삼차원 모델의 점진적인 렌더링과 오류 강인을 위한 효율적인 데이터 분할 방법 (CODAP)

송문섭*, 안정환**, 김성진*, 한만진*, 호요성**
*삼성종합기술원 신호처리연구실 **광주과학기술원 정보통신공학과
전화: (0331) 280-9245 / 팩스: (0331) 280-9207

Data Partitioning for Error Resilience and Incremental Rendering of 3D Model

MunSup Song*, Jeong-Hwan Ahn**, Sung Jin Kim*, Mahnjin Han*, Yo-Sung Ho**
*Samsung Advanced Institute of Technology **Kwangju Institute of Science and Technology
E-mail: mssong@sait.samsung.co.kr E-mail: jhahn@kjist.ac.kr

ABSTRACT

Applications using 3D models are increasing recently. Since 3D polygonal models are structured by a triangular mesh, the coding of polygonal models in strips of triangles is an efficient way of representing the data. These strips may be very long, and may take a long time to render or transmit. If the triangle strips are partitioned, it may be possible to perform more efficient data transmission in an error-prone environment and to display the 3D model progressively. In this paper, we devised the Component Based Data Partitioning (CODAP) which is based on Topological Surgery (TS). In order to support the error resilience and the progressively build-up rendering, we partition the connectivity, geometry, and properties of a 3D polygonal model. Each partitioned component is independently encoded and resynchronization between partitioned components is done.

1. 서론

최근 실제 사물을 가상 공간상에서 표현하기 위해 복잡한 삼차원 물체를 이용한 응용이 늘어나고 있다. 기존의 MPEG-1 과 MPEG-2 표준은 자연적인 오디오와 비디오 데이터를 저장하거나 전송하는데 역점을 두었지만, MPEG-4 Synthetic and Natural Hybrid Coding (SNHC) 에서는 컴퓨터 그래픽으로 만든 삼차원 물체를 이용하여 좀더 효과적이고 상호작용 하는 영상을 표현하는데 중점을 두고 있다.

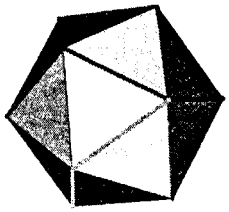
일반적으로 삼차원 물체는 삼각형 메쉬 구조(Mesh)로 나타내는데, 이것은 연결성(Connectivity) 정보, 기하학(Geometry) 정보, 색(Color)과 법선(Normal) 벡터, 그리고 텍스쳐(Texture) 정보 등으로 이루어져 있다. 연결성 정보는 삼각형 메쉬들이 어떻게 연결되었는지에 대한 정보를 가지고 있고, 기하학(Geometry) 정보는 삼각형 메쉬의 꼭지점 좌표값을 가지고 있다. 그리고 색, 법선 벡터, 텍스쳐 정보는 삼차원 물체를 렌더링하는데 필요한 정보를 가지고 있다.

기존의 삼차원 메쉬 데이터에 대한 부호화[1,2]는 메쉬 데이터 전체에 대하여 부호화 함으로써, 부호화된 데이터의 전송 시 전체의 비트열을 모두 받기 전에 일부를 복원하는 것이 불가능하다. 즉, 모든 비트열을 다 받아야만 부호화된 삼차원 모델을 복원할 수 있다. 또, 전송 시 발생하는 선로상의 오류는 극히 일부의 데이터를 손실시켜도 전체를 다시 받아야 하는 비효율적인 측면이 있다. 따라서, 에러가 있는 채널상에서 데이터를 보내기 위해, 보통 하나의 비트열을 여러 개의 단위로 분할하고, 분할된 단위별로 부호화하여 전송한다. 하나의 분할 단위에 에러가 삽입된다면 전체 비트열을 버리는 것이 아니라, 에러가 있는 분할 단위만을 버리고 나머지 비트열은 복원하는 것이다. 따라서 복호기는 송신단에 잃어버린 분할 단위만을 요구하여 다시 전송받고 원래의 삼차원 모델을 복원할 수도 있다. 본 논문에서는 이러한 삼차원 모델의 점진적인 전송과 오류가 있는 채널상에서 강인하도록 Component 에 기반을 두고 부호화하는 Component Based On Data Partitioning (CODAP) [3]을 제안한다.

