

# 파동분할 방식의 순차적 삼차원 메쉬 압축 부호화

김태완, 안정환, 임동근, 호요성  
 광주과학기술원 정보통신공학과  
 광주광역시 북구 오룡동 1번지

## Sequential 3D Mesh Coding using Wave Partitioning

Tae-Wan Kim, Jeong-Hwan Ahn, Dong-Keun Lim and Yo-Sung Ho  
 Kwangju Institute of Science and Technology (K-JIST)  
 1 Oryong-dong, Puk-gu, Gwang-ju, Korea  
 E-mail: twkim@kjist.ac.kr

### 요약

본 논문은 파동분할(Wave Partitioning) 방식을 이용한 순차적(Sequential) 메쉬 부호화 방식을 제안한다. 파동분할 방식은 호수에 물방울이 퍼져 나가는 자연 원리를 이용하여 초기 삼각형의 주위에 삼각형을 덧붙여 가면서 하나의 SPB(Small Processing Block)을 만들어내는 방식이다. 이 방식을 이용하여 하나의 모델을 서로 독립적인 SPB로 분할하고, 각각의 SPB 내에서 초기 삼각형을 중심으로 그것에 덧붙여진 삼각형에 의해 만들어진 원 또는 반원을 찾는다. 또한, 그 원주를 따라 순차적으로 꼭지점을 구하면 각각의 꼭지점들은 일정한 패턴으로 늘어서게 되고, 이를 이용하여 연결성 정보 없이 부가 정보만 이용하여 모델을 순차적으로 무손실 부호화한다.

### 1. 서론

최근 인터넷의 발달에 힘입어 삼차원 모델의 전송과 저장에 대한 중요성이 날로 커지고 있으며, 이를 위해 많은 방법들이 연구되고 있다. 삼차원 모델 부호화 방식은 크게 점진적(Progressive Mesh) 부호화 방식[1]과 순차적(Sequential Mesh) 부호화 방식[4]으로 나눌 수 있다.

QEM(Quadratic Error Metric)[2]을 비롯한 점진적 메쉬 단순화 방식(Simplification)은 손실 부호화 방식이기 때문에 정밀한 부분을 표현하는 모델의 전송이나 저장에는 적합하지 않다. 또한, 처음부터 모델의 전체적인 윤곽을 확인할 수는 있지만, 일부분의 오류만으로도 전체 모델을 다시 전송해야 한다.

반면, 순차적 메쉬 부호화 방식의 경우에는 삼차원 모델이 분할된 후 부호화가 수행되므로, 일부분의 오류는 전체적인 모델에 영향을 미치지 않으며, 모델의 전송이나 저장에 있어서 원하는 부분을 부호화하는데 사용될 수 있다.

본 논문에서는 점진적 메쉬 부호화 과정을 고찰하고, 연결성 정보의 압축에 초점을 둔 파동분할 방식을 이용한 순차적 메쉬 부호화 방식을 제안한다.

### 2. 점진적 메쉬 부호화

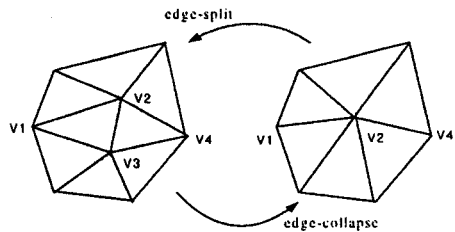


그림 1. edge-collapse 와 edge-split

그림 1에 보인 것처럼, 인접한 두 개의 꼭지점 v2와 v3를 edge-collapse[1] 방법을 이용하여 하나의 꼭지점으로 통합한다. 이 과정에서 인접한 두 면인 (v1, v2, v3)와 (v2, v3, v4)는 없어진다. 따라서 초기 메쉬를 n번의 연속적인 edge-collapse 방법으로 단순화한다.

$$(\hat{M} = M^n) \xrightarrow{ecol_{n-1}} \dots \xrightarrow{ecol_1} M^1 \xrightarrow{ecol_0} M^0$$

edge-split[1] 과정은 edge-collapse 과정의 역과정으로 하나의 꼭지점 v2에 또 다른 꼭지점 v3가 추가되어 새로운 삼각형 (v1, v2, v3)와 (v2, v3, v4)가 만들어지고, 모델은 좀 더 상세히 표현된다. 그림 1의 edge-split 과정에서 vertex-split도 동시에 일어나므로, vertex-split은 edge-split과 동일한 의미로 사용된다. 즉, 임의의 모델은 단순화된 기본 모델과 n개의 vertex-split 변수인 vsplit(v1, v2, v3, v4, A<sub>i</sub>)로 표현될 수 있다. 또한, n번의 vertex-split 과정을 통해 단순화된 모델은 완전히 복원된다. 이러한 표현 방식이 점진적인 부호화 방식이다. 여기서 A는 속성 정보를 의미하며, 색상 정보, 법선 벡터, 질감 정보 등을 포함한다.

