

---

# 준균일 메쉬 재구성을 이용한 메쉬 시퀀스 압축 기법

## Animated Mesh Compression with Semi-regular Remeshing

---

안민수  
삼성종합기술원

Minsu Ahn(atom@postech.ac.kr)

---

### 요약

최근 공동 연결 관계와 연속적인 정점 위치들의 움직임으로 이루어진 메쉬들의 모임, 즉 메쉬 시퀀스를 압축하는 연구가 활발하게 이루어지고 있다. 본 논문은 Khodakovsky 등이 제시한 준균일 메쉬 압축방법에 기반한 메쉬 시퀀스의 압축 알고리즘을 제시하고자 한다. 준균일 메쉬 시퀀스로의 메쉬 재구성을 이용한 메쉬 시퀀스 압축 알고리즘은 크게 두 부분으로 이루어진다. 첫 번째 부분은 주어진 비균일 메쉬 시퀀스로부터 준균일 메쉬 시퀀스를 생성하는 것이다. 준균일 메쉬를 생성하기 위해 본 논문에서는 MAPS 알고리즘을 사용하였다. 하지만 단일 메쉬에 대해 적용이 가능한 MAPS 알고리즘을 메쉬 시퀀스에 그대로 적용할 수 없다. 따라서 주어진 애니메이션에서의 정점 움직임을 고려하여 유사한 움직임을 가지는 영역별로 분할하고, 이 분할 정보과 정점의 움직임을 고려할 수 있도록 MAPS 알고리즘을 확장하였다. 두 번째 단계에서는 웨이블릿 변형과 메쉬 분할 정보를 이용해 준균일 메쉬를 압축하였다. 각 분할 영역의 변환 정보를 고려해 분할 영역 내 정점의 위치를 예측하고, 참조 프레임과의 차이값을 압축함으로써 효율적으로 준균일 메쉬 시퀀스를 압축하였다.

■ 중심어 : | 메쉬 압축 | 메쉬 시퀀스 | 메쉬 정규화 |

### Abstract

This paper presents a compression method for animated meshes or mesh sequences which have a shared connectivity and geometry streams. Our approach is based on static semi-regular mesh compression algorithm introduced by Khodakovky et al. Our encoding algorithm consists of two stages. First, the proposed technique creates a semi-regular mesh sequence from an input irregular mesh sequence. For semi-regular remeshing of irregular mesh sequences, this paper adapts the MAPS algorithm. However, MAPS cannot directly be performed to the input irregular mesh sequence. Thus, the proposed remesh algorithm revises the MAPS remeshing using the clustering information, which classify coherent parts during the animation. The second stage uses wavelet transformation and clustering information to compress geometries of mesh sequences efficiently. The proposed compression algorithm predicts the vertex trajectories using the clustering information and the cluster transformation during the animation and compress the difference other frames from the reference frame in order to reduce the range of 3D position values.

■ keyword : | Mesh Compression | Mesh Sequences | Semi-regular Mesh |

## I. 서론

동일한 연결 관계를 가지는 메쉬들의 모임인 메쉬 시퀀스는 일반적으로 캐릭터 애니메이션, 컴퓨터 게임, 그리고 물리 시뮬레이션과 같은 다양한 응용 분야에 활용되고 있으며, 메쉬 시퀀스를 효율적으로 저장하고 전송하기 위해 이에 대한 압축 연구가 활발하게 진행되고 있다. 일반적으로 메쉬 시퀀스는 크게 두 가지 부분, 메쉬 간에 공동으로 사용하는 연결 관계와 시간에 따라 변하는 정점들의 모임으로 이루어져 있다.

기존 메쉬 시퀀스 압축 알고리즘들은 시간에 따라 변하는 정점의 움직임을 어떻게 압축할 것이냐에 초점을 두었다[5][8-10]. 즉 공동으로 사용하는 연결 관계는 수정없이 기존의 메쉬 연결 관계 압축 알고리즘을 적용하고, 정점들 간의 유사성을 이용하여 메쉬 시퀀스를 압축하는 방법을 사용하였다. 이와는 달리 본 논문에서는 메쉬 시퀀스에 대해 메쉬 정규화를 적용하여 압축 방법을 제시하고자 한다.

