

3차원 삼각형 메쉬 시퀀스 압축 기법

준회원 양 정 휴*, 종신회원 이 상 옥**

Compression of 3D Triangle Mesh Sequences

Jeong-Hyu Yang* Associate Member, Sang-Uk Lee** Life Member

요 약

3차원 데이터가 여러 분야에서 응용됨에 따라 방대한 크기의 데이터를 압축하기 위한 연구가 진행되고 있다. 그러나, 지금까지의 3차원 데이터 부호화에 대한 연구는 주로 정지 모델에 대해서 초점이 맞춰져 왔고, 3차원 시퀀스에 대한 연구는 아직까지 미비하다. 본 논문에서는 3차원 시퀀스 부호화를 위하여 간단하면서도 효율적인 알고리즘을 제안한다. 제안하는 알고리즘은 벡터 단위의 움직임 벡터 예측 기법을 이용하는 것으로서, 이웃하는 벡터의 움직임 벡터들의 높은 상관도를 이용한다. 이러한 높은 상관도를 이용하기 위하여 벡터의 이웃을 정의하고, 이웃의 움직임 벡터들로부터 벡터의 움직임 벡터를 예측한다. 제안한 알고리즘은 기존의 세그먼트에 기반한 방법보다 더 간단한 구조를 가지면서도 훨씬 우수한 압축 성능을 보여준다.

ABSTRACT

As 3D data is widely used for various applications, much attempts have been made to compress a huge amount of 3D data. Until now, however, most efforts have been focused on the compression of 3D static data, and little attempts to compress 3D sequence data have been made. In this paper, we propose a new geometry compression algorithm for 3D mesh sequences, based on the vertex-wise motion vector(MV) prediction scheme. In general, the MV of a vertex is highly correlated to those of adjacent vertices. To exploit high correlation, we define a neighborhood of a vertex, and predict the MV of the vertex from those of the neighborhood. It is shown that the proposed algorithm has simpler structure than the existing segment-based algorithm[10]. Furthermore, intensive simulation results demonstrate that the proposed algorithm yields better compression performance than the existing algorithm.

1. 서 론

최근, 3차원 데이터가 가상현실, 비디오 게임, CAD, 의료 영상 등과 같은 여러 응용 분야에서 사용됨에 따라 그 중요도가 점점 높아지고 있다. 그러나, 3차원 데이터를 표현하기 위해서는 일반적으로 상당히 많은 정보량이 필요하기 때문에 저장과 전송을 위해서는 막대한 저장 용량과 넓은 네트워크 대역폭이 필요하게 된다. 따라서 3차원 데이터를 효율적으로 압축하기 위한 연구가 필요하다. 3차원 데이터를 표현하는 방법 중에서 가장 널리 사용되는

방법 중의 하나가 삼각형 메쉬인데, 크게 정지 메쉬와 메쉬 시퀀스로 분류할 수 있다. 2차원 영상 데이터와의 유사성을 보면, 정지 메쉬는 정지 영상에 해당하고, 메쉬 시퀀스는 동영상에 해당한다.

지금까지의 3차원 메쉬 부호화에 있어서는 정지 메쉬 압축에 대부분의 노력이 기울어져 왔으며, 이에 대한 접근 방향으로는 전양자화 후의 무손실 부호화 방법^[1-4]과 점진적 부호화 방법^[5-8] 등이 있다. 메쉬 시퀀스의 경우, 3차원 애니메이션이나 모델링 기법들이 여러 분야에서 응용되고 있고, 실시간 3차원 영상 취득 장치 기술의 발달과 함께 많은 실제 3차원 데이터가 생성되리라 예상되지만, 부호화에

* 서울대학교 전기.컴퓨터공학부 신호처리연구실(jayu@ipl.snu.ac.kr), ** 서울대학교 (sanguk@ipl.snu.ac.kr)
논문번호 : 010346-1120, 접수일자 : 2001년 11월 20일

대한 연구 시도는 아직까지 미비하다^{10,14)}.

메쉬 시퀀스를 부호화하기 위해서는 이웃 프레임 간의 대칭점을 찾아야 하므로, 일반적인 메쉬 시퀀스의 부호화는 어려운 문제가 된다. 동형 시퀀스(isomorphic sequence)는 메쉬 시퀀스의 특수한 형태로서, 전체 시퀀스에 대해서 벡터스 개수가 일정하고 위상학적 정보가 유지되는 시퀀스이다. 동형 시퀀스에서는 이웃 프레임 간의 대칭점을 쉽게 찾을 수 있으며, 현재까지 제안된 알고리즘들은 이러한 동형 시퀀스에 국한하고 있다¹⁰⁻¹⁴⁾. 애니메이션 시퀀스는 대부분 동형 시퀀스이며, 실제 물체에 대한 시퀀스의 경우, 예를 들면, 물체 표면에 점을 표시하여 움직임을 추적하는 방법으로 얻을 수 있다¹¹⁾.

