을 저장함으로써 압축이 이뤄지게 된다.

본 논문에서는 전체 볼륨 데이터를 16x16x16의 크기를 갖는 단위블럭(unit block)이라 불리는 부분 블럭으로 분할하고, 각 단위블럭에 대해 3D 웨이블릿 변환을 2회 반복적으로 수행하였다. 복원을 위한 8개의 수식에서 나타나는 56번의 +/- 연산은 중복된 계산을 제거함으로써 24번으로 줄일 수 있다. 복원 공식은 가시화를 수행하면서 랜덤 액세스하는 과정에서 적용되며, 수식에서 사용되는 계수의 90% 이상은 이미 0으로 되어 있다.

### 2.2 압축 스킴

Shapiro는 웨이블릿 계수들이 갖는 특별한 성질을 이용하여 상당히 좋은 성능을 갖는 제로트리 인코딩(zero-tree encoding)이라 불리는 기법을 구현하였다. 그러나 디코딩 속도가 너무 느리기 때문에, 빠른 랜덤 액세스를 필요로 하는 응용에서는 적당하지 않은 방법이다[16].

본 논문에서 구현된 인코딩 스킴은 높은 압축율과 빠른 랜덤 액세스라는 두 가지의 목표를 위해서 구현되었다. 그림1은 본 논문에서 구현된 인코딩 스킴

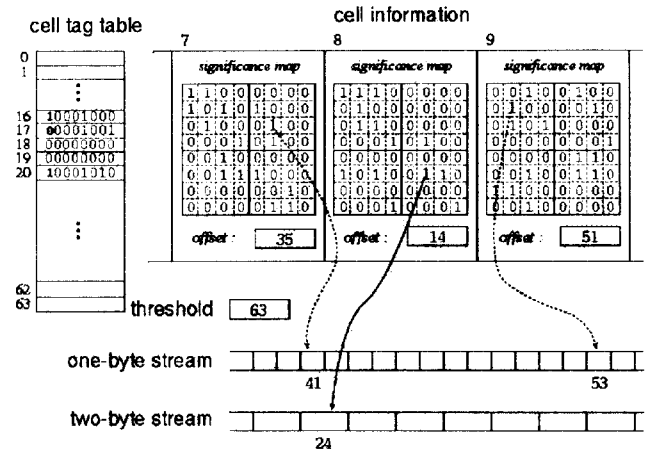
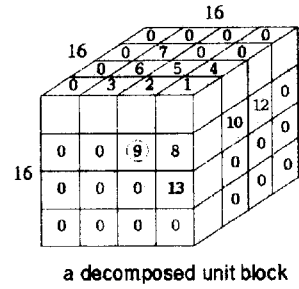


그림 1: 3차원 웨이블릿 인코딩 스킴

김을 보여준다. 웨이블릿으로 분해된 16x16x16 크기의 단위블럭을 살펴보면 이미 0으로 치환된 한계값  $\tau$  이하인 계수들이 전체에서 많은 부분 (일반적으로, 90% 이상)을 차지하고, 또한 데이터의 인접성(coherence)으로 인해 0인 계수들이 모여있을 가능성이 높다는 사실을 발견할 수가 있다. 따라서, 분해된 하나의 단위블럭을 다시 4x4x4 크기를 갖는 셀(cell)이라 불리는  $4^3(=64)$ 개의 부분 블럭으로 분할한다. 그리고 각 셀에 포함된 계수들이 모두 0인지 아닌지를 확인하는 과정을 거친다. 만일 어떤 셀에 포함된 계수가 모두 0인 경우라면 그 셀의 태그(tag)를 0으로 만들고, 0이 아닌 웨이블릿 계수가 하나라도 존재하는 경우라면 1부터 시작해서 하나씩 증가된 태그를 각 셀에 할당한다. 이 태그는 셀 태그 테이블(cell tag table)에 저장된다.