최근 단일 준균일 메쉬를 압축하는 많은 알고리즘들이 제시되고 있다[1][6]. 균일한 연결 관계를 가지는 준균일 메쉬(semi-regular mesh)의 구조적 특성 때문에, 연결 관계 압축시 장점이 있을 뿐만 아니라, 오랜 기간 동안 발전해 온 영상 압축 알고리즘을 적용할 수 있다. 본 논문에서는 준균일 메쉬의 이러한 특성을 이용해 주어진 비균일 메쉬 시퀀스(irregular mesh sequence)를 준균일 메쉬 시퀀스로 전환하여 압축함으로써, 높은 압축률을 얻고자 한다. 이 때 준균일 메쉬 시퀀스는 입력 비균일 메쉬 시퀀스와 같이 모든 프레임에 대해서 동일한 연결 관계를 가진다.

준균일 메쉬 압축 기법을 적용하기 위해서 우선 비균일 메쉬를 준균일 메쉬로 전환하는 메쉬 재구성(remeshing) 방법이 필요로 하며, 기존에 이와 관련된 많은 알고리즘이 제시되었다[2]. 하지만 이러한 방법들은 대부분 하나의 비균일 메쉬에 대해 적용 가능한 방법으로 메쉬 시퀀스에 대해 바로 적용할 수 없다. 따라서 애니메이션 기간 동안 모든 정점의 움직임을 고려해 준균일 메쉬 시퀀스를 생성하는 알고리즘이 필요하다. 본 논문에서는 이러한 문제점을 해결하기 위해 메쉬 시

퀀스 내 정점 이동 경로와 형태를 동시에 고려한 준균일 메쉬 시퀀스 생성 알고리즘을 소개하고자 한다. 또한 메쉬 시퀀스 내 정점의 움직임을 효율적으로 압축하기 위해서, 본 논문에서는 서로 유사한 움직임을 갖는 정점들로 분류하는 방법을 사용하였다. 즉 유사한 움직임을 가지는 정점 또는 면 단위로 메쉬를 분할하고, 이들의 움직임을 표현하는 변형 행렬(transform matrix)을 추출하여 분할 영역 내 정점의 위치를 예측하는데 사용하였다.

## II. 배경 지식 및 이전 연구

메쉬 시퀀스  $\mathbf{M} = \{M_1, \dots, M_T\}$ 은 동일한 연결 관계를 가지는 메쉬들의 모임이며, 이를 구성하는 각 메쉬  $M_t$ 를 시간  $t$ 에 대한 프레임 메쉬(frame mesh)라고 하겠다. 이러한 프레임 메쉬들은 주어진 시간  $t$ 에서의 메쉬 형태를 표현한다. 그리고 각 프레임 메쉬는 삼각형으로 이루어진 이차원 다양체(2-manifold)로 가정한다.

### 1. 준균일 메쉬 생성 및 압축

비균일 메쉬로부터 준균일 메쉬를 얻기 위해 가장 널리 사용되는 방법이 Lee 등이 제시한 MAPS 알고리즘이다[2]. MAPS 알고리즘은 크게 두 부분으로 이루어진다. 첫 번째, 주어진 비균일 메쉬를 단순화해 기본 메쉬를 생성한다. 이 과정에서 입력 메쉬와 중간 단순 메쉬 간의 대응 관계는 계속 유지된다. 두 번째 과정은 입력 메쉬와 기본 메쉬 간의 대응 관계를 이용해 리샘플링(re-sampling)과 삼각형 사분할을 연속적으로 적용하여 준균일 메쉬를 생성하는 것이다.

이렇게 얻어진 준균일 메쉬는 기본 메쉬에 속하는 정점을 제외한 다른 정점은 6개의 정점과 이웃한다. 또한 기본 메쉬의 삼각형을 연속적으로 사분할(1-to-4 subdivision)함으로써, 추가되는 정점의 위치는 분할 전 삼각형의 정점들의 위치로부터 예측할 수 있어 압축이 용이하다. 이러한 장점 때문에 준균일 메쉬를 압축하는 다양한 방법들이 제시되었다[1][6].

