

삼차원 모델의 기하학 정보 부호화 방법

호요성^o, 안정환

광주과학기술원 정보통신공학과

Geometry Coding for 3D Model Representation

Yo-Sung Ho^o, Jeong-Hwan Ahn

Kwangju Institute of Science and Technology

hoyo@kjist.ac.kr, jhahn@gogh.kjist.ac.kr

요약

복잡한 실제 사물을 가상 공간상에 표현하기 위해 삼차원 모델의 이용이 최근 급증하고 있다. 기존의 MPEG-1 과 MPEG-2 표준에서는 자연적인 오디오와 비디오 데이터를 효율적으로 저장하거나 전송하는데 역점을 두었지만, 현재 진행중인 MPEG-4 표준화 작업의 SNHC 그룹에서는 자연 영상 뿐만 아니라, 컴퓨터 그래픽으로 만든 삼차원 물체를 효과적으로 표현하는데 중점을 두고 있다. 본 논문에서는 삼차원 모델의 기하학 정보를 효과적으로 부호화하는 방법을 제안하였다. 우선 삼차원 물체의 Vertex Spanning Tree 에 따라 꼭지점 좌표값들의 선형 예측을 통해 얻은 예측 오류를 (r, θ, ϕ) 의 구 좌표계로 표현한다. 이때 r 값은 LMS Fitting 방법을 이용하여 Rayleigh 분포로 근사화하여 최적 양자화를 수행하고, (θ, ϕ) 는 양자화된 r 값에 따라 적응적으로 구의 표면적을 분할하여, 양자화 오류가 r 값과 관계없이 균일하도록 부호화한다.

1. 서론

디지털 비디오와 오디오 신호를 효율적으로 전송하거나 처리할 수 있는 MPEG-1 과 MPEG-2 국제 표준이 성공적으로 개발되어 널리 이용됨에 따라, 인간의 생활을 보다 편리하고 다양하게 만들어 줄 멀티미디어 시대가 열리고 있다. 이러한 멀티미디어 기술은 우리 사회 전반에 걸쳐 큰 변혁을 일으키고 있다. 예를 들면, 기존에는 서로 구분되어 사용되던 컴퓨터와 통신, 방송, 가전의 여러 다중 매체들도 멀티미디어 기술의 등장과 더불어 서로 융합되어 복합적인 산업 형태로 변화하고 있다. 이러한 변화의 영향으로 영상 서비스에 대한 기술 개발도 영상 정보의 효율적인 전달 및 저장이라는 기존의 동영상 압축 표준이 갖는 기능 뿐만 아니라, 인터넷상에서 WWW(World Wide Web) 서비스의 급증에 따른 엄청난 양의 멀티미디어 정보를 처리할 수 있는, 다양한 형태의 창조형 멀티미디어 서비스가 창출되고 있다.

MPEG-1 과 MPEG-2 표준은 CD-ROM 이나 디지털 텔레비전과 같은 실시간적인 비디오와 오디오 신호의 부호화에 초점을 두고, 자연적인(Natural) 비디오와 오디오 정보를 일정한 프레임율로 저장하거나 전송하는 데에만 중점을 두고 있다. 하지만 최근에는 비디오와 오디오 부호화 기술과 컴퓨터 그래픽 기술이 점차 융합되어, 카메라로 입력된 자연 영

상과 컴퓨터에 의해 생성된 합성 영상이 혼합된 복합 영상에 대한 영상 서비스의 요구가 증대되고 있다. 또한 인터넷상에서의 멀티미디어 서비스가 보편화됨에 따라, 삼차원 컴퓨터 그래픽 정보를 네트워크를 통해 송수신하는 공동 설계 작업, 원격 게임, 원격 교육 등과 같은 대화형 그래픽 응용 서비스가 점차 증가하고 있는 추세이다.

따라서 새로운 멀티미디어 요구에 부합하고자, MPEG-4 표준화 작업의 SNHC(Synthetic and Natural Hybrid Coding)[1] 그룹에서는 기존의 표준에는 없던 삼차원 영상과 가상 현실 등 자연 영상과 합성 영상의 표현에 그 중점을 두고 있다. 이러한 혼합 영상의 처리 기법은 영화, 만화, CAD, 컴퓨터 그래픽, 의료 영상, 디자인, 가상 현실 등의 다양한 멀티미디어 분야에서 널리 응용될 수 있을 것이다.