$$M \xrightarrow{vsplit_n} M \xrightarrow{vsplit_{n-1}} \dots \xrightarrow{vsplit_1} (M^n = \hat{M})$$

### 3. 순차적 매쉬 부호화

#### 3.1 삼차원 모델의 파동분할 방법

제안한 방법은 물방울이 물 위에 떨어져 원 또는 반원을 그리며 퍼져 나가는 파동 현상을 이용한다. 그림 2에 보인 것처럼, 임의의 초기 삼각형을 정하고, 이 삼각형의 세 모서리를 공유하는 삼각형 A를 화살표 방향으로 덧붙인다. 덧붙여진 삼각형 A의 모서리를 공유하는 삼각형 B를 화살표 방향으로 덧붙인다. 다시 덧붙여진 삼각형 B의 모서리를 공유하는 삼각형 C를 화살표 방향으로 덧붙인다. 덧붙여지는 삼각형의 순서에 관계없이 이러한 과정을 원하는 수의 삼각형을 가지는 SPB(Small Processing Block)을 얻을 때까지 반복적으로 수행한다.

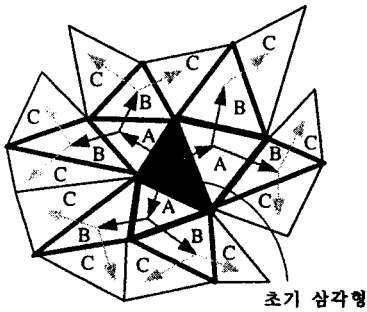


그림 2. 파동분할을 이용한 SPB 형성

#### 3.2 파동분할을 이용한 순차적 부호화

파동분할 방법에 의해 모델이 분할되면, 분할된 각각의 SPB내에서 초기 삼각형의 접선과 초기 삼각형에 덧붙여진 삼각형의 외부 접선으로 만들어지는 원 또는 반원의 원주를 따라 시계방향 또는 반시계방향으로 꼭지점을 순차적으로 구한다.

하나의 원주를 따라 꼭지점이 구해지면, 그 원주의 바깥에 있는 다음 원주를 따라 꼭지점이 연속적으로 구해진다. 하나의 원주를 Rank로 정의하면, 각 꼭지점은 해당하는 Rank에 포함된다. 각 Rank에 포함된 꼭지점의 개수는 각 꼭지점의 모 꼭지점과 부호화 변수와 함께 부호화된 후 꼭지점 정보와 같이 전송된다. 복호화 부분에서는 전송된 꼭지점 스트림을 각 Rank내에서의 개수로 분할하여 사다리 모양의 배열을 만들고, 이것을 기반으로 복호화를 수행한다.

일반적으로 분할하고자 하는 부분의 중심 또는 가장자리에 위치한 임의의 삼각형을 초기 삼각형으로 정한다. 분할된 매쉬는 초기 삼각형의 위치에 따라 원 SPB와 반원 SPB로 나누어진다. 원 SPB는 초기 삼각형이 SPB 중심에 위치하는 경우로, 초기 삼각형 주위의 삼각형이 초기 삼각형에 덧붙여져 원 형태를 이룬다. 반면, 반원 SPB는 초기 삼각형이 SPB 가장자리에 위치하는 경우로 초기 삼각형 주위의 삼각형이 초기 삼각형에 덧붙여져 반원 형태를 이룬다.

#### 3.2.1. 원 SPB

그림 3에서 굵은 선은 하나의 Rank를 나타낸다.