원래 Visible Human CT 데이터가 각 복셀당 12bits를 사용하기 때문에 웨이블릿 변환이 수행된 후, 모든 웨이블릿 계수는 2bytes에 저장될 수 있다. 그러나 효과적인 메모리 사용을 위해, 태그가 0이 아닌 셀을 두 가지 타입(type)으로 구분함으로써 압축율을 높일 수 있다. 첫번째 타입은 셀 내의 모든 계수가  $[-(\tau+128), -\tau]$  또는  $[\tau, \tau+127]$ 의 범위에 있는 경우이고, 두번째 타입은 위의 범위에 속하지 않

는 계수를 하나 이상 포함하는 경우이다. 이와같이 두가지로 구분된 셀의 타입에 대한 정보는 태그를 저장하는 셀 태그 테이블내에 인코딩 된다. 그리고 두 가지 타입의 셀에 포함된 계수들은 1-byte 스트림(stream)과 2-byte 스트림이라 불리는 두 개의 1차원 배열에 저장된다. 복원 과정에서 랜덤 액세스를 지원하기 위해서는 위치에 대한 정보도 함께 인코딩해야 하는데, 이를 위해 셀 정보(cell information)라 불리는 추가적인 정보의 유지가 필요하다. 여기에는 셀 내의 각 계수값이 여기에는 계수가 0인지 아닌지를 표현하는 의미지도(significance map)라 불리는 4x4x4의 비트-플래그(bit-flag)와 현재 셀내에서 첫 번째로 0이 아닌 계수에 대한 해당 스트림 내에서의 주소를 포함하는 오프셋(offset)정보가 포함된다.

그림 2의 알고리즘은 분해되고 인코딩된 하나의 16x16x16의 단위블럭으로 부터 인덱스 (i,j,k)에 위치한 계수가 어떻게 디코딩 되는지를 보여준다.

### 3 성능 분석

#### 3.1 복원과 저장 비용

하나의 단위블럭으로 부터 하나의 복셀값을 복원하는 경우, 하나의 복셀을 복원하기 위해서는 14번의 +/-과 15개의 웨이블릿 계수들이 디코딩되어야만 한다. 그림2의 알고리즘에서, [case 1], [case 2]인 경우는 매우 적은 비용으로 디코딩이 가능하고 전체 계수의 90% 이상이 여기에 해당된다. [case3]의 경우라 하더라도 평균적으로 2.5번의 테이블 액세스와 해당 스트림의 주소를 결정하기 위한 추가적인 덧셈만으로 디코딩이 이뤄진다. 따라서, 소비되는 전체적인 계산 비용은 상당히 낮다.

블룸 가시화 기법들은 복셀값을 액세스하는 패턴에서 상당히 높은 지역성(locality)을 나타낸다. 따라서, 성능의 향상을 위해 단위블럭 내의 4x4x4 부분영역을 새로운 복원 단위로 생각할 수 있다. 64(=4·4·4)개의 전체 복셀들의 복원을 위해 한 복셀당 평균 (9·24)/64 = 3.375번의 +/-연산과 1번의 디코딩이 필요하다. 물론 비트 연산이나 주소 계산을 위한 추가적인 비용이 필요하지만 데이터 압축의 효율성을 고려할 때 이것은 상당히 적은 비용이 된다. 단위블럭은 압축전에 16x16x16x2bytes를 갖고 있다. 우선 한계값과 태그 테이블을 저장하기 위해 각각 2bytes와 64bytes가 소비된다. 적어도 하나의 0이 아닌 계수를 포함하는 셀의 경우는 추가적인 정보의 저장에 필요한데, 의미지도의 저장을 위한 8bytes(=4·4·4 bits)와 오프셋을 위해 2bytes가 필요하다. 전체 셀의 개수 (4·4·4 = 64)중에서 0이 아닌 태그를 갖는 셀의 갯수의 비율을

- Find the cell C that contains the index (i,j,k).
- If the tag for C is 0, return 0. [case 1]
- Compute the relative index (i',j',k') in C.
- If the bit-flag for (i',j',k') is 0, return 0. [case 2]
- Count the number of preceding non-zero coefficients by table access.
- Add the displacement to the offset to compute the correct address.
- Access the appropriate data stream, and return the value. [case 3]