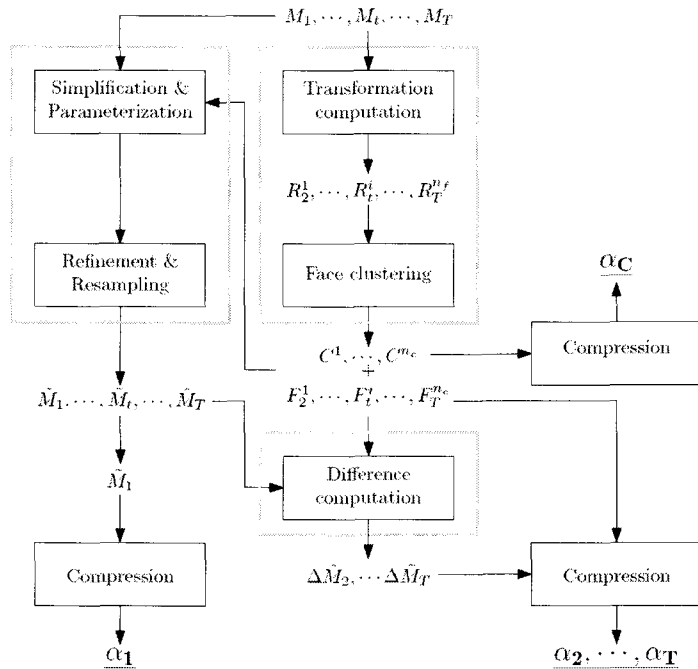


그림 1. 메쉬 시퀀스 압축 과정

## 2. 메쉬 시퀀스 압축

최근 10여년간 메쉬 시퀀스 압축 분야에 대해 활발히 연구가 진행되었으며 대부분 연구 내용은 정점들의 이동 경로의 압축에 초점을 두었다. Lengyel은 총체적인 접근 방법(heuristic approach)을 통해 정점 단위로 메쉬를 분할하고, 각 분할 영역의 움직임을 예측자(predictor)들의 선형 조합으로 표현하고, 선형 조합 정보를 압축하였다[9]. Satter 등은 주성분 분석(principal component analysis)을 통해 메쉬를 분할하고, 주 이동 경로(eigen-trajectory)로 정점의 움직임을 표현하고, 압축하는 방법을 제시하였다[5]. 또한 Alexa 등은 주성분 분석을 통해 평균 움직임과의 차이를 계산함으로써, 이를 통해 정점의 이동 경로를 압축하였다[10].

Briceno 등은 기하 영상(geometry image) 알고리즘 [6]을 각 프레임 메쉬에 적용하고, 비디오 압축 방식을 사용하여 메쉬 시퀀스를 압축하는 방식을 제안하였다 [7].

Guskov와 Khodakovsky는 웨이블릿(wavelet) 기반의 메쉬 시퀀스 압축 방식을 제안하였다[8]. 다음 프레

임에서의 정점의 위치를 예측하기 위해서 이방성 웨이블릿 기반을 정의하였으며, 이 방식은 메쉬 시퀀스의 특징점을 잘 보존하는 장점을 가지고 있다.

앞에서도 언급했듯이 메쉬 시퀀스를 압축하는 대부분의 알고리즘은 정점들의 움직임을 압축하는데 초점을 두고 있다. 하지만 메쉬는 정점의 위치 정보뿐만 아니라, 이들 간의 연결 정보를 포함하고 있다. 따라서 압축률을 높이기 위해서는 연결 정보를 효율적으로 압축할 필요가 있다. 이러한 이유로 본 논문에서는 준균일 메쉬 시퀀스로서의 메쉬 재구성을 통해 메쉬 시퀀스를 압축하고자 한다. 특히 메쉬 재구성을 위해 정점의 위치뿐만 아니라 움직임을 동시에 고려한 메쉬 시퀀스 단순화 알고리즘을 제시하고자 한다. 또한 Khodakovsky가 제시한 방법[1]을 통해 정점의 움직임도 효율적으로 압축할 수 있었다. 이외에 유사한 움직임을 가지는 영역 단위로 분할하여 분할 영역 전체의 움직임을 통해 분할 영역 내 정점의 움직임을 예측함으로써, 압축률을 높이고자 하였다.