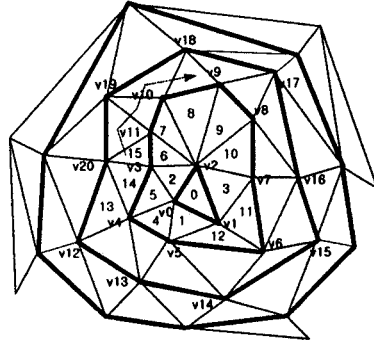


그림 3. 분할된 원 SPB

$v_0$ 가  $v_1 \sim v_5$ 에 의해 둘러싸여 있을 때,  $v_0$ 를 꼭지점  $v_3 \sim v_5$ 의 모 꼭지점(Mother Vertex)으로 정의하고,  $v_3 \sim v_5$ 를  $v_0$ 의 자 꼭지점(Son Vertex)로 정의한다. 꼭지점  $v_1$ 과  $v_2$ 는  $v_0$ 와 같은 Rank에 포함되기 때문에  $v_1 \sim v_2$ 는  $v_0$ 의 자 꼭지점이 되지 않는다. 따라서,  $v_1$ 은  $v_6 \sim v_7$ 의 모 꼭지점이 되고,  $v_2$ 는  $v_8 \sim v_{11}$ 의 모 꼭지점이 된다. 초기 삼각형의 세 꼭지점  $v_0, v_1$ , 그리고  $v_2$ 로 이루어진 폐곡선을 Rank I이라 한다. Rank I에 포함된 초기 삼각형의 꼭지점을 제외한 임의의 Rank내에서의 꼭지점 스트림은 이전 Rank인 (Rank-1)에 속하는 모 꼭지점을 중심으로 반 시계방향으로 회전하면서 얻어진다. 그러면 Rank I에 놓여진 꼭지점을 모 꼭지점으로 하여 얻어진 새로운 꼭지점  $v_3 \sim v_{11}$ 의 집합이 Rank II가 된다.

그림 4는 그림 3의 원 SPB에 대해 각각의 Rank에 포함되는 꼭지점 스트림을 일렬로 나열한 그림이다. Rank내에서의 꼭지점과 이전 Rank인 (Rank-1)에 포함되어 있는 모 꼭지점과의 관계는 진한 직선을 이용하여 나타내었다. 하나의 Rank는 원 모양으로 폐곡선을 이루기 때문에 Rank 마지막에 초기 꼭지점을 한번 더 넣어 폐곡선임을 나타냈다. 삼각형 내의 번호는 그림 3에 나타난 삼각형과 대응되는 삼각형을 나타낸다.

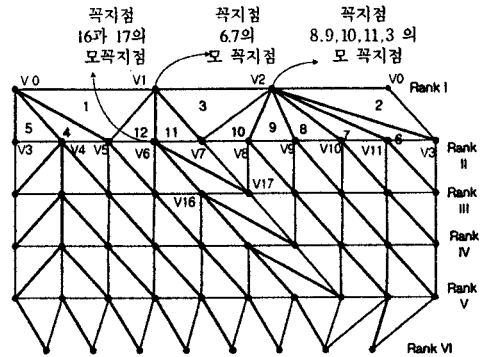


그림 4. 원 SPB의 순차적 꼭지점 배열

### 3.2.2 반원 SPB

그림 5에서 보는 바와 같이, v4와 v5, v11과 v12를 공유하는 삼각형이 없기 때문에 Rank I를 제외한 하나의 Rank는 반원 형태를 이룬다.

원 SPB와 같은 원리를 이용하여 v0은 v3~v4의 모 꼭지점이 되고, v1는 v5~v6의 모 꼭지점이 된다. 그리고 v2는 v7~v8의 모 꼭지점이 된다. 따라서 Rank I에는 v0~v2가 포함되고, Rank II에는 v3~v8이 포함된다.

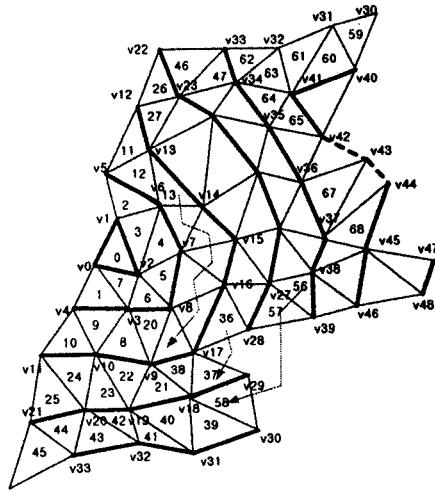


그림 5. 분할된 반원 SPB

그림 6은 그림 5의 반원 SPB에 있어서 각각의 Rank에 포함되는 꼭지점 스트림을 일렬로 나열한 그림이다. v4와 v5 사이와 v11과 v12 사이에 연결성 정보가 없는 것은, 그림 5에서 보는 바와 같이, v4와 v5 그리고 v11과 v12를 공유하는 삼각형이 없는 경우이다. 이처럼 매쉬가 두 부분으로 나누어지는 경우는 1비트의 변수로 구별한다.