삼차원 물체는 보통 삼각형 Mesh 구조를 이용하여 표현하는데, 이것을 크게 연결성(Connectivity) 데이터, 기하학(Geometry) 데이터, 그리고 광도 측정(Photometry) 데이터로 분류할 수 있다. 연결성 데이터는 주어진 점들이 어떻게 연결되어 Polygon 을 구성하는지에 대한 정보를 나타내며, 기하학 데이터는 삼차원 물체를 구성하는 각 점들의 위치 정보로서 보통 32 Bit 의 유동 소수점으로 표현한다. 그리고 광도 측정 정보는 삼차원 모델의 표면에 Rendering 하거나 Shading 하기 위한 정보로서, 각각의 점에 대한 색깔, 표면의 Normal Vector 성분, Texture 좌표축

등의 정보를 포함한다.

실제 주어진 물체를 표현하는데 필요한 모델의 삼각형 수는 보통 10^5 에서 10^7 정도로 그 데이터의 양이 엄청나게 많다. 따라서 이러한 삼차원 데이터를 그대로 전송하거나 저장, 또는 Rendering 하는데 많은 어려움이 따른다. 그러므로 이러한 응용에 삼차원 데이터를 효율적으로 압축하여 전송하거나 저장하는 방법이 매우 필요하다.

미국의 IBM 에서는 제안한 방법[2]에서는 우선 각 꼭지점의 좌표값을 좌표축 별로 균등 양자화 (Uniform Quantization)를 적용하고, 양자화된 값들을 이용하여 삼차원 물체의 Vertex Spanning Tree 에 따라 선형 예측을 수행한다. 이때 예측된 값과 실제 값의 차이를 엔트로피 부호화한다. 이 방법은 균등 양자화를 적용하는 최대값과 최소값의 차이가 크므로, 전체적으로 큰 양자화 오류를 가진다.

본 논문에서는 DPCM 을 이용하여 각 꼭지점 좌표 (X,Y,Z)의 선형 예측값을 구하고, 그 예측 오류를 (r, θ, ϕ) 의 구 좌표계로 변환시킨다. 이렇게 변환된 값들은 각각 그들의 통계적인 특성을 이용하여 적용적으로 양자화한다. 여기서 r 은 LMS Fitting 알고리즘을 이용하여 Rayleigh 분포로 근사화시켜 최적 부호화를 수행하였고, (θ, ϕ) 는 양자화된 r 의 값에 따라 적용적으로 구의 표면을 분할하여 부호화하였다. 또한 VRML 데이터에 대한 실험을 통해 기존의 방법과 제안된 방법의 성능을 비교하였다.

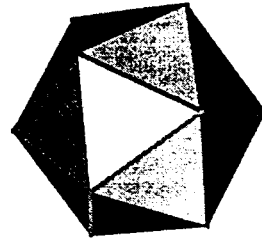
2. 연결성 정보의 부호화

2.1 Topological Surgery

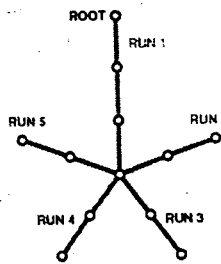
Topological Surgery 방법은 IBM[2]에서 처음 제안한 방법으로서, 그림 1에서 나타낸 것과 같이, 삼차원 Mesh를 Vertex Spanning Tree와 Triangle Spanning Tree를 이용하여 표현한다.

처음에 삼차원 Mesh의 Cut Edge를 따라 삼차원 물체를 자르고, 잘려진 Edge에 존재하는 꼭지점들이 그림 1 (b)와 같이 Vertex Spanning Tree의 Node가 된다. 이때 각 Node는 Child Node를 가지고 있는지의 여부에 따라 Branch Node가 결정되며, Terminal Node인지 아닌지에 따라 Leaf Node로 분류된다. 그림 1의 경우에는 1개의 Branch Node가 있고, 4개의 Leaf Node, 1개의 Root Node가 있다. 여기서 Root Node란 각 Tree의 시작 Node를 의미한다. 하나의 Branch Node에서 나뉘어진 각각의 꼭지점들을 Vertex Run이라 한다.