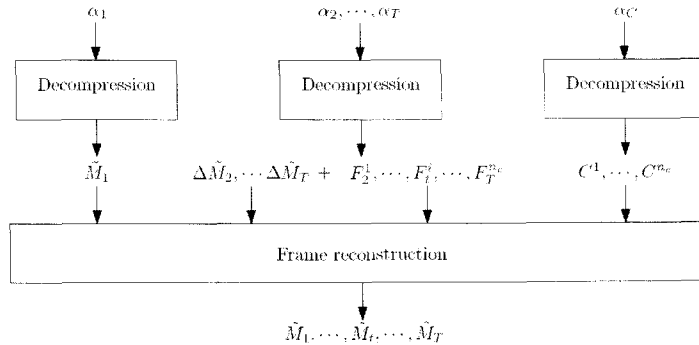


그림 2. 메쉬 시퀀스 압축 해제 과정

### III. 전체 과정

본 장에서는 주어진 입력 메쉬 시퀀스를 압축하는 전체 과정에 대해서 설명하도록 하겠다. 본 논문에서 제시한 압축 알고리즘은 크게 두 가지 과정, 준균일 메쉬 시퀀스 생성과 델타 부호화 (delta coding)으로 이루어진다. 첫 번째, 주어진 메쉬 시퀀스  $\mathbf{M} = \{M_1, \dots, M_T\}$  으로부터 준균일 메쉬 시퀀스  $\tilde{\mathbf{M}} = \{\tilde{M}_1, \dots, \tilde{M}_T\}$ 로 전환하였다. 두 번째 과정에서는 이렇게 얻어진 준균일 메쉬 시퀀스  $\tilde{\mathbf{M}}$ 에 대해 웨이블릿 기반 압축 알고리즘을 적용하였다. 본 논문에서는 각 프레임에 해당하는 메쉬의 정점 위치에 대해서 독립적으로 웨이블릿 기반 알고리즘을 적용하는 것이 아니라, 기준과의 차이값에 대해 웨이블릿 기반 압축 알고리즘을 적용하였다. 즉, 메쉬 시퀀스 내 첫번째 메쉬를 참조 메쉬 (reference mesh)로 선택하고 나머지 메쉬는 참조 메쉬와의 차이값만을 계산하여 압축한다. 이렇게 두 과정을 거쳐 메쉬 시퀀스를 압축하는 과정을 [그림 1]과 같이 도식화할 수 있으며, 이를 정리하면 다음과 같다.

1. 프레임  $t$ 에 해당하는 메쉬  $M_t$ 의 각 삼각형에 대해 참조 프레임에서 현재 위치로의 이동하는 변형 행렬  $R_t^i$ 를 계산한다.
2. 삼각형의 변형 행렬  $R_t^i$ 을 이용해  $\mathbf{M}$ 을 유사한 변형 행렬을 가지는 삼각형의 모임  $C_t$ 로 분할한다.
3. 각 정점  $p_t^i$ 의 위치 정보와 분할 영역 정보  $C_t$ 를 이용해 메쉬 시퀀스  $\mathbf{M}$ 로부터 준균일 메쉬  $\tilde{\mathbf{M}}$ 을

얻는다. 이에 대한 자세한 설명은 4장에서 하도록 하겠다.

4. 참조 프레임 메쉬를 압축하여  $\alpha_1$ , 분할 영역 정보를 압축하여  $\alpha_C$ 를 얻는다.
5. 나머지 프레임 메쉬의 정점에 대해 참조 프레임 메쉬 정점 위치 차이값을 계산한 후, 이 차이값을 압축함으로써,  $\alpha_2, \dots, \alpha_T$ 를 얻는다. 이에 대한 자세한 내용은 5장에서 설명하도록 하겠다.

이렇게 얻어진 준균일 메쉬를 압축 데이터를 압축 해제하는 과정은 [그림 2]와 같다. 우선 참조 프레임 메쉬의 압축 데이터  $\alpha_1$ 과 분할 정보 압축 데이터  $\alpha_C$ 를 압축 해제하고, 이를 이용해 나머지 프레임 메쉬의 압축 데이터를 복원한다. 이 과정도 5장에서 설명하도록 하겠다.

### IV. 준균일 메쉬로의 재구성

본 장에서는 비균일 연결 관계를 가지는 메쉬 시퀀스를 준균일 연결 관계를 가지는 메쉬 시퀀스로 재구성하는 알고리즘에 대해서 설명하도록 하겠다. 앞서도 언급했듯이, 본 논문에서는 준균일 메쉬 시퀀스를 생성하기 위해서 MAPS 알고리즘을 확장, 적용하였다[2]. MAPS 알고리즘을 통해 생성한 준균일 메쉬의 성능은 기본 메쉬의 형태로부터 좌우된다. 따라서 기본 메쉬가 입력 메쉬의 특징을 잘 반영해야 입력 메쉬와 시각적으로

로 오차가 없는 '평균일' 메쉬를 얻을 수 있다. 그리고 본 논문에서는 하나의 메쉬가 아니라 메쉬 모임을 다루는 것이기 때문에, 이들 모두 특징을 잘 반영하는 기본 메쉬를 얻어야 입력 메쉬 시퀀스와 적은 오차를 가지는 '평균일' 메쉬를 얻을 수 있다.

### 1. 평균 이동 기반 메쉬 영역 분할 (Clustering)

평균일 메쉬를 얻기 위해서, 우선 메쉬 시퀀스  $M$ 으로부터 메쉬 단순화 과정을 통해 기본 메쉬 (base mesh)  $M^0$ 를 생성해야 한다. 단일 메쉬의 경우, 단순히 정점의 위치만 고려하면 되지만, 여러 개의 메쉬로 이루어진 메쉬 시퀀스로부터 기본 메쉬를 생성할 때에는 정점의 위치뿐만 아니라, 정점의 움직임까지 같이 고려해야 한다. 즉, 인접한 정점의 위치와 크게 다른 정점들뿐만 아니라, 주위의 정점보다 활발하게 움직이는 정점들을 유지하면서 기본 메쉬  $M^0$ 를 생성하고자 한다. 이러한 문제를 해결하기 위해 본 논문에서는 유사한 움직임을 가지는 영역별로 메쉬를 분할하고, 메쉬 단순화 과정에서 분할 영역 간 경계 부분이 유지되도록 하였다. 왜냐하면 분할 영역 간 경계 부분이 가장 활발하게 변하는 부분이기 때문이다.