그림 2: 웨이블릿 디코딩 알고리즘

$\alpha$  (=  $\frac{\text{0이 아닌 셀의 갯수}}{64}$ )라고 하자. 이것은 셀 정보를 저장하기 위해 (8+2)·64· $\alpha$  bytes가 필요함을 의미한다. 그리고 0이 아닌 전체 계수 중에서 1-byte 스트림에 저장되는 계수의 비율을  $\beta$  (=  $\frac{1\text{-byte 스트림에 저장된 계수의 갯수}}{\text{사용된 모든 0이 아닌 계수의 갯수}}$ )라고 하고, 압축에서 사용된 0이 아닌 계수의 비율을  $\gamma$ 라고 하자. 그러면 압축을  $\rho$ 는 다음과 같이 정리된다:

$$\begin{aligned} \rho &= \frac{2+64+(8+2)\cdot 64\cdot \alpha+16\cdot 16\cdot 16\cdot \gamma\cdot (\beta+2\cdot (1-\beta))}{16\cdot 16\cdot 16\cdot 2} \\ &= \frac{33}{4096} + \frac{5\cdot \alpha}{64} + \gamma\cdot (1-\frac{\beta}{2}) \\ &\approx 0.008057 + 0.078125\cdot \alpha + \gamma\cdot (1-0.5\cdot \beta) \end{aligned}$$

#### 3.2 실험 결과

본 실험은 MIPS R10000 CPU (175MHz)와 256MByte의 메인 메모리를 갖는 SGI Octane 워크스테이션에서 수행되었다. 표 1의 압축율과 복원 화질에 대한 실험은 512x512x1440 해상도를 갖는 720MByte의 Visible Man Fresh CT 데이터에 대한 실험 결과이다. 그림 3은 피부(skin)에 해당하는 물질을 블룸 광선추적법(volume ray-casting)을 이용하여 가시화한 이미지를 보여준다. 실제로, 10%와 15%로 압축된 데이터에 대한 이미지는 원본 이미지와 거의 구별할 수 없을 정도의 좋은 화질을 나타낸다. 원본 데이터와 압축으로부터 복원된 데이터의 복셀값 사이의 차이(difference)나 왜곡(distortion)을 수치적으로 측정하기 위해, SNR과 PSNR을 3차원 블룸 데이터에 대한 경우로 확장하여 적용하였고, 동일한 샘플링 위치에서 각각의 노말벡터(normal vector)가 이루는 오차 각도의 크기를 측정하였다. 실제 실험에서는 뼈와 피부 각각에 해당하는 복셀에 대해서만 노말벡터를 계산하였다.

표 2의 복원 시간에 대한 실험 결과는 동일한 데이터의 상위 512개 슬라이스를 대상으로 하였다 (512x512x512x2byte = 256MByte). 랜덤 액세스 속

		Desired Ratio of the Wavelet Coef's Used				
		3%	5%	7%	10%	15%
Compression Performance	Size(MB)	26.31	38.27	48.98	63.69	85.99
	p	27.4	18.8	14.7	11.3	8.3
	a	0.12	0.19	0.25	0.32	0.40
	β	0.66	0.73	0.78	0.82	0.86
	γ	0.03	0.05	0.07	0.09	0.11
Errors in Voxel Values	SNR(dB)	22.6	25.9	28.6	32.0	36.4
	PSNR(dB)	44.5	47.8	50.4	53.8	58.2
Errors in Normals	Skin(deg)	15.4	11.4	8.6	6.0	3.9
	Bone(deg)	24.7	19.0	15.0	10.8	6.7