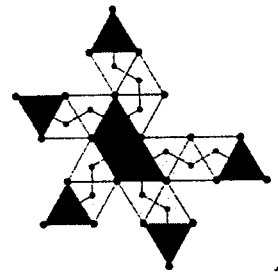
Triangle Spanning Tree는 그림 1 (c)에 나타내었다. Triangle Spanning Tree에서 각 Node는 하나의 삼각형 면을 가리킨다. 여기서 각 Run은 Run의 길이와 Leaf Node의 여부, 그리고 다음 삼각형이 오른쪽에 연결되어 있는지 왼쪽으로 연결되어 있는지를 나타내는 Marching Bit로 표현된다.



(a) 원래의 삼차원 모델



(b) Vertex Spanning Tree



(c) Triangle Spanning Tree

그림 1. 삼차원 모델과 Spanning Trees

그림 1 (b)의 Vertex Spanning Tree에서 각 꼭지점들은 Root Node로부터 하나의 경로 위에 있으며, 각 꼭지점의 좌표값들은 보통 16 비트로 표현된다. Triangle Spanning Tree에서 꼭지점들은 인접 꼭지점들과 서로 상관성을 가지고 있으므로, 이런 성질을 이용하여 다양한 압축 기법을 적용할 수 있다. 이때 전송되는 정보들은 다음과 같다.

- (1) VTREE: Vertex Spanning Tree Table (Run 길이, 가지 비트, 잎 비트로 구성됨)
- (2) VCOR: 과거의 Node 값을 이용하여 예측한 값과 실제 현재 값과의 차이를 부호화한 꼭지점 좌표값
- (3) TTREE: Triangle Spanning Tree Table (Run 길이, 잎 비트로 구성됨)
- (4) MARCH: Triangle Tree의 Marching 유형

그리고 고정된 크기를 가진 부가 정보가 더해진다

2.2 Arithmetic Encoding

위에서 설명한 Spanning Tree의 정보를 전송하기 위해서는 EPFL에서 제안한 Arithmetic Coding[5]을 이용하여 부호화한다.

가. Vertex Spanning Tree의 부호화 방법

Vertex Spanning Tree는 Vrun, Vleaf, Vchild, Loopstart, Loopend 등의 이진수 String 정보를 가진다. 만일 Tree가 Loop를 가지면 Loopmap이 필요하다. Vertex Spanning Tree에는 Run Node, Branch Node, Leaf Node 등 3가지 종류의 Node가 있다. 이 Node들은

Child Node 의 갯수로 분류된다. 즉, Leaf Node 는 Child Node 가 없으며, Run Node 는 하나의 Child Node 를 가지고 있고, Branch Node 는 2 개나 혹은 그 이상의 Child Node 를 가진다.

Root Node 로 시작하여 DFS (Depth First Search) 순서대로 Tree 에 있는 모든 꼭지점과 Edge 를 탐색한다. Branch Node 에서는 반시계 방향으로 탐색한다. 어떤 Edge 들은 탐색하는 동안 하나의 Loop 를 형성할 수 있다. 이러한 Edge 를 Loop Edge 라고 한다. 어떤 임의의 탐색된 Z Node 에 대해 다음과 같은 방법을 이용하여 5 개의 String 을 갱신한다. 여기서 e 는 Z 에 도달하기 위해 탐색된 Edge 를 뜻한다.

- (a) Z 가 이미 탐색된 Node 라면, 즉, e 가 Loop 를 형성하면, Vrun 에 0 을 첨가하고, Vleaf 와 Loopend 에는 1 을 첨가한다.
- (b) Z 가 Run Node 라면, Vrun 에 1 을 첨가한다.
- (c) Z 가 Leaf Node 라면, Vrun 에 0 을 첨가하며, Vleaf 에 0 을 첨가한다. 그리고 Z 의 Children 의 갯수에서 2 를 뺀 갯수만큼 0 을 Vchild 에 첨가하고 뒤에 1 을 추가한다. Z 의 각 Branch 에 대해서, 만일 Branch 가 Incoming Loopedge 를 가지고 시작하면 1 을 Loopstart 에 더하고, 그렇지 않다면 0 을 첨가한다.

Loopend 와 Loopstart 는 Loopedge 의 Edge List 를 정의하는데, Loopmap 은 Loopend 부터 Loopstart 까지의 Edge List 를 저장한다. 각 String 에 포함되는 0 과 1 의 갯수는 표 1 과 같다.