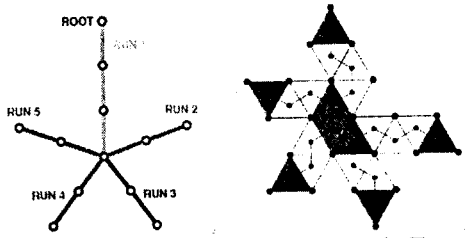
2. Component Based Data Partitioning

2.1 Topological Surgery

Topological Surgery(TS) 방법[1]은 IBM 에서 처음 제안한 방법으로서, 그림 1 에서 나타낸 것과 같이, 삼차원 메쉬를 Vertex Graph(VG)와 Triangle Tree(TT)를 이용하여 표현한다.



(a) 원래의 삼차원 모델



(b) Vertex Graph (c) Triangle Tree

그림 1. 삼차원 모델과 Spanning Trees

처음에 삼차원 메쉬 모델의 Cut Edge 를 따라 삼차원 물체를 자르고, 잘려진 Edge 에 존재하는 꼭지점들이 그림 1 (b)와 같이 VG 의 노드가 된다. 이때 각 노드는 자식 노드를 가지고 있는지의 여부에 따라 가지 노드가 결정되며, 끝 노드인지 아닌지에 따라 잎 노드로 분류한다. TT 는 그림 1 (c)에 나타내었다. TT 에서 각 노드는 하나의 삼각형 면을 가리킨다. 여기서 각 런(Run)은 런의 길이와 잎 노드의 여부, 그리고 다음 삼각형이 오른쪽에 연결이 되어 있는지 왼쪽으로 연결되어 있는지를 나타내는 Marching 비트로 표현된다.

TS 방법에서 원래의 모델을 복원하기 위해서는 먼저 VG 를 이용하여 Bounding Loop 를 생성한다. TT 에서 분기하는 삼각형의 제 3 의 꼭지점을 Y-vertex 라고 하며, TT 의 비트열을 이용하여 Y-vertex 를 계산하고, 스트립(Strip) 형태의 일련의 연결성을 갖는 삼각형을 구성한다. 따라서, TS 방법에서는 모든 비트열이 들어와 있어야 복원이 가능하기 때문에 전송 중 에러가 생겼을 경우, 전체 비트열을 다시 전송해야 하는 비 효율성이 있다. 또 복원한 메쉬 데이터가 큰 경우, 전체 데이터를 모두 전송 받고 이를 복원하거나 렌더링하는 동안 사용자는 무한정 대기하고 있어야 한다. 따라서 이러한 문제점을 해결하기 위해서 메쉬 데이터를 전송로의 대역폭이나 복호기의 특성을 고려하여 효율적인 단위 크기로 작게 나누고, 이 단위 비트열을 처리 단위로 복호 및 렌더링 할 수 있어야 하며, 오류에 대한 강인성을 제공해야 한다. 또한 현재 수신된 일부 데이터만으로도 일부 복원과 렌더링까지 가능하도록 점진적 형태의 부호화 및 복호화 방법을 제공해야 한다.

2.2 점진적 전송 및 탐색 방향 결정

위에서 설명한 문제점을 해결하기 위해 Bounding Loop 와 Y-vertex 를 효율적으로 처리하는 방법을 제안한다. TT 에서 가지에 해당하는 Y-vertex 의 위치는 삼각형의 렌더링 및 분할 단위의 크기를 결정하는데 아주 중요하다. 보통 Y-vertex 의 위치는 그림 2 와 같이 가지의 Boundary Offset 을 계산하여 구한다. 그림 2 에서 Branching Triangle 은 b[m-3], b[m-10], b[n+2]로 구성되어 있는데 오른쪽 가지의 삼각형의 개수에 대한 정보를 가지고 Y-vertex 의 Bounding Loop 에서의 인덱스인 m-10 를 구한다.