유사한 움직임을 가지는 영역  $C = \{C^1, \dots, C^{m'}\}$ 으로 분할하기 위해, 본 논문에서는 삼각형의 회전 행렬을 계산하고 이들 값에 대해 평균 이동(Mean-Shift) 알고리즘을 적용하였다[3]. 현재 프레임  $t$ 에서 메쉬를 구성하는 각 삼각형  $f_t^i$ 에 대해 참조 프레임 메쉬의 위치로부터 현재 위치로의 변형 행렬  $R_t^i$ 를 계산한다. 이렇게 얻은 삼각형  $f^i$ 의 회전 행렬 모임  $\{R_t^i\}$ 에 대해 평균 이동 알고리즘을 적용하여 입력 메쉬 시퀀스를 유사한 회전 행렬을 가지는 삼각형의 모임으로 분할하였다.

평균 이동 알고리즘은 영역 분할을 위해 가장 많이 사용되는 알고리즘으로 다른 알고리즘에 비해 장점이 많다. 첫 번째,  $k$ -평균 분할 알고리즘 등 다른 분할 알고리즘에 비해 돌출부 (outlier)에 민감하지 않다. 두 번째, 분할 영역의 수를 사용자가 지정할 필요가 없다. 이러한 이유로 본 논문에서는 평균 이동 알고리즘을 사용하여 입력 메쉬 시퀀스를 분할하였다.

평균 이동 알고리즘을 적용한 후, 대부분의 분할 영역들은 적당한 수의 삼각형과 형태를 가지게 된다. 하지만 일부 분할 영역이 적은 수의 삼각형들로 이루어졌다면, 이 분할 영역을 이웃하는 분할 영역 중 유사한 회전 행렬을 가지는 것과 합치도록 했다. 이렇게 메쉬 시퀀스에 대해 영역 분할을 적용한 결과는 [그림 3]과 같다.



그림 3. 메쉬 영역 분할 예제

이렇게 얻어진 분할 영역은 기본 메쉬를 생성하기 위해 사용될 뿐만 아니라, 메쉬 재구성을 통해 얻은 '평균일' 메쉬를 압축할 때도 사용한다. 다시 말해 같은 분할 영역에 속하는 정점들은 메쉬 시퀀스 내에서 유사한 움직임을 가지기 때문에, 전체 분할 영역의 움직임을 안다면 이에 속하는 정점 또는 면의 움직임을 예측할 수 있다. 따라서 각 분할 영역에 대해 참조 프레임 메쉬로부터 주어진 시간  $t$ 로의 변형 행렬 (transformation matrix)의 모임  $\{F_t^i\}$ ,  $F_t^i \in \mathbb{R}^{4 \times 4}$ 을 근사적으로 구함으로써, 주어진 시간  $t$ 의 정점  $v$ 의 위치를 수식 (1)과 같이 표현할 수 있다.

$$p_t = F_t^i \cdot p_1 + \Delta p_t \tag{1}$$

이 때  $|\Delta p_t|$ 는 매우 작은 값을 가질 것으로 기대되며,  $p_t$  대신  $\Delta p_t$ 를 압축함으로써, 높은 압축률을 얻을 수 있었다.

### 2. 기본 메쉬 생성

기본 메쉬를 생성하기 위해 본 논문은 Garland 등이 제시한 단일 메쉬 단순화 알고리즘[3]을 확장하였다. 즉 단일 메쉬에 대해 정의한 이차 에러 기준 (quadratic error metric)을 메쉬 시퀀스에 적용할 수 있도록 재정의하고, 이를 이용해 에지 붕괴 (edge collapse) 연산자를 연속적으로 정의함으로써, 메쉬 시퀀스에 대한 기본 메쉬를 얻을 수 있었다. 이렇게 얻어지는 기본 메쉬도 입

력 메쉬 시퀀스와 같이 공동 연결 관계와 시간에 따라 변하는 정점들로 이루어진다.