표 1. Vertex Spanning Tree 에서 0 과 1 의 발생 빈도

String	0 의 갯수	1 의 갯수
Vrun	Nvruns	nvertices + nvloops - nvruns
Vleaf	nvruns - nvleaves	nvertices
Vchild	nvruns - nvleaves	2 nvertices - nvruns + nvloops - 1
Loopstart	nvleaves - 1	nvloops
Loopend	nvleaves - nvloops	nvloops

나. Triangle Spanning Tree 의 부호화 방법

Triangle Spanning Tree 는 Trun, Tleaf, Tmarching, Polygonedge 의 4 개의 이진 String 을 가진다. Triangle Spanning Tree 에서 Node 는 Run Node, Leaf Node, Branch Node 의 3 종류가 있다. 만일 Node 가 하나의 Child Node 를 가지면 Run Node 라 하고, Child Node 가 없다면 Leaf Node 라 하고, 2 개의 Child Node 를 가지면 Branch Node 라 한다.

Root Node 에서 DFS 순서대로 Triangle Spanning Tree 을 탐색한다. 탐색되는 Node Z 에 대하여, 다음과 같이 4 개의 이진수 String 을 갱신한다.

- (a) 만일 Mesh 가 삼각형이 아니면, 그 Mesh 를 삼각형으로 만들고 새롭게 첨가되는 Edge 를 Polygon Edge 라 하며, Polygonedge String 에 1 을 첨가한다.
- (b) Z 가 Leaf Node 이면, Trun 에 0 을 첨가하고, Tleaf 에 1 을 첨가한다.
- (c) Z 가 Branch Node 이면, Trun 에 0 을 첨가하고, Tleaf 에 0 을 첨가한다.
- (d) Z 가 Run Node 이면, Trun 에 1 을 첨가한다. 만일 Z 가 왼쪽 자식을 가지면, Tmarching 에 0 을 첨가하고, 오른쪽 자식을 가지면 1 을 첨가한다.

각 String 에 포함되는 0 과 1 의 갯수는 표 2 와 같다.

표 2. Triangle Spanning Tree 에서 0 과 1 의 발생 빈도

String	0 의 갯수	1 의 갯수
Trun	2 nbranches + 1	ntriangles - 2 ntbranches - 1
Tleaf	nbranches	ntbranches + 1
Tmarching	ntleftchild	ntriangles - 2 ntbranches - ntleftchild - 1
Polygonedge	ntriangles - npolyedges	npolygonedges

다. String 의 부호화 방법

위에서 설명한 방법대로 Spanning Tree 를 부호화 하면, 각 String 들은 표 1 과 표 2 에서 정의한 갯수만큼 a 개의 0 과 b 개의 1 을 가진다. 각 String 을 산술 부호화하고 복호화하기 위해서, 처음에 각 String 에 대한 확률을 다음과 같이 정의한다.

$$p(0) = \frac{a}{a+b}, \quad p(1) = \frac{b}{a+b}$$

만일 처음에 0 이 부호화되면 a 의 갯수를 하나 감소시키고, 1 이 부호화되면 b 의 갯수를 감소시킨다. 이렇게 해당되는 0 이나 1 의 갯수를 감소시키고 나서, 위에서 정의한 확률 모델을 다시 계산한다. 이러한 과정을 통하여 확률 분포를 갱신하면서, 각 String 을 산술 부호화한다.

Tmarching 과 Polygonedge 의 String 에 대해서는 4 가지 경우를 생각한다. 즉, 0 다음에 a 개의 0 이 나오는 경우, 0 다음에 b 개의 1 이 나오는 경우, 1 다음에 c 개의 0 이 나오는 경우, 1 다음에 d 개의 1 이 나오는 경우를 고려하여 다음과 같이 확률을 정의한다.

$$p(0 | \text{previous} = 0) = \frac{a}{a+b}, \quad p(1 | \text{previous} = 0) = \frac{b}{a+b}$$

$$p(0 | \text{previous} = 1) = \frac{c}{c+d}, \quad p(1 | \text{previous} = 1) = \frac{d}{c+d}$$

각 String의 Symbol을 부호화하고 나서, 위에서 설명한 것처럼, 확률 모델을 계속 갱신하면서 산술 부호화한다.