[13]에서 Beck 등은 객체 기반의 비디오 코딩 알고리즘을 제안하면서 2차원 객체를 동형 메쉬로 표현하고 2차원 메쉬 시퀀스 부호화를 수행하였다. 또한, Lengyel은 메쉬 분할(segmentation)에 바탕을 둔 동형 메쉬 시퀀스 부호화 알고리즘을 제안하였다¹⁰⁾. [10]에서 메쉬 시퀀스의 각 프레임은 먼저 몇 개의 세그먼트(segment)로 분할되고, 각 세그먼트들의 움직임은 유사(affine) 변환으로 표현된다. 유사 변환 계수와 예측 오차 값들은 양자화된 후 부호화된다. 이 방법은 2차원 비디오 영상 코딩 기법과 비슷하고, 메쉬 세그먼트는 비디오 영상의 매크로 블록에 해당된다. 비디오 영상 코딩에서 매크로 블록은 비디오 영상 신호의 균일한 특성 때문에 쉽게 얻을 수 있지만, 3차원 메쉬 시퀀스에서 메쉬 분할은 아주 복잡한 문제이며, 많은 복잡한 계산을 요구한다. Alexa 등은 선형 공간에 시퀀스 기하학 정보를 투영하여 주요 요소 분석법(principal component analysis)을 통해 동형 메쉬 시퀀스를 부호화하였다¹²⁾.

본 논문에서는 동형 메쉬 시퀀스에 대한 효율적인 부호화 알고리즘을 제안한다. 제안하는 기법은 동형 시퀀스의 특성을 충분히 이용함으로써 기존의 방법들¹⁰⁻¹⁴⁾에 비해서 장점을 가진다. 즉, 동형 시퀀스에서는 각 벡터스의 움직임을 정확하게 추적할 수 있으므로, 메쉬 세그먼트의 움직임 벡터가 아닌 개별 벡터스들의 움직임 벡터를 구함으로써, 복잡한 메쉬 분할 문제를 피할 수 있다. 그리고, 이웃한 벡터스들은 지역적 특성으로 인하여 공간적, 시간적으로 높은 상관도를 가지므로 이웃한 벡터스들의 움직임 벡터 정보를 이용함으로써 잉여 정보를 용이하게 제거할 수 있다. 또한, 벡터스 단위의 간단한 구조를 가지므로 실시간 구현이 용이하다. 본 논문

에서는 2단계로 예측이 이루어지는데, 먼저 이웃의 움직임 벡터들의 가중치 합으로 일차 예측이 이루어지며, Lagrange multiplier 기법을 도입하여 최적의 가중치를 구한다. 두 번째 단계에서는 일차 예측 후의 오차 벡터가 벡터스 주위의 형태 변화를 반영한다는 관찰을 통해서 이웃 벡터스의 오차 벡터들의 평균, 또는 자신의 이전 프레임에서의 오차 벡터를 통해서 이차 예측을 한다. 예측 오차는 양자화된 후 엔트로피 부호화된다. 부호화기에서는 벡터스 단위의 움직임 벡터 필드와 첫번째 프레임의 위상학적, 기하학적 정보를 가지고서 전체 메쉬 시퀀스를 복원할 수 있다. 제안한 알고리즘은 간단한 구조를 가지면서도 좋은 성능을 보여준다.

본 논문은 다음과 같이 구성된다. II장에서 시퀀스 부호화를 위한 프레임 내 부호화 기법과 프레임 간 부호화 기법을 다루고, III장에서는 실험 결과를 보여준다. 그리고, IV장에서 논문의 결론을 맺는다.

II. 3차원 메쉬 시퀀스 부호화

1. 문제 설정 및 가정

일반적으로, 3차원 물체의 표면과 그 움직임이 평활할 때, 표면의 기하학적 정보를 유한개의 이산적인 벡터스들을 가지고 근사화한다. 즉, 시간 t 의 프레임에서 N_t 개의 벡터스의 기하학적 정보와 연결 정보로 물체를 표현할 수 있다. 그런데, 이웃한 프레임 사이에는 상당히 많은 시간 잉여 정보가 있기 때문에 이러한 잉여 정보를 제거함으로써 3차원 물체 움직임 시퀀스를 표현하는 데에 필요한 데이터량을 상당히 줄일 수 있다. 일반적으로, 각 프레임의 벡터스는 개수와 위상학적 연결 정보가 각각 다르기 때문에 이웃한 프레임 사이의 벡터스나 세그먼트의 움직임 벡터를 찾는 것이 어려워진다. 그러나, 현존하는 3차원 메쉬 시퀀스에서는 벡터스의 개수와 연결 정보가 모든 프레임에 대해서 똑같이 일정한 동형 시퀀스가 대부분이다. 본 논문에서는 이러한 사실을, 일반적인 경우에 대한 하나의 가정으로 하여 메쉬 시퀀스의 각 프레임 사이에 존재하는 시간 잉여 정보를 압축하는 기법을 제안한다. 이러한 가정 하에서 메쉬 시퀀스는 첫번째 프레임과 벡터스 단위의 움직임 벡터 필드에 의해서 완전히 표현된다.