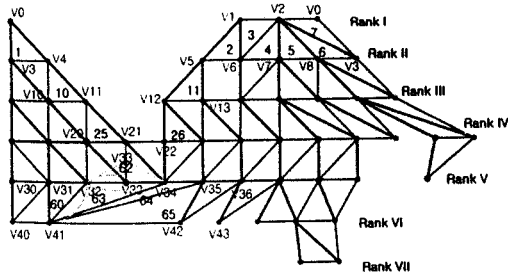


그림 6. 반원 SPB의 순차적 꼭지점 배열

### 3.3. 모 꼭지점을 이용한 순차적 복호화

부호화 과정에서 모 꼭지점은 모 꼭지점이 포함되는 Rank 내에서의 위치로 부호화된다. 즉, 아래의 표 2에서 보는 바와 같이, v42의 모 꼭지점은 Rank V에

포함되어 있는 꼭지점 v35가 된다. v35는 Rank V내에서 0에서부터 6번째 해당하는 꼭지점이므로, v42의 모 꼭지점은 5로 부호화된다. 따라서, 꼭지점 v43의 모 꼭지점은 같은 방법으로 6으로 부호화 되고, 이들 꼭지점 v42와 v43의 모 꼭지점의 차분은 1이 된다.

현재의 꼭지점을 Cv, 부호화 과정에서 Cv 다음에 구해진 꼭지점을 Nv, 그리고 Cv와 Nv의 모 꼭지점을 각각 Mv1과 Mv2로 정의하면, 이들의 관계는 다음과 같이 정리된다.

표 1. 그림 5에서 꼭지점 사이의 관계

Cv	Nv	Mv1	Mv2	Mv1-Mv2
v6	v7	v1	v1	0
v7	v8	v1	v2	1
v8	v9	v2	v2	0

표 2. 그림 6에서 꼭지점 사이의 관계

Cv	Nv	Mv1	Mv2	Mv1-Mv2
v40	v41	v30	v31	1
v41	v42	v31	v35	4
v42	v43	v35	v36	1

Mv1과 Mv2의 차분은 크게 세 가지로 분류된다.

<유형 1> : |Mv1 - Mv2|=0

이 경우에는 Rank내에서의 Cv, Nv 그리고 (Rank-1)에서의 Mv1으로 이루어진 삼각형을 구한다. 그림 6에서 꼭지점 v3와 꼭지점 v4의 모 꼭지점 사이의 차분은 0이 된다. 이것은 유형 1에 해당되므로 Rank내에 있는 Cv인 v3와 Nv인 v4 그리고 (Rank-1)에서의 Mv1인 v0를 이용하여 삼각형을 만든다.

<유형 2> : |Mv1 - Mv2|=1

이 경우도 유형 1과 마찬가지로 꼭지점 Cv와 꼭지점 Nv 그리고 Mv1을 이용하여 삼각형을 만든다. 하지만 이 경우에는 모 꼭지점의 차이가 1이기 때문에, (Rank-1)상에 있는 두 꼭지점 Mv1과 Mv2 그리고 Cv로 이루어지는 삼각형을 추가로 구해야 한다.

예를 들어, 그림 4에서 v5와 v6의 Mv의 차분은 1이며 유형 2에 속하게 된다. 이 경우 유형 1의 경우처럼 삼각형 (v5, v6, v1)을 만든다. 그 다음 두 꼭지점의 모 꼭지점들과 현재의 꼭지점으로 이루어지는 삼각형 (v0, v1, v5)를 추가로 만들어야 한다.

<유형 3> : |Mv1 - Mv2|≥2

이 경우는 반원 형태로 만들어진 SPB의 양쪽 두 부분이 서로 접하는 경우와 완전한 원이나 반원을 이루지 못하는 경우 발생한다. Cv, Nv, Mv1, Mv2 그리고 Mv1과 Mv2사이의 꼭지점들로 삼각형이 만들어진다. 만들어지는 삼각형의 유형은 각각 1비트의 변수로 정의되고, 각 꼭지점마다 적용된다.

예를 들어, Cv인 v41과 Nv인 v42 그리고 이것들의 모 꼭지점에 해당하는 두 꼭지점 v31과 v35, 그리고 모 꼭지점 사이의 꼭지점인 v32~v34로 이루어지는 삼

각형을 구한다. 즉 삼각형(v31, v32, v41), (v32, v34, v41), (v41, v34, v35), (v41, v35, v42), 그리고 (v32, v33, v34)가 만들어진다. 3가지 유형에 있어서 각 Rank의 마지막 꼭지점은 처음 꼭지점과 같게 되고 그것의 모 꼭지점은 n개의 꼭지점을 가진 (Rank-1)의 (n-1)번째 꼭지점이 된다. 즉 그림 6에서 Rank II의 7번째 꼭지점 v3의 모 꼭지점은 Rank I의 3번째 꼭지점 v2가 된다.