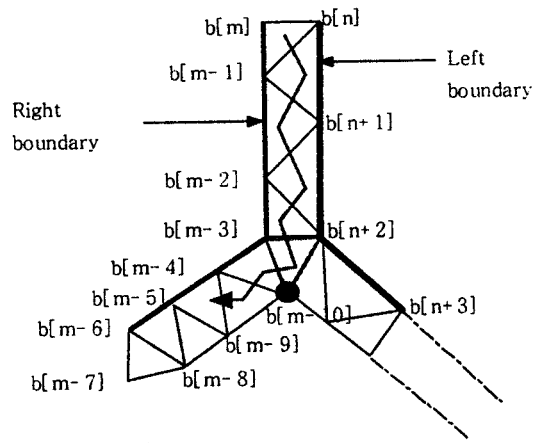


그림 2. Triangle Tree 에서 Bounding Loop 인덱스와의 관계

TS 에서 탐색의 순서는 항상 시계 방향이다. 만일 왼쪽에 있는 가지가 오른쪽 가지보다 길다면 복호기는 왼쪽에 있는 가지를 다 복호할 때까지 기다려야 Y-vertex 의 위치를 알 수 있다. 또한 TT 를 자를 때, 한쪽 가지에 있는 삼각형의 개수가 작거나 크다면, 하나의 분할에 들어가는 삼각형의 개수 또한 균일하지 않을 수 있다. 따라서 짧은 런을 가진 부분부터 탐색을 하여 불필요한 지연을 막고, 분할 단위에 들어가는 삼각형의 개수를 균일하게 한다. 이를 위해 1 비트의 (왼쪽, 오른쪽)플래그 비트를 삽입하여 부호화 순서를 결정한다. 여기서 1 비트의 플래그 비트는 TT 의 깊이에 의해서 결정된다. 그림 3 에서는 5 비트로 순차적으로 빠른 메쉬를 복원할 수 있다. 보통 Triangle Data 는 Triangle March (tm) 정보를 가지고 있다. 그러나 여기에 td_orientation 필드를 삽입하여 탐색순서를 알려준다.

만든다. 가상 비트는 다음의 두 가지 경우에 대해 한다. 분할이 하나의 런, 하나의 잎으로 끝나면, (trun, tleaf)에 가상의 (1,1)을 삽입한다. 분할이 가지에서 끝나면, (trun, tleaf)에 (1,1)을 삽입한다. 만일 분할이 가지에서 끝나면, 기하학 정보와 광도 측정정보는 마지막 Branch Triangle 에 대해서는 부호화하지 않는다. 왜냐하면 이 가지에 대한 Y-vertex 인덱스는 가지와 다음 분할의 루트 삼각형 이전의 오른쪽의 Triangle 에 의해서 구할 수 있기 때문에, 마지막 가지 삼각형은 복원할 수 있다.

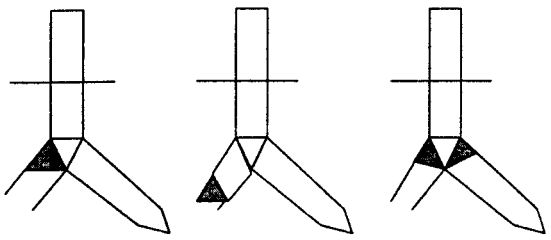


그림 5. 가상 삼각형 (짙은 회색):
(a) 잎 (b) 런 (c) 가지

TT/TD 가 분할 되고 가상 CC 로 형성될 때, 복호기는 다음의 조건을 이용하여 가상 비트를 찾을 수 있다.

right_bloop_idx – left_bloop_idx – 1 > TT 에서 복원한 삼각형의 총 개수

만일 이 조건이 사실이면, 가상 비트가 존재한다. 하나 혹은 두 개의 가상 삼각형을 찾기 위해서는, 끝으로부터 세 번째 삼각형이 가지인지 아닌지를 검사하여 찾는다. 만일 가지이면, 분할은 가지에서 끝나고 두 개의 가상 삼각형이 더해진 것이다. 그렇지 않다면 분할은 잎이나 런에서 끝나고 끝 단에 단 하나의 가상 삼각형이 더해진다.

3. 실험결과

실험에서는 VRML 모델에 대해 제안한 방법을 적용하였다. 실험 데이터의 특징은 표 1 과 같다.