2. 3차원 메쉬 시퀀스 부호화 알고리즘의 구성

본 논문에서 제안하는 메쉬 시퀀스 부호화 알고리

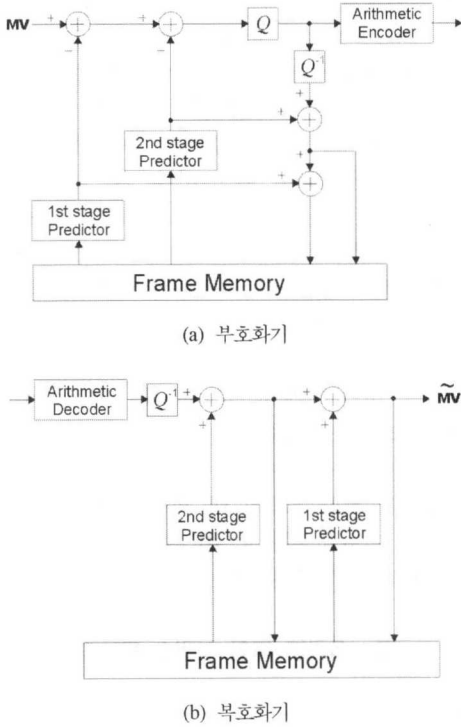


그림 1. 프레임간 부호화기 및 복호화기의 블록도

들은 프레임 내 부호화(Intra-frame coding) 기법과 프레임 간 부호화(Inter-frame coding) 기법으로 구성된다. 시퀀스의 첫번째 프레임에는 프레임 내 부호화 기법이 적용되며 기존의 정지 메쉬 부호화 기법을 사용한다^[2,4]. 여기서는 프레임 내의 공간적 상관도를 이용하여 기하학적 정보를 압축할 수 있으며, 버텍스 간의 위상학적 정보를 추출하게 된다. 두번째 프레임부터는 프레임 간 시간 잉여 정보를 제거하기 위해 본 논문에서 제안하는 프레임 간 부호화 기법이 적용되며 버텍스 단위의 움직임 벡터 필드를 예측하게 된다. 그림 1은 프레임 간 부호화를 위한 부호화기의 블록도를 보여준다.

3. 프레임 내 부호화

프레임 내 부호화를 위한 압축 알고리즘은 연결 정보 부호화 기법과 기하학적 정보 부호화 기법으로 구성된다. 연결 정보 부호화를 위해서 본 논문에서는 Touma와 Gotsman이 제안한 기법을 적용한다^[2]. 이 방법에서는 삼각형 메쉬에서 임의의 버텍스와 직접 연결되는 버텍스들은 일정한 방식에 의해서 순서가 정해질 수 있다는 사실을 이용하여, 특정한 순서에 따라 각 버텍스의 연결 버텍스 개수

(vertex degree)들의 리스트로서 연결 정보를 부호화한다. 'ADD', 'SPLIT', 'MERGE'의 세 가지 연산이 부호화 순서를 도와준다. 이 세 가지 연산들 중에서 'ADD' 연산이 가장 빈번하게 나타나고, 연결 버텍스 개수도 6을 중심으로 밀집되어 분포하는 특성이 있다. 이러한 불균일 분포 특성을 이용해서 산술 부호화 기법을 적용한다.

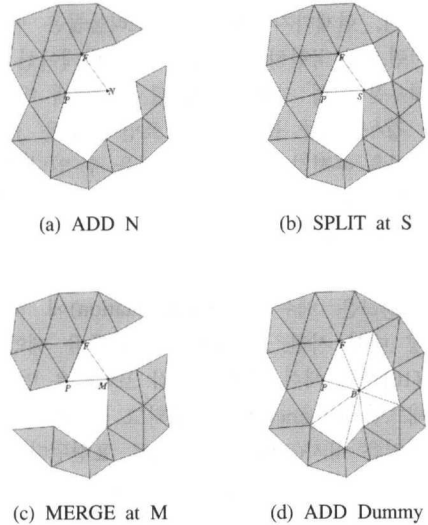


그림 2. 4가지 위상학적 패턴. F는 포커스, P는 이전에 부호화된 버텍스

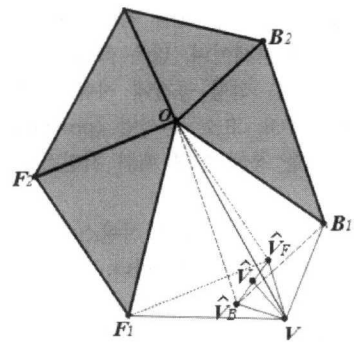


그림 3. 쌍방향 평행사변형 예측 기법.

그림 2는 부호화 과정에서 나타나는 4가지 위상학적 패턴의 예를 보여준다. 기하학적 정보 부호화를 위해서 각 버텍스의 위치는 쌍방향 평행사변형 기법에 의해서 예측된다^[4]. 이 기법은 앞 절의 연결 정보 부호화 기법이 내포하고 있는 삼각형 팬아웃(fanout) 구조를 기반으로 하는 것으로서, 쌍방향 예측이 가능한 버텍스에 대해서는 쌍방향 예측을 하

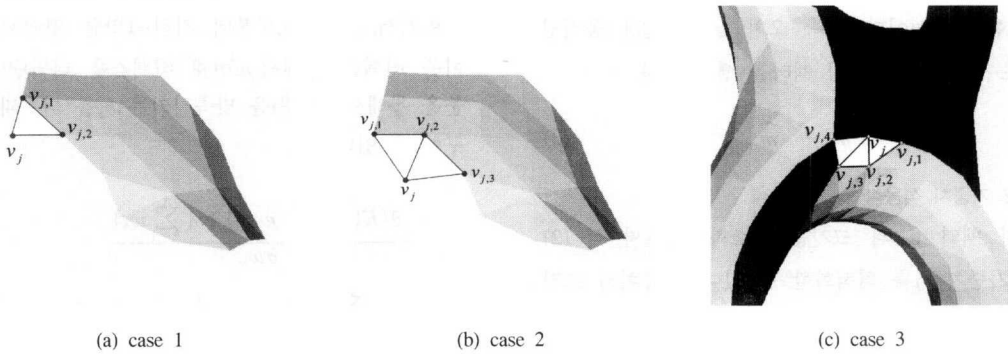


그림 4. 거리 1만큼 떨어져 있는 이웃의 예.