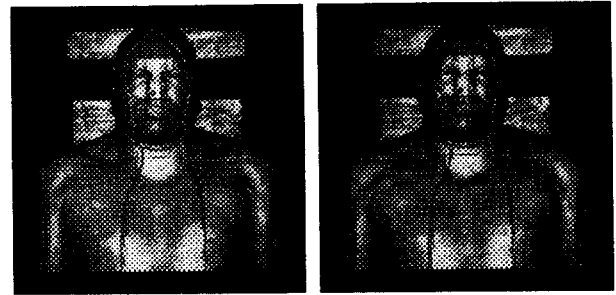
표 1: 압축율과 복원 화질에 대한 실험 결과

		Desired Ratio of the Wavelet Coef's Used					
		Original	3%	5%	7%	10%	15%
Pure Random		2.18	7.75	8.20	8.62	9.15	9.84
Cell-Wise	All	19.95	30.31	31.58	32.52	33.62	35.05
	Skin	6.70	10.13	10.75	11.32	12.10	13.02

표 2: 복원 시간에 대한 실험 결과

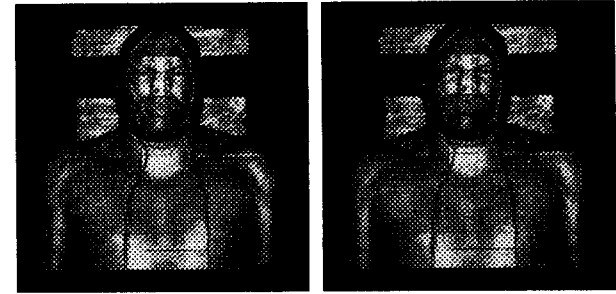
도 측정을 위해 두 가지 경우를 고려했다. 첫 번째는 순수한 랜덤 액세스(pure random access)의 경우이다. 복원을 위한 오버헤드(overhead)를 측정하기 위해 압축하지 않은 볼륨 데이터를 배열로부터 백만회에 걸쳐 랜덤 액세스하고 동일한 수행을 압축된 데이터에 대해 적용하였다. 실험 결과, 압축된 데이터로부터 액세스하는 시간이 약 4배정도 더 소비됨을 확인할 수 있었다. 두 번째는 4x4x4의 셀 단위로 복원을 수행하는 경우이다. 우선, 모든 셀이 앞에서 뒤의 순서로 복원되는데 소비되는 시간을 측정하였다. 그리고, 뼈나 피부에 해당하는 복셀을 적어도 하나 이상 포함하고 있는 단위블럭 내의 셀만을 복원하는데 걸린 시간을 측정하였다. 이러한 선택적인 복원을 수행할 때는 각 단위블럭에 대한 최대-최소 값 정보를 이용하였으며, 피부의 경우에 실제로 11,000개의 단위블럭(16<sup>3</sup> · 11,000개의 복셀)이 복원되었다. 결과에 의하면 모든 단위블럭을 복원하는 경우, 압축된 데이터로부터의 복원 시간이 원본 데이터의 액세스 시간의 약 1.6배 정도가 됨을 확인할 수 있었다. 그리고 피부에 대해 수행한 복원의 경우, 원본 데이터와 압축된 데이터로부터의 복원 시간이 약 2.4초에서 6.4초밖에 차이가 나지 않음을 확인할 수 있었는데, 이것은 볼륨 렌더링과 같이 CPU 시간이 수행 시간의 대부분을 차지하고 수백 초 이상을 필요로 하는 응용에서는 거의 무시할 수 있는 시간 차이이다.

#### 4 볼륨 네비게이션 시스템



(a) Original

(b) 7%



(d) 5%

(c) 3%

그림 3: Visible Man의 볼륨 렌더링 이미지

본 절에서는 앞에서 설명한 압축 기법을 통해 압축된 볼륨 데이터를 64~128 MByte의 메인 메모리를 갖는 PC나 범용의 워크스테이션상에서 인터랙티브하게 네비게이션할 수 있는 SGVN (SoGang Volume Navigation) 시스템에 대해 설명한다. 본 시스템은 사용자 인터페이스를 이용해 가시화를 원하는 부분을 정확히 정의할 수 있고, 카메라의 이동도 자유롭게 제어 가능하다. 또한 인터페이스를 통해 가시화 파라미터들을 변화시키면서 다양한 결과의 이미지들을 생성할 수 있다.