3. 기하학 정보의 부호화

본 논문에서 제안한 방법의 블록도를 그림 2에 제시하였다. 먼저 전처리 단계에서는 입력 값들에 대해 좌표축 별로 예측 오류에 대한 평균과 분산을 계산한다. 여기서 계산된 값을 가지고 부호화 블록에서는 좌표축 별로 예측 오류를 정규화한 후, 이를 구 좌표계로 변환한다. 그리고 r, θ, φ 별로 각각의 통계적인 특성을 고려하여 적응적으로 양자화한 후, Huffman 부호화기를 이용하여 엔트로피 부호화한다.

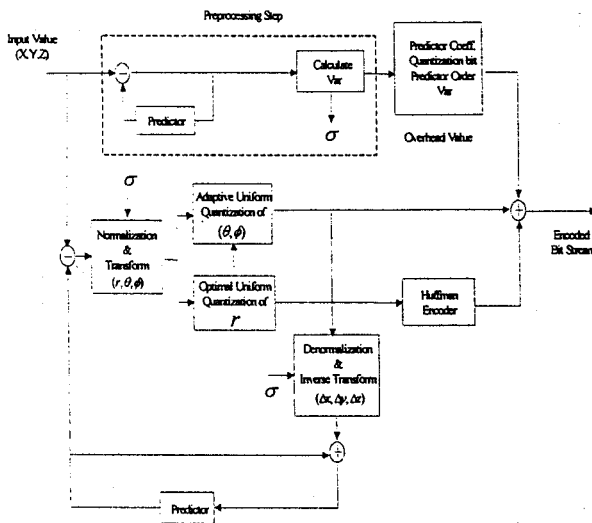


그림 2. 제안된 삼차원 모델의 부호화기

3.1 선형 예측 방법

삼차원 모델의 각 꼭지점 $v_n = (x_n, y_n, z_n)$ 의 예측 값 $\overline{v_n} = (\overline{x_n}, \overline{y_n}, \overline{z_n})$ 은 IBM에서 제안한 Vertex Spanning Tree에서 정의된 꼭지점들에 대한 Farther-Son 관계에 따라 식(1)처럼 선형 예측할 수 있다.

$$\overline{x_n} = \sum_{i=1}^p \lambda_{xi} x_{n-i}, \quad \overline{y_n} = \sum_{i=1}^p \lambda_{yi} y_{n-i}, \quad \overline{z_n} = \sum_{i=1}^p \lambda_{zi} z_{n-i} \quad (1)$$

여기서 p는 예측 차수이며, $\lambda_{xi}, \lambda_{yi}, \lambda_{zi}$ 는 예측 계수이다. 예측 계수의 값은 Levinson-Durbin 알고리즘[4]을 이용하여 X, Y, Z 별로 구해진다. 그리고 예측 오류 $(\Delta x_n, \Delta y_n, \Delta z_n)$ 는 식(2)처럼 구한다.

$$\Delta x_n = x_n - \overline{x_n}, \quad \Delta y_n = y_n - \overline{y_n}, \quad \Delta z_n = z_n - \overline{z_n} \quad (2)$$

3.2 예측 오류의 정규화 및 구 좌표 변환

좌표값에 대한 예측 오류 $\Delta x, \Delta y, \Delta z$ 가 구해지면, 전처리 과정에서 얻은 평균과 분산으로 예측 오류를 평균은 0이고 분산이 1인 분포로 정규화한다. 정규화된 각 예측 오류 벡터 $(\Delta x, \Delta y, \Delta z)$ 는 식 (3)의 관계를 이용하여 구 좌표계로 변환된다

$$\begin{aligned} r &= \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \\ \theta &= \tan^{-1} \Delta y / \Delta x \\ \phi &= \cos^{-1} \Delta z / r \end{aligned} \quad (3)$$

이때 정규화된 각각의 예측 오류 $\Delta x, \Delta y, \Delta z$ 는 일반적으로 Laplacian 분포를 가진다. 그러나 반지름 r은 식 (3)과 같은 Chi-square 분포의 형태를 가진다. 따라서 r의 분포를 최적으로 양자화하기 위해서는 r의 분포를 정확하게 알아야 한다. 본 논문에서는 r의 분포를 LMS (Least Mean Square) Fitting 방법을 적용하여 식 (4)와 같이 근사화하였다.

$$f_r(r) = r e^{-\alpha r^2}, \quad r \geq 0 \quad (4)$$

그림 3에서 보는 것처럼, $\alpha = 0.372$ 일 때 식 (4)는 원래 r의 분포와 거의 일치함을 알 수 있다.