고, 그 외의 버텍스에 대해서는 순방향 예측을 한다. 이 방법은 순방향 평행사변형과 역방향 평행사변형을 이용함으로써 순방향 평행사변형만을 이용하는 경우보다 예측 오차의 분산을 줄여준다. 예측 오차는 양자화를 거친 후, 엔트로피 부호화를 한다. 예측 오차는 0을 중심으로 감소하며 퍼져나가는, 불균일한 확률 분포를 갖기 때문에 엔트로피 부호화가 좋은 성능을 보여준다. 그림 3은 쌍방향 평행사변형 예측이 적용되는 예를 보여준다.

4. 프레임 내 부호화

4.1 움직임 벡터 예측

버텍스의 움직임 벡터는 이웃의 움직임 벡터를 이용하여 예측된다. 버텍스 v 의 이웃은 기하학적으로 가까운 버텍스들이 아니라, v 로 부터 미리 정해진 거리 내에 위치해 있고 이미 부호화된 버텍스들의 집합이다. 각 프레임에 대해서 연결 정보는 변하지 않는다고 가정하므로, 각 버텍스의 이웃은 첫 번째 프레임의 부호화 과정 중에 찾을 수 있고, 전체 시퀀스에 대해서 고정된다. 두 개의 버텍스 v_1, v_2 사이의 위상학적 거리를 두 버텍스를 연결하는 최소의 에지 개수로 정의하고 $d(v_1, v_2)$ 로 나타내자. 그리고, j 번째 버텍스 v_j 가 부호화되고 있다고 가정할 때, v_j 로부터 거리 n 이내에 존재하는 이웃은 다음과 같이 정의된다.

$$N(v_j, n) = \{s | s \in E(j) \text{ and } d(v_j, s) \leq n\} \quad (1)$$

여기서 $E(j)$ 는 이미 부호화된 $(j-1)$ 개의 버텍스들의 집합을 나타낸다. 그림 4는 부호화 과정에서 나타나는, 거리 1만큼 떨어져 있는 이웃의 예를 보여준다. 그림 4(a)는 이웃이 2개인 경우이고, 그림

4(b)는 부호화되는 버텍스가 현재 포커스에서 마지막으로 부호화되는 경우, 이 정보를 통해서 암시적인 이웃을 알게 되는 경우이다. 그림 4(c)는 ‘SPLIT’, ‘MERGE’ 연산과 그 밖의 예외 처리에서 나타나는 경우로서 이웃에 속한 버텍스들이 부호화되는 버텍스를 기하학적으로 둘러싸는 경우이다. 그림 5는 점선 화살표로 표시된, 버텍스 v 의 움직임 벡터를 이웃의 정보를 이용해서 예측하는 것을 보여준다. 이 경우, 2개의 짧은 화살표는 거리 1의 이웃의 움직임 벡터를 나타내고, 3개의 가는 화살표는 거리 2의 이웃의 움직임 벡터를 나타낸다. 움직임 벡터의 상관성은 확률적으로 이웃의 거리가 멀리 떨어져 있을수록 줄어들기 때문에 본 논문에서는 거리가 1인 이웃만을 이용한다. 실제 실험을 통해서 이웃의 거리가 멀수록 부호화 성능이 나빠짐을 확인하였다.

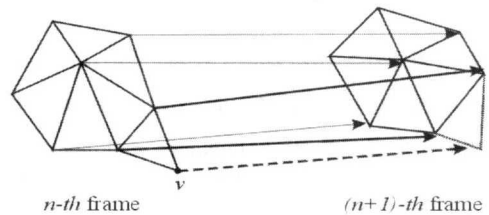


그림 5. 이웃 정보를 이용한 움직임 벡터 예측.

버텍스 v 의 실제 움직임 벡터와 복원된 움직임 벡터를 각각 m_v , \tilde{m}_v 라고 하자. 그러면, v_j 의 움직임 벡터는 식 (2)와 같이, 거리 N 까지의 이웃의 움직임 벡터의 가중치 합으로 예측한다.

$$\hat{m}_{v_j} = \sum_{u_i \in N(v_j, N)} w_k \tilde{m}_{u_i} \quad (2)$$

여기서 w_k 는 k 번째 이웃 버텍스 u_k 의 움직임 벡터

에 대한 가중치이다. 예측 오차 e_{v_j} 는 실제 움직임 벡터로부터 예측 움직임 벡터를 뺀 값이다.

$$e_{v_j} = m_{v_j} - \hat{m}_{v_j} \quad (3)$$

4.2 가중치 적응식

오차 벡터 e_{v_j} 의 크기를 최소화함으로써 식 (2)에서 가중치 w_k 를 최적화할 수 있다. 가중치의 조건은 다음과 같다.