Hong은 Visible Human 데이터의 대장(colon)을 가상적으로 탐사하는 대장 네비게이션 시스템(colonoscopy system)을 개발했다[4]. 그들은 전처리를 통해 대장만을 추출하고 이를 다각형 데이터(polygon data)로 변환시켜 시스템에서 사용하기 때문에 빠른 속도로 네비게이션이 가능하지만 직접 볼륨 렌더링(direct volume rendering)에 의해 생성된 이미지가 아니라는 한계를 갖고 있다. Visible Human 데이터를 매우 섬세하게 가시화하는 상용화된 시스템으로 Voxel Man이 있다[18]. 이 시스템은 CT, MRI 데이터와 RGB 데이터를 이용하여 매우 다양하고 좋은 화질의 이미지 생성하지만, Visible Human 데이터의 머리 부분만을 가시화하기 위해서 96MByte 이상의 메인 메모리를 필요로 하기 때문에, 전체 데이터의 네비게이션을 위해서는 수십

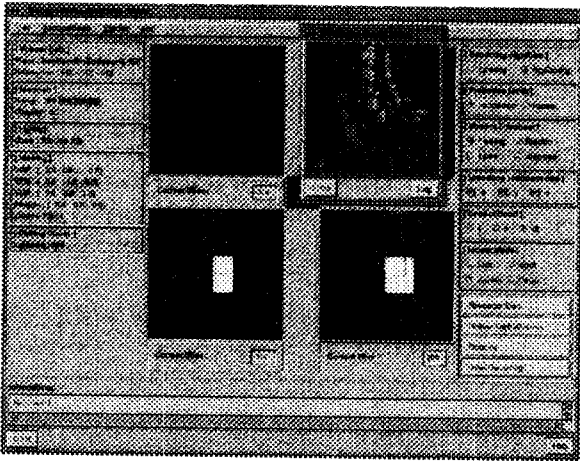


그림 4: SGVN 시스템의 사용자 인터페이스

GByte의 메모리가 요구된다는 한계를 갖는다.

## 5 효과적인 볼륨 네비게이션

### 5.1 캐쉬

일반적으로 네비게이션은 연속된 이미지가 부드럽게 실시간으로 변화하는 형태로 나타나기 때문에 현재 프레임(frame)에서 가시화를 위해 사용된 복셀 정보가 다음 프레임을 위해 다시 사용되는 것이 일반적이다. 따라서 이러한 정보들을 캐쉬(cache)에 저장함으로써 불필요한 반복적인 복원 연산 비용과 노말벡터(normal vector), 그레이디언트(gradient)를 계산하기 위한 반복적인 추가 비용을 상당히 많이 줄일 수 있으며, 전체적으로 가시화 속도를 향상시킬 수 있다. 물론 적절한 캐쉬의 크기를 결정하기 위해서 사용되는 데이터의 크기, 네비게이션을 수행하는 컴퓨터의 메인 메모리 크기 등이 고려되어야 한다.

### 5.2 속도 향상 기법

Visible Man 데이터를 이용하여 네비게이션을 수행하는 경우, 가시화를 위해 정의된 물질의 밀도값의 범위를 결정하게 되는데 이 값의 범위에 따라 뼈가 가시화 되기도 하고 피부가 가시화 되기도 한다. 그리고 데이터의 인접성(coherence)에 의해 유사한 밀도값을 갖는 복셀들이 모여있는 경우가 많다. 따라서 압축의 단위로 사용된 각 단위블럭들에 대한 최대, 최소 복셀값의 정보를 유지함으로써 가시화 속도를 향상시킬 수 있다. 그리고 광선추적법(ray-casting)을 가시화 알고리즘으로 사용하는 경우, 이른 광선 종료(early ray termination) 기법[7]을 이용하여 속도를 향상시켰고, 실험을 통해 최적의 샘플링(sampling) 간격을 결정하므로써 속도와

화질을 모두 높일 수 있었다. 일반적으로 광선추적법에 비해 화질은 조금 떨어지지만 더 빠른 속도를 내는 것으로 알려진 스플래팅(splatting) 기법을 사용하는 경우에는 풋프린트 테이블(footprint table)의 크기를 적절하게 조절함으로써 가시화 속도를 향상시킬 수 있다.