### 3.4 부호화 알고리즘 흐름도

그림 7은 부호화 알고리즘의 흐름도를 나타낸 것이다. 여기서 유의해야 할 점은, 부호화기의 마지막 부분에 복호기를 설치하는 것이다. 이것은 복호화부에서 찾지 못하는 연결성 정보(삼각형 정보)가 존재할 경우를 대비한 것으로 이들에 대한 정보는 부호화를 거치지 않고, 부호화 변수, 모 꼭지점, Rank내의 꼭지점 개수, 그리고 꼭지점 정보와 같이 전송된다.

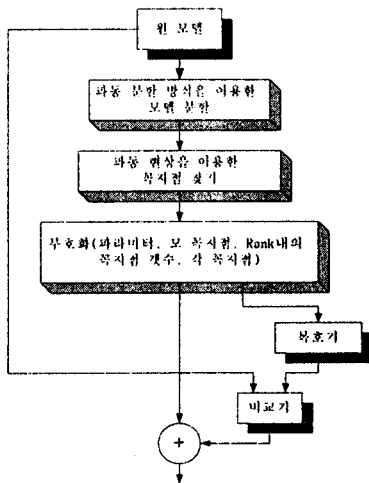
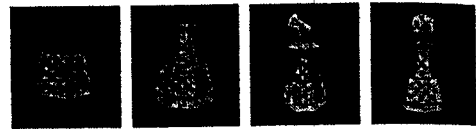


그림 7. 부호화 흐름도

## 4. 실험 및 결과

본 논문에서는 삼차원 모델을 4개로 분할하여 부호화하였다. 분할된 SPB의 크기는 초기 메쉬에 덧붙여지는 삼각형의 개수를 기준으로 정하였다. 그리고 꼭지점 정보의 압축을 배제하고, 연결성 정보만을 부호화 하는데 초점을 두었다. 그림 8은 본 알고리즘을 PAWN 모델, MUSHROOM 모델 그리고 KING 모델에 적용하여 삼차원 모델이 순차적으로 부호화되는 과정을 나타내고 있다. 그림 8에 나타난 것처럼, 순차적 메쉬 부호화는 무손실 부호화이기 때문에 원 모델과 똑같이 부호화되었다.

일반적으로 삼각형의 개수는 꼭지점의 개수보다 약 2배가 많다. 제안한 알고리즘에서는 삼각형에 대한 연결성 정보 대신에, 각 꼭지점에 그 꼭지점의 모 꼭지점 정보와 변수 정보를 추가하였으므로 상당한 압축 효율을 얻을 수 있을 것으로 예상된다.



(a) PAWN 모델



(b). MUSHROOM 모델



(b) KING 모델

그림 8. 순차적 부호화

## 5. 결론

본 논문에서는 오류 내성과 연결성 정보의 효율적인 압축을 위해 과동분할 방식을 이용한 순차적 메쉬 압축 부호화 방식을 제안하였다. 제안한 방식에서는 삼차원 모델의 저장과 전송을 위해 기하학 정보와 연결성 정보를 모두 전송하는 기존의 방식에서 벗어나, 연결성 정보 없이 기하학 정보와 그것의 부가 정보만을 이용하여 삼차원 모델을 부호화하였다. 실험 결과에서 보인 것처럼, 손실 없이 삼차원 모델을 복원할 수 있었다. 또한, 이 방법이 기하학 정보의 압축에 초점을 둔 기존의 여러 부호화 방식과 결합된다면, 상당한 압축 효율을 얻을 수 있을 것이다.

## 감사의 글

본 연구는 광주과학기술원(K-JIST) 초고속망네트워킹연구센터(UFON)를 통한 한국과학재단 우수연구센터(ERC) 지원금에 의한 것입니다.

## 참고 문헌

- [1] H. Hoppe, "Progressive Meshes," *SIGGRAPH*, pp. 99-108, Aug. 1996.
- [2] M. Garland and P. S. Heckbert, "Surface Simplification using Quadric Error Metrics", *SIGGRAPH*, pp. 209-216, Aug. 1997.
- [3] G.K. Wallace, "The JPEG Still Picture Compression Standard," *Communication of the ACM*, 34(4), pp. 31-44, April 1991.
- [4] 송문섭, 안정환, 김성진, 한만진, 호효성, "삼차원 모델의 점진적인 렌더링과 오류 강인을 위한 데이터 분할 방법(CODAP)," 전자공학회 추계 학술 대회, pp. 1089-1092, 11월, 2000.