$$\sum_{k=1}^K w_k = 1 \quad (4)$$

여기서 K 는 이웃의 개수이다. 식 (2)~(4)로부터 오차 벡터에 관한 식들을 다음과 같이 나타낼 수 있다.

$$\begin{aligned} e_{v_j} &= m_{v_j} - \sum_{k=1}^K w_k \tilde{m}_{v_{j,k}} \\ &= \sum_{k=1}^K w_k (m_{v_j} - \tilde{m}_{v_{j,k}}) \\ &= \sum_{k=1}^K w_k e_{v_{j,k}} \end{aligned} \quad (5)$$

$$e_{v_j}^T e_{v_j} = \sum_{m=1}^K \sum_{n=1}^K w_m w_n e_{v_{j,m}}^T e_{v_{j,n}} \quad (6)$$

그리고 $e_{v_j}^T e_{v_j}$ 의 기대값 $E\{e_{v_j}^T e_{v_j}\}$ 는 다음과 같다.

$$\begin{aligned} E\{e_{v_j}^T e_{v_j}\} &= \\ &= \sum_{m=1}^K \sum_{n=1}^K w_m w_n E\{e_{v_{j,m}}^T e_{v_{j,n}}\} \end{aligned} \quad (7)$$

결국, 가중치를 최적화하는 문제는 식 (4)의 제한 조건 하에서, 식 (7)의 $E\{e_{v_j}^T e_{v_j}\}$ 를 최소화하는 제한조건 문제와 같고, Lagrange multiplier 기법을 도입하면 다음과 같은 비제한조건 문제로 바뀐다.

$$\text{minimize } (E\{e_{v_j}^T e_{v_j}\} + \lambda \sum_{k=1}^K w_k). \quad (8)$$

문제를 간단하게 하기 위해서 이웃 버텍스들 간의 위상학적 거리에 따라 다음과 같은 가정을 할 수 있다.

$$\begin{aligned} E\{e_{v_{j,m}}^T e_{v_{j,n}}\} &= \\ &= \begin{cases} \alpha & \text{if } "d(v_{j,m}, v_{j,n}) = 0 \\ \beta & \text{if } "d(v_{j,m}, v_{j,n}) = 1 \\ \gamma & \text{if } "d(v_{j,m}, v_{j,n}) = 2 \end{cases} \\ & \text{for } \forall m, n, j. \end{aligned} \quad (9)$$

여기서, $v_{j,m}$ 은 v_j 로부터 거리 1만큼 떨어져 있는 이웃 버텍스 중에서 m 번째 버텍스를 나타낸다. 미분을 통해서 식 (8)을 만족시키는 w_k 를 다음과 같이 구할 수 있다.

$$\begin{aligned} \frac{\partial (E\{e_{v_j}^T e_{v_j}\} + \lambda \sum_{k=1}^K w_k)}{\partial w_m} & \\ &= 2 \sum_{n=1}^K w_n E\{e_{v_{j,m}}^T e_{v_{j,n}}\} + \lambda, \\ &= 0, \quad \text{for } 1 \leq m \leq K. \end{aligned} \quad (10)$$

식 (10)을 매트릭스로 표현하면 다음과 같다.

$$\begin{aligned} Aw &= \lambda' 1, \\ w &= \lambda' A^{-1} 1, \\ A &= (a_{mn}), \\ a_{mn} &= \begin{cases} \alpha & \text{if } d(v_{j,m}, v_{j,n}) = 0 \\ \beta & \text{if } d(v_{j,m}, v_{j,n}) = 1 \\ \gamma & \text{if } d(v_{j,m}, v_{j,n}) = 2 \end{cases} \\ w &= [w_1, w_2, \dots, w_K]^T, \\ \lambda' &= \frac{1}{2} \lambda, \\ 1 &= [1, 1, \dots, 1]^T. \end{aligned} \quad (11)$$

식 (4)의 조건에 따라 λ 가 정해지므로 w 는 다음과 같이 구할 수 있다.

$$\begin{aligned} w &= \frac{1}{S_{A^{-1}}} A^{-1} 1, \\ S_{A^{-1}} &= \text{sum of all elements in } A^{-1} \end{aligned} \quad (12)$$

4.3 오차 벡터의 예측

버텍스 v_j 의 이웃이 v_j 로부터 일정 거리 내에서 v_j 와 연결된 모든 버텍스들로 구성된다면 v_j 의 움직임 벡터는 앞 절에서와 같이 이웃 버텍스들의 움직임 벡터의 가중치 합으로 구하는 것이 최적이라고 할 수 있다. 그러나, 그림 4에서 보는 바와 같이, 버텍스 v_j 의 이웃은 이미 부호화가 된 버텍스들로 구성되기 때문에 이웃 버텍스들은 v_j 로부터 한쪽으로 편중된 위치에 분포하게 된다. 따라서 이웃 버텍스들의 움직임 벡터들은 버텍스 v_j 의 움직임을 충분히 반영하지 못하게 되므로 앞 절에서와 같은 예측 후의 오차 벡터들은 여전히 잉여 정보를 지니게 된다.