### 5.3 LOD

웨이블릿 기반 압축 기법이 갖는 좋은 특성 중에 하나는 다해상도(multi-resolution) 표현이 가능하다는 사실이다. 이것은 자연스럽게 LOD (Level Of Details)를 구현할 수 있도록 만들어 준다. 2번에 걸쳐 분해된 16x16x16 크기의 단위블럭에는 기본적으로 4x4x4 해상도를 갖는 축소된 볼륨 데이터가 내재되어 있고, 한번의 복원 과정을 통해 8x8x8 해상도의 볼륨 데이터가 생성된다. 물론 복원이 한번 더 수행되면 원래 단위블럭의 해상도를 갖는 볼륨을 얻게된다. 따라서, 저해상도의 볼륨으로 가시화를 하게 되면 화질은 떨어지지만 복원 속도가 빠르기 때문에 가시화 속도를 높일 수 있다. 반대로 고해상도를 사용하는 경우라면 가시화 속도는 상대적으로 느려지지만 좋은 화질을 얻을 수 있다.

## 6 사용자 인터페이스

그림 4는 구현된 SGVN 시스템의 사용자 인터페이스를 보여준다. 가운데 윈도우의 이미지는 현재 뷰 볼륨으로 선택된 목뼈 부분을 광선추적법을 이용하여 가시화한 결과를 보여주고 있다.

Visible Human CT 데이터는 머리에서 발끝까지를 1878개로 나눈 각 단면을 512x512의 해상도를 갖는 화일로 구성된다. 본 시스템에서는 128x128의 해상도를 갖는 3개의 윈도우를 이용해 위에서 본 단면 이미지와 정면, 측면에서 본 단면 이미지를 디스플레이 한다. 이때, 3개의 단면 이미지의 해상도가 128x128로 결정되어 있으므로, 웨이블릿의 다해상도 표현을 이용하여 복원 과정을 1회만 수행하고 가시화함으로써 디스플레이 속도를 향상시킬 수 있다. 마우스의 클릭으로 3가지 방향에 대한 새로운 단면 이미지를 순차적으로 디스플레이할 수 있고, 마우스의 끌기(dragging)를 이용하거나 또는 텍스트 윈도우에 직접 단면 번호를 입력함으로써 임의의 단면을 디스플레이 할 수 있다.

볼륨 데이터 내부의 임의의 부분에 대한 가시화를 위해 뷰 볼륨을 설정해야 한다. 본 시스템에서는 3가지 방향에 대한 단면 이미지에서 초기 카메라의 위치와 방향, 그리고 뷰 볼륨의 크기 등을 마우스를 이용해 자유롭게 결정할 수 있도록 구현하였다.

카메라 제어를 위해 가능한 변환에는 이동

(panning), 회전(rotation), 확대(zoom), 축 고정(axis locking)등의 기능이 있고, 이들은 마우스를 이용해 수행된다. 이러한 수행의 결과는 윈도우내의 좌표의 변화를 통해 확인할 수 있다.

메인 메뉴는 다른 일반적인 시스템에서 쉽게 볼 수 있는 기본적인 기능들과 몇가지 특별한 기능들로 이루어져 있으며, 네비게이션을 하는 동안에 수시로 바꿀 수 있는 선택 메뉴들은 단면 이미지를 디스플레이하는 윈도우들의 오른쪽에 정리하였다.

## 7 결론

본 논문에서는 방대한 볼륨 데이터의 가시화 문제를 일반적인 계산 환경에서 가능하도록 하는 효과적인 웨이블릿 기반 3D 압축 스킴을 제안하였으며, 성능 분석을 통해 높은 압축율과 빠른 랜덤 액세스가 가능함을 확인하였다. 이 기법은 제한된 메모리를 갖는 PC나 범용의 워크스테이션상에서 방대한 데이터의 가시화를 효과적으로 가능하도록 만들어 줌으로써 고성능의 컴퓨터를 접하기 힘든 일반 사용자들에게 유용한 가시화 정보를 제공할 수 있다. 현재는 Daubechies 등과 같은 기저를 이용하여 실험을 진행하고 있다.