메쉬 시퀀스의 정점에 대한 이차 에러 기준을 정의하기 전에 앞서 메쉬 단순화 과정에서 유지되어야 하는 정점들의 종류는 다음과 같다. 첫 번째, 단일 메쉬의 단순화 과정과 같이 주위 정점의 위치에 비해 위치가 많이 차이 나는 정점은 유지되어야 한다. 두 번째, 애니메이션 과정에서 주위 정점들에 비해 활발하게 움직이는 정점들도 유지되어야 한다. 마지막으로 메쉬 단순화 과정에도 분할 영역들 간의 관계는 반드시 유지되어야 한다. 앞에서도 언급했듯이 유사한 움직임을 갖도록 영역을 분할했기 때문에, 분할 영역의 경계 부분이 애니메이션 기간 동안 가장 활발하게 움직이는 부분이다. 이러한 정점들 유지하기 위해서 정의한 이차 에러 기준은 수식 (2), (3)과 같다.

$$Q_i^j = (1-r) \cdot Q_{p_i} + r \cdot Q_{\Delta p_i} \quad (2)$$

$$Q_i^j = \sum_{i=2}^T Q_i^j \quad (3)$$

수식 (2)에서  $Q_{p_i}$ 는 시간  $t$ 에서의 정점  $p_i$ 의 위치에 대해 Garland 등이 정의한 이차 에러 기준이고,  $Q_{\Delta p_i}$ 는 이전 시간  $(t-1)$ 에서의 정점 위치와 현재 시간  $t$ 에서의 정점 위치 간의 차이값에 대한 이차 에러 기준이다.  $Q_{p_i}$ 에 의해 계산되는 값은 주위의 정점 위치와  $p_i$ 가 차이가 클수록 커지고,  $Q_{\Delta p_i}$ 는 정점  $p_i$  위치가 주위보다 활발히 움직일 때, 큰 값을 가지게 된다. 두 이차 오차  $Q_{p_i}$ ,  $Q_{\Delta p_i}$ 의 가중치 합으로 현재 시간  $t$ 에서의 정점  $p_i$ 에 대한 이차 에러 기준  $Q_i^j$ 을 정의하였다. 또한 모든 시간을 고려한 정점  $p_i$ 의 최종 이차 에러 기준은 수식 (3)와 같이  $Q_i^j$ 의 합으로 정의하였다.

분할 영역 정보와 동시에 본 논문에서 정의한 에러 기준으로 그 값이 작은 순서로 에지 붕괴 연산자를 연속적으로 적용함으로써, [그림 4]와 같은 결과를 얻었다. 또한 에지 붕괴 연산자를 같은 분할 영역에 속하는 에지에 대해서만 적용하였다. [그림 4]에서 알 수 있듯이 기본 메쉬의 각 면은 움직임이 유사한 영역과 대응된다.



그림 4. 메쉬 단순화 예제

### 3. 메쉬 매개변수화

준균일 메쉬 시퀀스를 생성하기 위해서, 기본 메쉬와 주어진 입력 메쉬 시퀀스 간의 대응 관계를 구해야 한다. 즉, 메쉬 단순화 과정에서 에지 붕괴 연산자를 통해  $M^t$ 에서  $M^{t-1}$ 로 변형될 때,  $M^t$ 을 구성하는 모든 정점을  $M^{t-1}$  위로 대응시켜야 한다. 이를 위해서 본 논문에서는 MAPS[2]의 방법을 사용하였다. 즉, 메쉬 시퀀스를 구성하는 각 프레임 메쉬  $M_t^i$ 은  $M_t^{i-1}$ 로 단순화되면서, 두 메쉬 간의 대응 관계를 계산하였다.

### 4. 메쉬 세분화 및 리샘플링

준균일 메쉬를 생성하기 위해, 본 논문에서는 기본 메쉬를 구성하는 삼각형을 연속적으로 사분할한 후, 새로이 추가한 점을 기본 메쉬와 원본 메쉬 시퀀스 간의 대응 관계를 이용해 입력 메쉬 시퀀스의 표면에서의 위치를 계산하였다. 즉, 준균일 메쉬 시퀀스를 구성하는 각 준균일 프레임 메쉬의 위치는 원본 프레임 메쉬로부터 각각 독립적으로 얻어진다. 이렇게 얻어진 준균일 메쉬 시퀀스는 동일한 연결 관계를 가지고 있으며, 애니메이션 과정에서 정점의 위치만 이동하게 된다.