표 1. 실험 데이터의 특징

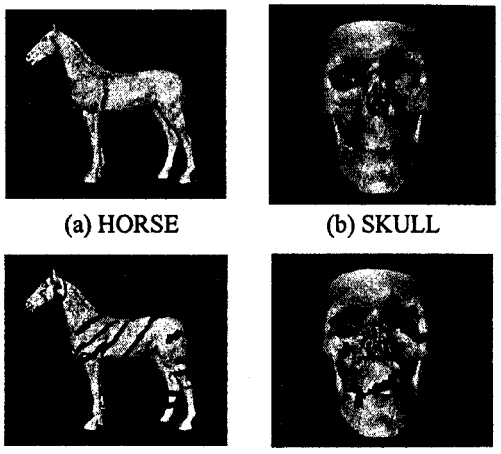
이름	꼭지점 수	다각형의 수
HORSE	11135	22258
SKULL	10952	22104

분할 단위의 크기는 180byte, 360byte, 1440byte 로 하였다. 채널상의 오류는 32000bps 의 비트율에서 0.00005 의 비율로 랜덤 오류가 발생한다고 가정하였다. 표 2 는 이러한 조건에서 복원한 다각형의 비율을 보여준다.

표 2. 랜덤 오류에 대한 실험 결과

모델	분할 길이 (Byte)	복원율 (%)
HORSE	180	88.9
	360	86.9
	1440	52.0
SKULL	180	89.0
	360	76.9
	1440	30.3

그림 6 (a)와 (b)는 원래의 모델이고, (c)와 (d)는 오류가 생긴 분할을 버리고 모델을 복원한 그림이다. 그림에서 보듯이 전송 받은 비트만으로도 모델의 복원이 가능하며, 비트열을 전송 받으면서 복호화와 랜더링이 가능하다.



(a) HORSE (b) SKULL

(c) 복원한 HORSE (d) 복원한 SKULL

그림 6. 복원한 모델

4. 결론

본 논문에서는 Component 에 기반을 두고 삼차원 메쉬 데이터를 효과적으로 분할하여 점진적인 전송과 오류에 강인하도록 하는 방법을 제안하였다. 제안한 방법은 효과적으로 메쉬를 분할하여 부가정보를 최소화할 수 있고, 채널 오류에 강인함을 알 수 있다.

참고문헌

[1] G. Taubin and J. Rossignac, "Geometry Compression through Topology Surgery," ISO/IEC JTC1/SC29/ WG11 MPEG98/M3059, Feb. 1998.

[2] Jeong-Hwan Ahn and Yo-Sung Ho, "Adaptive Opimal Quantization for 3D Mesh Representation using the Spherical Coordinate System", SPIE 1999.

[3] Mahn-Jin Han, Mun-Sup Song, Sung-Jin, Euee S. Jang, and YangSeock Seo, "Results of M5 CE", ISO/IEC JTC1/SC29/ WG11 MPEG98/M4516, Feb. 1999.

tm_1	tm_2	$td\_orientation_2$	...	tm_i	$td\_orientation_i$

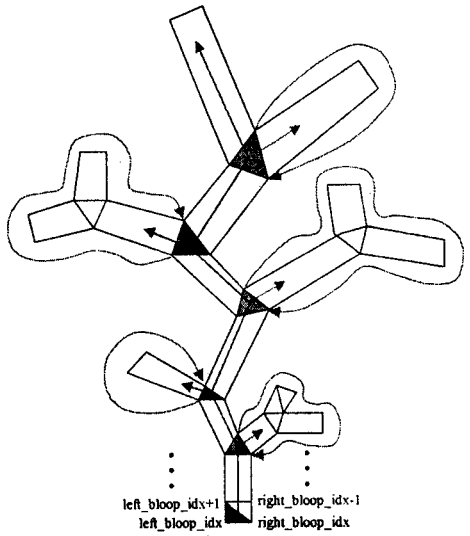


그림 3. 탐색 순서의 결정

2.3 오류 강인을 위한 분할(Partition) 방법

Boundary Loop Table 을 만들 때, Stitch 를 위하여 Vertex Indices 가 필요하다. 만일 이전 분할 단위가 채널의 오류로 인해 깨지거나 잃어버린다면, Vertex Graph 와 Triangle Tree/Triangle Data 에 해당하는 서로의 관계를 알려주기 위한 정보가 필요하다. 따라서 이러한 정보를 주기 위하여 Vertex Graph ID 와 현재 분할 단위에 대한 시작 점으로써 Bounding Loop 의 2 개의 Left/Right Visiting Indices 가 필요하다. 이것을 이용하면 분할에 대한 연결성 정보, 기하학 정보 등을 만들 수 있다.