먼저 오차 벡터의 기하학적 의미를 알아보자. 이 산적인 시간 n 과 $(n+1)$ 에서, 버텍스 v_j 와 거리 1만큼 떨어진 이웃의 m 번째 버텍스 $v_{j,m}$ 의 좌표 벡

터가 각각 $c_{v_j}(n)$, $c_{v_{j,m}}(n)$ 에서 $c_{v_j}(n+1)$, $c_{v_{j,m}}(n+1)$ 로 변한다고 하자. 두 버텍스의 움직임 벡터의 차이 벡터 $e_{v_{j,m}}$ 는 다음과 같다.

$$\begin{aligned} e_{v_{j,m}} &= (c_{v_j}(n+1) - c_{v_j}(n)) \\ &\quad - (c_{v_{j,m}}(n+1) - c_{v_{j,m}}(n)) \\ &= (c_{v_j}(n+1) - c_{v_{j,m}}(n+1)) \\ &\quad - (c_{v_j}(n) - c_{v_{j,m}}(n)) \end{aligned} \quad (13)$$

즉, 식 (13)에 의해서 움직임 벡터의 차이 벡터는 두 에지 벡터의 차이 벡터와 같음을 알 수 있다.

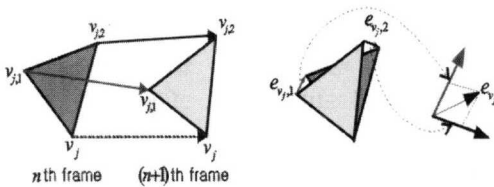


그림 6. 오차 벡터의 예.

그림 6은 이웃의 개수가 2인 경우의 오차 벡터의 예를 보여준다. 한 버텍스와 거리 1만큼 떨어진 이웃 버텍스를 연결하는 에지를 이웃 에지라고 하자. 식 (3)의 버텍스 v_j 의 오차 벡터 e_{v_j} 는 이웃 에지 벡터들의 시간적인 차이 벡터들의 가중치 합 중 크기가 최소인 벡터라고 할 수 있다. 즉, e_{v_j} 는 이웃 에지 벡터들의 시간적 변이를 반영하며, 이는 곧 버텍스 v_j 주위의 지역적인 형태 변화를 나타낸다.

$E\{ e_{v_{j,m}}^T e_{v_{j,n}} \}$ 는 모든 j, m, n 에 대해서 두 버텍스의 쌍 $(v_{j,m}, v_{j,n})$ 을 두 버텍스의 위상학적 거리에 따라 분류하여 평균하고, 시간적으로도 평균하므로 프레임이 증가함에 따라 이웃한 프레임 간의 $E\{ e_{v_{j,m}}^T e_{v_{j,n}} \}$ 의 변화는 매우 작다고 가정할 수 있으며 따라서 가중치의 변화도 작다고 할 수 있다. 따라서 오차 벡터의 예측에 이용할 수 있는 중요한 가정은 3차원 물체의 움직임이 평활하여 물체의 변형이 선형적으로 이루어진다는 가정과, 공간적인 지역성에 의해서 인접한 버텍스 간의 형태 변형은 상당한 상관도가 있다는 가정이다. 첫 번째 가정에 의해서, 이전 프레임에서의 오차 벡터들과 현재 프레임에서의 오차 벡터들의 상관도를 구하고 모든 버텍스에 대해서 평균을 하여 예측 방식을 결정한다. 두개의 문턱치 $t_1, t_2 (t_1 < t_2 < 1)$ 에 대해서 상관도가 t_2 보다 클 경우, 이전 프레임에서의 오차 벡터로 현재

프레임의 오차 벡터를 예측하고, 상관도가 t_1, t_2 사이 값일 때는 예측을 하지 않는다. 그리고 상관도가 t_1 보다 작을 경우는 두 번째 가정에 의해서 이웃 버텍스들의 오차 벡터들을 평균하여 다음과 같이 예측한다.

$$\hat{e}_{v_j} = \frac{1}{K} \sum_{u_k \in N(v_j, 1)} \tilde{e}_{u_k} \quad (14)$$

여기서 K 는 이웃의 개수이고 \tilde{e} 는 복원된 오차 벡터를 나타낸다. 예측 후의 오차 r_{v_j} 는 다음과 같으며, \tilde{r}_{v_j} 로 양자화된 후에 산술부호화 기법에 의해서 부호화된다.

$$r_{v_j} = e_{v_j} - \hat{e}_{v_j} \quad (15)$$

복원된 움직임 벡터와 오차 벡터는 다음과 같이 표현된다.

$$\begin{aligned} \tilde{m}_{v_j} &= \hat{m}_{v_j} + \tilde{e}_{v_j}, \\ \tilde{e}_{v_j} &= \hat{e}_{v_j} + \tilde{r}_{v_j}. \end{aligned} \quad (16)$$

5. 양자화와 엔트로피 부호화

프레임 내 부호화와 프레임 간 부호화 과정에서 생긴 예측 오차는 데드존(dead-zone) 스칼라 양자화기를 사용해서 균일하게 양자화되고, 산술부호화된다. 양자화 스텝 간격은 첫번째 프레임에서 결정되며, N_s 비트로 첫 번째 프레임의 좌표값의 전체 범위를 표현할 수 있도록 정해진다.