본 논문에서 제시한 메쉬 시퀀스의 재구성 결과는 [그림 5]과 [표 1]와 같다. 메쉬 시퀀스의 재구성 결과를 좀 더 자세히 보여 주기 위해 [그림 5]에서 우측 하단에 확대해서 표현하였으며, 같은 색상의 삼각형은 기본 메쉬의 하나의 삼각형을 분할함으로써 생성된 것들이다. 또한 [표 1]에서 [그림 5]의 모델에 대한 통계 자료로써, 입력 메쉬 시퀀스의 정점 수, 단순화를 통해 얻은 기본 메쉬의 정점 수, 메쉬 시퀀스를 구성하는 프레임의 수, 그리고 입력 메쉬 시퀀스와 준균일 메쉬 시퀀스 사이의 평균 오차를 나타내고 있다. 각 프레임에 해당하는 입력 메쉬와 준균일 메쉬 간의 거리를 측정하여, 가장 긴

거리를 오차로 정의하였으며, 각 메쉬의 경계 상자 (bounding box)의 대각선 길이에 대한 상대적인 길이로 표시하였다.

표 1. 메쉬 재구성 결과

모델	입력 메쉬 정점 수	기본 메쉬 정점 수	프레임 수	평균 에러 ( $10^{-4}$ )
Horse	8,431	80	48	3.63

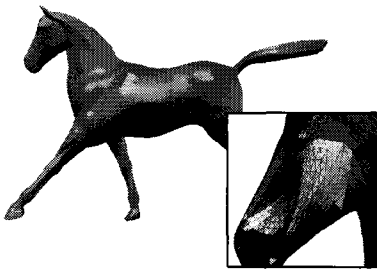


그림 5. 메쉬 재구성 결과

## V. 메쉬 시퀀스 압축

지금까지 입력 비균일 메쉬 시퀀스로부터 균일 메쉬 시퀀스를 얻는 방법에 대해서 설명하였다. 본 장에서는 이렇게 얻어진 균일 메쉬 시퀀스를 압축하는 알고리즘에 대해서 설명하도록 하겠다.

### 1. 압축 알고리즘

균일 메쉬 시퀀스  $\vec{M}$ 을 압축하는 과정은 크게 참조 프레임 메쉬 압축, 참조 프레임 메쉬와 다른 메쉬 프레임간의 차이값  $\Delta\vec{M} = \{\Delta\vec{M}_1, \dots, \Delta\vec{M}_T\}$  압축, 클러스터 정보 압축 등 세부분으로 이루어진다. 참조 프레임 메쉬 또는 첫 번째 프레임 메쉬 압축은 단순히 Khodakovsky가 제시한 방법 [1]을 사용하여, 압축 데이터  $\alpha_1$ 을 얻었다

참조 프레임 메쉬와의 차이값을 압축하기 위해서 각  $\vec{M}_i$ 에 대해 다음과 같은 과정을 반복해서 적용하여,  $\{\alpha_i = \{\beta_i, \gamma_i\}\}$ 을 얻는다. 이 때,  $\beta_i$ 는 차이값을 의미하고,  $\gamma_i$ 는 클러스터의 변환 행렬을 의미한다.

1. 각 정점  $v^i$ 에 대해, 해당하는 영역의 변환 행렬  $F_i^i$ 을 찾고, 수식 (1)을 이용하여,  $\Delta p_i^i$ 를 계산한다.
2. 균일 메쉬 압축 기법을 적용하기 위해  $\{\beta_i^i\}$ 로 이루어진 메쉬  $\Delta\vec{M}_i$ 를 구한다.
3.  $\Delta\vec{M}_i$ 를 웨이블릿 기반 압축을 통해  $\beta_i^i$ 를 얻는다.
4. 주어진 시간  $t$ 에서의 분할 영역의 변환 행렬의 모임  $\{F_i^i\}$ 를 압축하여  $\gamma_i$ 를 얻는다.

마지막으로, 각 정점마다 어떤 분할 영역에 속하는지 정보를 압축하여,  $\alpha_C$ 를 얻는다.

### 2. 압축해제 알고리즘

압축된 참조 프레임 메쉬 데이터  $\alpha_1$ 와 압축된 영역 분할 정보  $\alpha_C$ 를 압축 해제하고, 이를 이용해서  $\{\alpha_i = \{\beta_i, \gamma_i\}\}$ 를 압축 해제하는 과정은 다음과 같다.

1.  $\gamma_i$ 을 압축 해제하여, 각 분할 영역의 변환 행렬  $\{F_i^i\}$ 을 얻는다.
2. 각 정점  $v^i$ 에 대해, 참조 프레임 메쉬에서의 위치  $p_i^i$ 와 변환 행렬  $F_i^i$ 로부터  $\vec{p}_i = F_i^i \cdot p_i^i$ 를 계산한다.
3.  $\beta_i^i$ 를 압축해제 하여 차이값 메쉬  $\Delta\vec{M}_i$ 를 얻는다.
4. 각 정점  $v^i$ 에 대해,  $p_i^i = \vec{p}_i + \Delta p_i^i$ 를 계산하고,  $\vec{M}_i$ 를 얻는다.