1) 점진적 표현과 에러 강인

삼차원 모델은 다음과 같이 여러 개의 Connected Component (CC)의 집합으로 이루어져 있다.

Data for cc 1	Data for cc 2	...	Data for cc n

주어진 CC 에 대한 데이터는 다음과 같이 VG, TT, TD 으로 이루어져 있다.

Vertex Graph(VG)	Triangle Tree(TT)	Triangle Data(TD)

여기서 TT 는 tt_run_length 와 tt_leaf 필드로 이루어져 있다. CC 를 분할한 경우 다음의 구조로 이루어져 있다.

vg	vi # 1	bp # 1	tt # 1	td # 1	...

첫 번째 분할에서, CC 의 vg 는 위의 그림과 같다. 모든 파티션에 대한 vg 는 앞에 모아서 한꺼번에 보내거나 따로 따로 보낼 수도 있다. 그리고 Visiting Indices (vi), Boundary Prediction (bp)이 뒤따른다. vi 는 시작하는 좌우 Bounding Loop 의 인덱스를 가지고 있다. bp 필드는 파티션 사이에 기하학 정보와 특징 정보 사이의 예측에 대한 모드를 가지고 있다. 즉, 데이터를 분할 할 때 각 삼각형의 꼭지점에 대한 기하학 정보에 대해 이전 분할과 삼각형의 꼭지점이 접해있는데, 이 정보를 중복 부호화하여 분할별로 독립적으로 부호화하는 방법, 현재 분할에서는 이전 분할에서 방문하지 않은 기하정보에 대해서만 부호화하는 방법, 이미 부호화된 기하정보들 중에서 현재 분할에서 새로 나타나는 여러 개의 기하정보와 연결되어 나타나는 중요한 기하정보에 대해서만 중복되게 부호화하는 방법, 중복 되어 나타나는 기하정보를 샘플링하여 반은 이전 분할에서 부호화하고, 나머지는 현재 분할에서 부호화하는 방법 등이다. 만일 삼각형이 아닌 다각형이 존재한다면, 이 하나의 다각형에 여러 개의 가상의(Imaginary) 에지를 삽입하여 여러 개의 삼각형으로 나눈다. 그리고 이에 대한 정보를 전송한다.

2) 데이터 분할 타입

분할 타입은 정보(Vertex Graph, Triangle Tree, Triangle Data)에 따라 여러 가지로 나뉜다. 그림 4 는 이러한 3 가지 타입을 보여준다. 분할 타입은 2 비트의 Start Code(SC)로 나타낸다. 첫 번째 타입(partition_type_0)은 하나 혹은 여러 개의 CC 를 가지고 있다. 각 CC 는 1 비트의 last_component (lcc)로 끝나는데 이것은 CC 가 분할의 마지막이라는 것을 알려준다. 두 번째 타입 (partition_type_1)은 하나 혹은 그 이상의 vg 를 가지고 있다. 각 vg 에는 1 비트의 last_vg (lvg) flag 가 뒤따르며, 분할에서 마지막이라는 것을 알려준다. 세 번째 타입 (partition_type_3)은 tt 와 td 의 쌍을 가지고 있으며, Vertex Graph Id (vgid), vi 와 bp 모드가 뒤따른다.

sc	vg	tt	td	lcc	...	vg	tt	td	lcc

sc	vg	lvg	...	vg	lvg

sc	vgid	vi	bp	tt	td

그림 4. 분할의 타입: (a) 하나의 분할에 하나 혹은 그 이상의 vg, tt, td 가 있는 경우 (b) 하나의 분할에 하나 혹은 그 이상의 vg 가 있는 경우 (c) 하나의 tt 와 td 쌍을 가지고 있는 경우

3) Tree 의 분할 방법

메쉬 데이터를 분할하고 효율적으로 분할하기 위하여 분할크기에 대한 허용치를 설정하여 Triangle Tree 를 삼각형 단위로 분할한다. 그리고 분할된 TT/TD 데이터 사이의 독립을 유지하면서 Y-vertex 를 다루기 위해 가상의 비트를 삽입하여 분할된 데이터를 가상의 CC 로