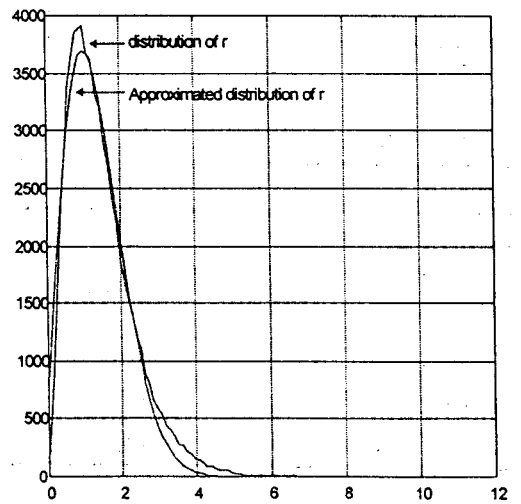


그림 3. r의 분포와 근사화

표 3은 식 (4)를 이용하여 설계한 최적 균등 양자화(Optimal Uniform Quantization) 간격을 나타낸다.

표 3에서 구한 양자화 Step Size를 이용하여 반지름 r를 최적 균등 양자화하고, 양자화 값에 대한 Index는 Huffman 부호화기를 이용하여 엔트로피 부호화한다. 여기서 양자화 크기와 분포는 고정되어 있기 때문에 Huffman 테이블과 양자화 크기를 수신단에 따로 전송할 필요는 없다.

표 3. Optimum Uniform Quantizer 의 Step Size

No. of Output Levels	Step Size
2	2.9061
4	1.4123
8	0.8125
16	0.4505
32	0.2476
64	0.1349
128	0.0729
256	0.0390
512	0.0208
1024	0.0110
2048	0.0058
4096	0.0030

3.3 r 과 (θ,φ)의 관계

구 좌표계에서는 보통 반지름 r과 (θ,φ)의 각도로 하나의 점을 표시하는데, r은 식 (4)의 형태로 Chi-square 분포를 가진다. 본 논문에서는 (θ,φ)를 최적으로 부호화하기 위하여 r에 기반을 둔 적응적 양자화 방법을 제안하였다.

일반적으로 구 좌표계에서는, 그림 4에 나타난 것과 같이, 반지름 r이 커지면 같은 각도에 대해서 구의 표면적 S는 r²에 비례하여 커진다.

만일 반지름 r인 구의 표면적을 r² 부분으로 분할하고, 구 표면에 r²개의 양자화 점을 균등하게 분포시킨다. 만약 반지름 r이 1/2로 작아지면, 표면적은 1/4로 줄어들므로, 양자화 점의 개수도 1/4로 줄어든다. 반대로, r이 2배로 커지면, 표면적은 4배로 커지므로, 양자화 점의 개수도 4배로 늘어난다. 이때 전체적인 양자화 평균 오류는 r의 크기와 관계 없이 거의 비슷해진다.

따라서 똑같은 양자화 오류를 얻기 위해서 r이 작을 때는 (θ,φ)에 적은 비트를 할당하고, r이 클 때에는 (θ,φ)에 상대적으로 많은 비트를 할당한다. 그런데 일반적으로 r의 분포는, 그림 3에 보인 것처럼, 대부분 0 부근에 많이 몰려 있으므로, 통계적인 관점에서 전체적인 부호화 비트수가 줄어들게 된다.

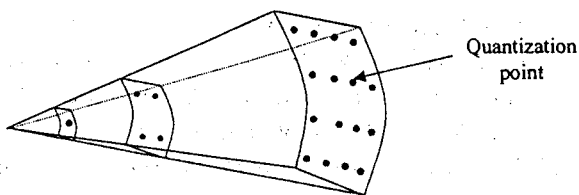


그림 4. 구의 반지름과 표면적

3.4 (θ,φ)의 적응적 부호화 방법

구 표면의 모든 점들은 x 축과의 각도인 θ와 z 축과의 각도 φ를 이용하여 나타낼 수 있다. (θ,φ)를 양자화하기 위해서, 처음에 구의 표면을 그림 5와 같이, 각 좌표축의 방향에 따라 8등분으로 나눈다. 따라서 각 분할된 면을 나타내기 위해 3비트가 필요하다. 이때 8등분으로 나눈 구 표면의 넓이는 같으므로, 1사분면에 있는 하나의 면만을 고려한다.