## VI. 결과

이렇게 얻은 균일 메쉬 시퀀스를 압축한 결과는 [표 2]와 같으며, 참조 프레임 메쉬를 압축하기 위해서 12비트 평면 (bit-plane)과 Loop 웨이블릿을 사용하여 압축하였다. 또한 나머지 프레임 메쉬에 대해서는 8 비트 평면을 사용하여 압축하였다.

표 2. 메쉬 시퀀스 압축 결과

모델	입력 메쉬 정점 수	원본 데이터 크기 (byte)	참조 메쉬 압축 크기 (byte)	전체 압축 크기 (byte)	평균 에러 ( $10^{-4}$ )
Horse	8,431	5,058,372	27,723	304,617	5.74

**VII. 결론**

본 논문은 메쉬 재구성 기법을 이용하여 비균일 메쉬 시퀀스를 준균일 메쉬 시퀀스를 전환하고, 이를 압축하는 방법을 제안하였다. 특히 비균일 메쉬를 준균일 메쉬로 재구성하기 위해 모든 프레임의 정점 위치 및 이들의 움직임을 고려한 메쉬 단순화 방법을 제안하였다. 메쉬 시퀀스를 단순화하기 위해서 유사한 움직임을 가지는 영역별로 분할하였으며, 에지 붕괴 연산자를 적용하기 위한 새로운 이차 에러 기준을 정의하였다. 또한 각분할 영역의 변형 행렬을 이용하여 전체 분할 영역 내 정점의 이동 경로를 예측하였다.

본 논문에서 사용한 영역 분할 알고리즘은 삼각형의 회전 정보에 이용한 것으로써, 전체 프레임에 대해 모든 회전 정보를 동시에 고려하기 때문에, 계산 시간과 메모리 사용량에 있어서 문제가 있을 수 있다. 현재 이러한 문제를 해결하기 위해서 보다 효율적인 메쉬 분할 알고리즘에 대한 연구와 압축률을 높이기 위한 연구를 진행하고 있다. 또한 준균일 메쉬 시퀀스 생성을 통해 압축뿐만 아니라, 편집 등의 다양한 응용분야에도 활용하고자 한다.

**참고 문헌**

[1] A. Khodakovsky, P. Schröder, and W. Sweldens, "Progressive Geometry Compression," In Proc. ACM SIGGRAPH 2000, pp.271-278, 2000.

[2] A. W. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobskin, "MAPS: Multiresolution Adaptive Parameterization of Surfaces," In Proc. ACM SIGGRAPH 1998, pp.95-104, 1998.

[3] M. Garland and P. S. Heckbert, "Surface Simplification using Quadric Error Metrics," In Proc. ACM SIGGRAPH 1997, pp.209-216, 1997.

[4] D. L. James and C. D. Twigg, "Skinning Mesh Animations," ACM Trans. Graphics, Vol.24,

No.3, 2005.

[5] M. Sattler, R. Sarlette, and R. Klein, "Simple and Efficient Compression of Animation Sequence," In Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation(2005), pp.209-217, 2005.

[6] X. Gu, S. J. Gortler, and R. H. Hoppe, "Geometry Image," ACM Trans. Graphics, Vol.21, No.3, pp.355-361, 2002.

[7] H. M. Briceno, R. P. V. Sander, L. McMillan, S. Gortler, and H. Hoppe, "Geometry Videos: A New Representation for 3D Animations," In Proc. ACM SIGGRAPH/Eurographics Symposium on Computer animation(2003), pp.136-146, 2003.

[8] I. Guskov and A. Khodakovsky, "Wavelet Compression of Parameterically Coherent Mesh Sequence," In Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation (2004), pp.183-192, 2004.

[9] J. Lengyel, "Compression of time-dependent geometry," In Proc. ACM Symposium on Interactive 3D Graphics(1999), pp.89-95, 1999.

[10] M. Alexa and W. Müller, "Representing animations by principal components," Computer Graphics Forum, Vol.19. No.3, pp411-418, 2000.

**저자 소개**

안 민 수(Minsu Ahn)

정희원



- 2001년 2월 : 포항공과대학교 컴퓨터공학과(공학석사)
- 2007년 2월 : 포항공과대학교 컴퓨터공학과(공학박사)
- 2007년 4월 ~ 현재 : 삼성전자 중합기술원 전문연구원

<관심분야> : 메쉬 처리, 실시간 렌더링