그리고, 제한된 메모리를 갖는 PC나 범용의 워크스테이션을 이용하여 데이터를 인터랙티브하게 네비게이션하는 시스템을 개발하였다. 웨이블릿의 다해상도 표현을 이용한 LOD의 구현과 캐쉬와 기존의 효과적인 가시화 기법들을 통해 효과적인 볼륨 네비게이션이 가능하다. 또한 GUI를 통해 카메라의 위치 결정과 가시화 파라미터 설정 등이 가능하도록 구현하였다. 본 시스템이 더 좋은 성능을 내기 위해서는 각 물질에 대한 정확한 분류(classification)의 문제를 해결해야 하고, 가시화의 속도를 더 향상 시켜야 한다. 그리고 압축된 RGB 데이터를 가시화에 이용함으로써 더 좋은 이미지를 얻을 수 있도록 확장시킬 계획이다.

## 참고문헌

- [1] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.
- [2] A. Fournier, editor. *Wavelets and Their Applications in Computer Graphics*. ACM SIGGRAPH '95 Course Notes, 1995.
- [3] M. H. Ghavamnia and X. D. Yang. Direct rendering of Laplacian pyramid compressed volume data. In *Proceedings of IEEE Visualization '95*, pages 192-199, October 1995.
- [4] L. Hong, S. Muraki, A. Kaufman, D. Bartz, and T. He. Virtual voyage: Interactive navigation in the human colon. In *Computer graphics : SIGGRAPH '97 Proceedings*, pages 27-34, August 1997.
- [5] I. Ihm and S. Park. Wavelet-based 3D compression scheme for interactive visualization of very large volume data. *Computer Graphics Forum*, 1998. (To appear)
- [6] I. Ihm and S. Park. Wavelet-based 3D compression scheme for very large volume data. In *Proceedings of Graphics Interface '98*, pages 107-116, Vancouver, Canada, June 1998.
- [7] M. Levoy. Effective ray tracing of volume data. *ACM Transactions on Graphics*, 9(3):245-261, July 1990.
- [8] S. Muraki. Volume data and wavelet transforms. *IEEE Computer Graphics and Applications*, 13(4):50-56, 1993.
- [9] G. M. Nielson and B. Shriver, editors. *Visualization in Scientific Computing*. IEEE Computer Society Press, 1990.
- [10] P. Ning and L. Hesselink. Fast volume rendering of compressed data. In *Proceedings of IEEE Visualization '93*, pages 11-18, October 1993.
- [11] NLM, [http://www.nlm.nih.gov/research/visible/visible\\_human.html](http://www.nlm.nih.gov/research/visible/visible_human.html), 1998.
- [12] S. Park, K. Koo, and I. Ihm. Wavelet-based 3D Compression schemes for the visible human dataset and their applications. In *Proceedings of Visible Human Project Conference '98*, Maryland, USA, October 1998. (To appear)
- [13] T. M. Rhyne, editor. *Visualizing and examining large scientific data sets: a focus on the physical and natural sciences*. ACM SIGGRAPH '94 Course Notes, 1994.
- [14] L. Rosenblum. editor. *Scientific Visualization: Advances and Challenges*. IEEE Computer Society Press, 1994.
- [15] P. Schroder and W. Sweldens, editors. *Wavelets in Computer Graphics*. ACM SIGGRAPH '96 Course Notes, 1996.
- [16] J. M. Shapiro. Embedded image coding using zerotree of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445-3462, December 1993.
- [17] E. Stollnitz, T. DeRose, and D. Salesin. *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann Publisher Inc., 1996.
- [18] U. Tiede, T. Schiemann, and K. H. Hohne. Visualizing the visible human. *IEEE Computer Graphics and Applications*, 16(1):7-9, 1996.