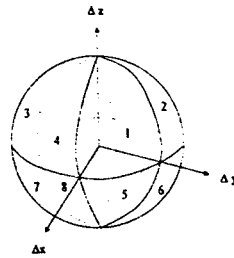
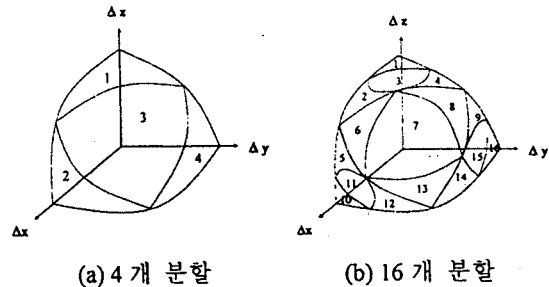


그림 5. 구 표면의 8분할 방법

양자화 비트와 양자화된 r의 크기가 증가하면 하나의 면은, 그림 6과 같이, 같은 면적을 가진 4개의 삼각형 형태의 모양으로 분할한다.



(a) 4개 분할 (b) 16개 분할

그림 6. 구 표면의 미세 분할 방법

4. 실험 결과

본 논문에서는 현재 3차원 데이터의 처리를 위해 널리 사용되고 있는 VRML (Virtual Reality Modeling Language)[4] 모델에 대해서 C언어를 이용하여 제안한 방법들을 구현하였다. 실험에 사용한 데이터는 EIGHT, FEMUR, HORSE, BEETHOVEN이며, 각 모델의 특징을 표 4에 정리하였다.

표 4. 실험에 사용한 삼차원 모델

모델 이름	꼭지점의 수	삼각형의 수
EIGHT	766	1535
FEMUR	3897	7798
HORSE	11135	22258
BEETHOVEN	2845	2812

꼭지점들의 Father-Son 관계를 알기 위하여 IBM 에서 만든 Vertex Spanning Tree 를 이용하였다. 그리고 복원한 영상과 원래 영상과의 오류를 구하기 위하여 다음의 방법을 사용하였다. 압축해서 복원한 모델의 표면에 일정한 수의 점을 찍고, 원래 모델의 같은 표면에서의 점들과의 평균 거리를 오류로 정의하였다. 그림 7 은 제안된 방법과 IBM 방법과의 결과를 비교한 것이다. 여기서 X 축은 꼭지점당 비트를 나타내며, Y 축은 모델의 오류를 나타낸다. 그림 7 에서 보듯이, 제안된 방법이 IBM 방법보다 같은 BPV(Bits Per Vertex)에 대해서 오류가 작으므로 성능이 더 좋은 것을 알 수 있다.