예측 오차는 0 주위에 모이는 특성이 있기 때문에 데드존 기법과 산술부호화 기법은 만족할만한 결과를 보여준다. 효율적인 엔트로피 부호화를 위해서 영벡터 (0,0,0)와 영벡터 주위의 32개의 좌표 벡터는 부호책에 저장된다. 나머지 스칼라 오차값들은 미리 정해진 문턱치보다 작을 경우에는 산술 부호화가 되고, 그렇지 않을 경우에는 고정 비트율로 부호화된다.

III. 실험 결과

1. 평가 척도: SNR

실제 모델 데이터와 부호화된 후에 복원되는 데이터의 차이를 비교하기 위해서 평가 척도 SNR을 다음과 같이 정의한다.

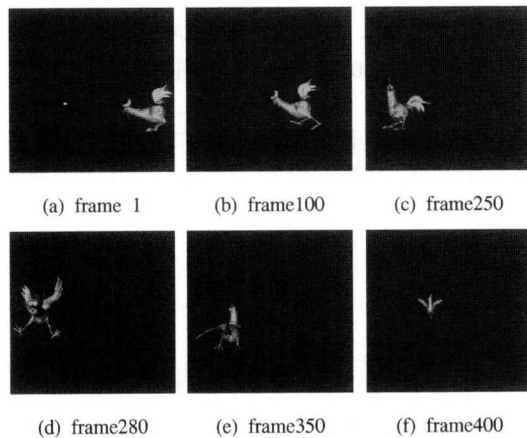
$$10 \log_{10} \frac{\sum_{m=1}^{F_{Total}} \sum_{n=1}^{N_{Total}} \| \mathbf{c}_{v_n}^m \|^2}{\sum_{m=1}^{F_{Total}} \sum_{n=1}^{N_{Total}} \| \mathbf{c}_{v_n}^m - \hat{\mathbf{c}}_{v_n}^m \|^2} \quad (17)$$

여기서 $\| \cdot \|$ 는 벡터의 크기, $\mathbf{c}_{v_n}^m$ 는 m 번째 프레임의 n 번째 버텍스의 좌표 벡터, $\hat{\mathbf{c}}$ 는 예측값을 나타낸다. 그리고, F_{Total} 과 N_{Total} 은 각각 전체 프레임 수와 전체 버텍스 개수이다. 두 데이터 간에 서로 상응하는 버텍스는 부호화기에서 부호화하는 과정에서 알 수 있고, SNR은 부호화 과정에서 계산한다.

2. 성능 평가

제안하는 알고리즘의 성능을 평가하기 위하여 2가지 메쉬 시퀀스 ‘chicken’과 ‘bounceball’ 모델을 가지고 실험을 하였다. ‘chicken’ 모델은 Lengyel이 제안한 알고리즘^[10]에서 쓰였던 실험 시퀀스로서 Softimage라는 애니메이션 툴에서 제작된 모델이고, ‘bounceball’ 모델은 3D Max Studio 애니메이션 툴에서 얻어진 것이다. 실험 모델의 데이터 구조를 보면, ‘chicken’ 모델은 400프레임 시퀀스로서 각 프레임은 3,030개의 버텍스와 5,665개의 삼각형으로 이루어져 있으며 전체 데이터를 표현하기 위해서 압축하지 않을 때, 14,577,900바이트가 필요하다. 그리고, ‘bounceball’ 모델은 67 프레임 시퀀스로서, 각 프레임에는 162개의 버텍스와 320개의 삼각형이 있고 이를 표현하기 위해서 132,168바이트가 요구된다.

그림 7과 그림 8은 각각 ‘chicken’ 모델과 ‘bounceball’ 모델의 일부 프레임을 보여준다. 실험을 위해서 양자화를 위한 N_s 는 6에서 12까지 변화



를 주었다. 그림 9는 거리가 1인 이웃에 대한 정보를 가지고 실험한 결과를 보여준다. 그림 9(a)는 ‘bounceball’ 시퀀스에 대한 결과이며, 그림 9(b)는 ‘chicken’ 시퀀스에 대한 결과이다. 각각에 대해서 오차 벡터에 대한 예측을 하는 경우와 하지 않는 경우에 대한 실험 결과가 제시되어 있다. 살펴보면, ‘bounceball’ 시퀀스에 대해서는 오차 벡터에 대한 예측을 하는 경우가 그렇지 않는 경우에 비해서, 같은 SNR에 대해서 8퍼센트 정도까지의 부호화 이득이 있고, ‘chicken’ 시퀀스의 경우는 17퍼센트 정도까지의 부호화 이득이 있다. 한편, ‘bounceball’ 시퀀스는 빠른 움직임과 형태 왜곡으로 프레임 간의 형태 변화가 많이 나타나는 특성을 가지기 때문에, 이웃의 움직임 벡터의 가중치 합으로 예측을 한 후의 오차 벡터가 크다. 그러나, 프레임 간 오차 벡터의 상관도가 작으므로 오차 벡터 예측은 대부분 이웃 버텍스들의 오차 벡터들의 평균으로 이루어진다. 빠른 움직임 때문에 전체적인 압축 효율은 좋지 않다. 반면, ‘chicken’ 시퀀스는 프레임 간 움직임이 매끄럽기 때문에 형태 변화가 거의 없거나 선형적인 형태 변화를 보여준다. 따라서 가중치를 통한 일단계의 예측으로 충분하거나, 프레임 간 오차 벡터의 상관도가 큰 경우 이전 프레임에서의 오차 벡터를 통해서 예측이 이루어지는 경우가 대부분이다.