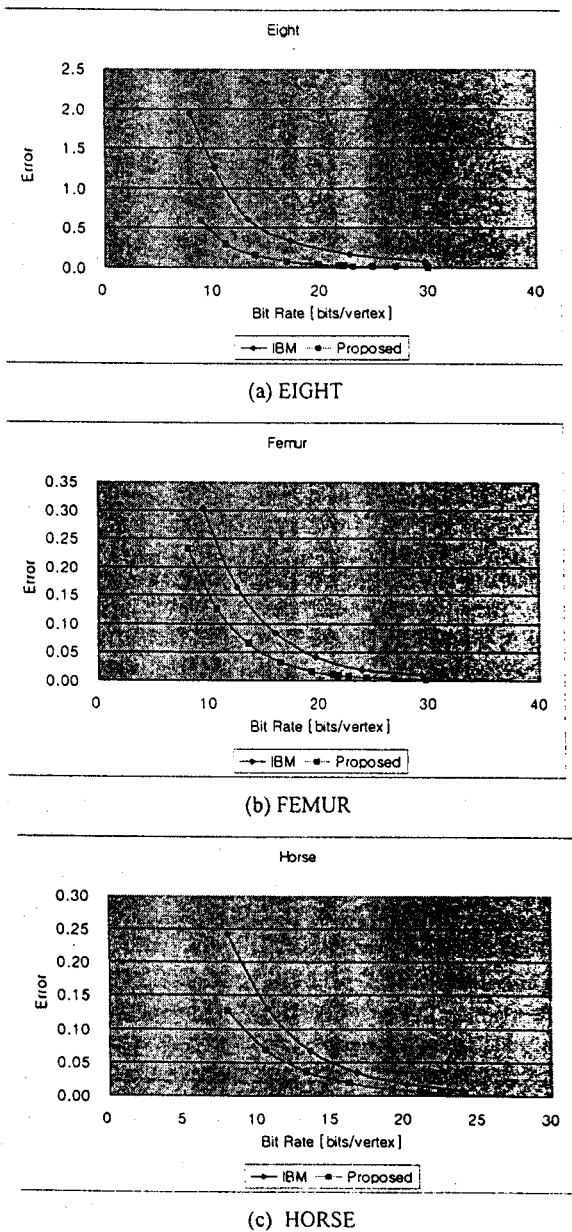
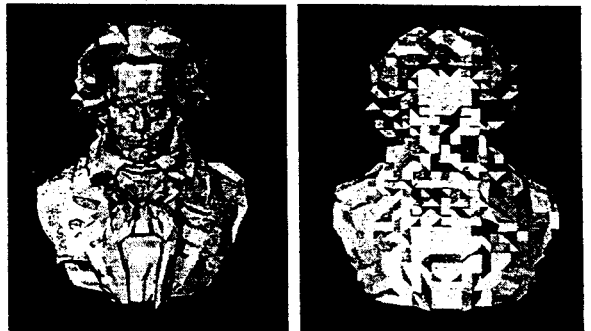


그림 7. 실험 결과 비교



(a) 제안한 방법 (b) IBM 방법

그림 8. 복원된 Beethoven 모델 (8 BPV)

그림 8 은 주관적인 평가를 위하여, 꼭지점당 8 Bit 를 이용하여 복원한 Beethoven 모델을 비교한 것이다. 일반적으로 삼차원 모델의 꼭지점은 넓은 범위에 불규칙적으로 분포되어 있다. 따라서 IBM 에서 제안한 방법처럼 예측 과정 전에 원래 꼭지점들을 균일하게 양자화하면, 양자화 Step 의 크기가 아주 크게 된다. 이 경우, 양자화 과정에서 같은 양자화 범위 안에 있는 모든 꼭지점들은 똑같은 양자화 값으로 표현되므로, 복원된 모델에서 여러 개의 꼭지점들이 하나의 꼭지점으로 합쳐져 해상도가 떨어지는 Vertex Merging 문제가 발생할 수 있다. 이것은, 그림 8 (b)에서 볼 수 있듯이, 복원된 모델의 표면을 울퉁불퉁하게 만드는 심각한 문제를 발생시킨다.

5. 결론

본 논문에서는 삼차원 모델의 기하학 정보 부호화에 대한 새로운 방법을 제안하였다. 삼차원 모델의 (X,Y,Z)의 꼭지점 좌표값을 (r,θ,φ)의 구 좌표계로 변환하고, r 에 대한 통계적인 특성을 이용하여 최적으로 부호화하였다. (θ,φ)는 구 표면의 분할 방법을 이용하여 r 에 관계없이 오류가 일정하도록 양자화하였다. 실험 결과에서 보듯이, 제안된 방법이 넓은 범위의 비트율에서 IBM 방법보다 평균 오류가 작았으며, 주관적인 평가도 우수하였다.

6. 참고문헌

- [1] Ad hoc group on MPEG-4 SNHC, "MPEG-4 SNHC Verification Model 9.0", ISO/IEC JTC1/SC29/WG11 MPEG98/M3809, July 1998.
- [2] G. Taubin and W. Horn, F. Lazarus, J. Rossignac, "Geometric Coding and VRML," ISO/IEC JTC1/SC29/WG11 MPEG98/M3061, Feb. 1997.
- [3] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.
- [4] J. Hartman and J. Wernecke, *The VRML 2.0 Handbook*, Addison-Wesley, 1996.