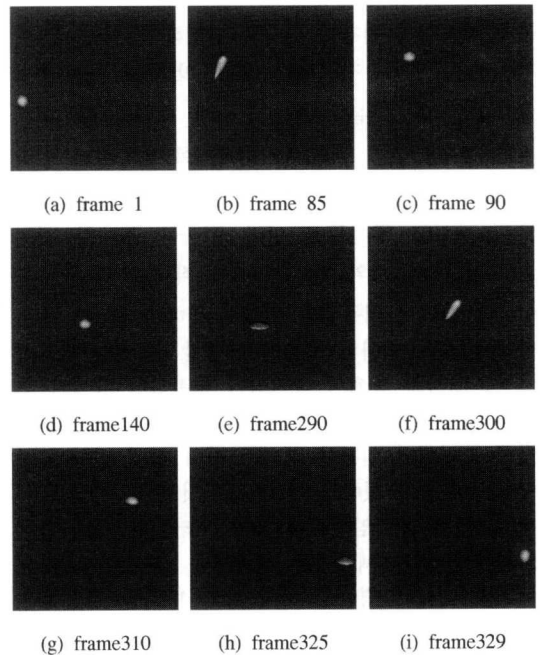


그림 9(b)에서 세그먼트에 기반한 알고리즘^[10]의 결과를 포함했다. 기존의 논문들이 비트 할당량 대비 성능 비교 결과를 제대로 제시하지 않고 있으므로 [10]의 결과만 포함하였다. [10]에서는 양자화 후 엔트로피 부호화가 이루어지지 않았다. 이런 조건하에서 비교를 해보면, SNR을 고정할 때, 제안한 알고리즘이 세그먼트에 기반한 방법보다 최소 3배 이상의 더 적은 비트율이 필요함을 주목할 수 있다. 주요한 성능 향상은 엔트로피 부호화 기법의 적용으로부터 온 것이고, 또 다른 향상 요인으로는 부호화 과정에서 이미 알 수 있는 정보를 이용해서 부호화가 이루어지고, 유사 변환과 같은 예측 변환 등의 기법을 이용하지 않으므로 계수를 부호화 할 필요가 없이 예측 오차만 부호화 한다는 점을 들 수 있다. 그리고, 제안한 알고리즘은 많은 계산을 요구하는 메쉬 분할 과정을 요구하지 않으므로 [10]의

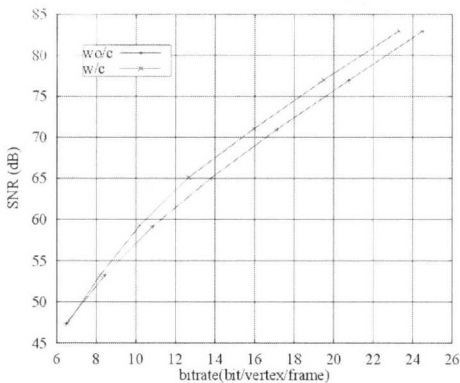
방법보다 훨씬 간단하다.

IV. 결론

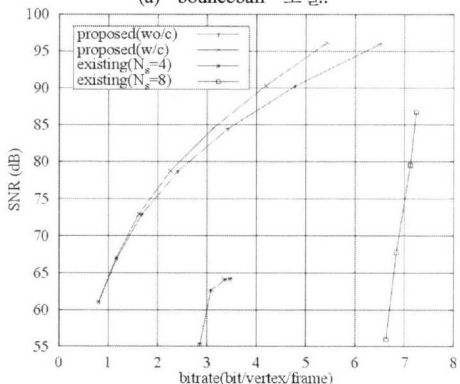
3차원 메쉬 시퀀스는 시간 잉여 정보가 많기 때문에 잉여 정보를 제거해 줌으로써 데이터를 효과적으로 압축할 수 있다. 본 논문에서는 벡터스 단위의 움직임 예측기법을 이용한 3차원 메쉬 시퀀스 부호화 알고리즘을 제안하였다. 시퀀스의 프레임에 구성하는 벡터스의 개수가 일정하고, 위상학적 정보도 변하지 않는다는 가정 하에서 각 벡터스의 움직임 벡터는 쉽게 예측할 수 있다. 움직임 벡터들 사이의 상관성을 이용하기 위하여, 벡터스의 이웃을 정의하고 이웃의 움직임 벡터들의 가중치 합으로부터 벡터스의 움직임 벡터를 예측하였다. 그리고, Lagrange multiplier를 도입하여 오차가 최소가 되도록 가중치를 구했다. 그리고, 물체의 형태 변화를 반영하는 오차 벡터를 이웃의 오차 벡터와 이전 프레임에서의 오차 벡터를 통해서 예측하였으며, 오차 벡터의 예측은 좋은 성능을 보여주었다. 실험 결과는 제안한 알고리즘이 기존 방법^[10]보다 성능이 우수함을 보여주었다.

참고 문헌

- [1] G. Taubin and J. Rossignac, "Geometric compression through topological surgery," *Research Report* RC-20340, IBM Research Division, 1996.
- [2] C. Touma and C. Gotsman, "Triangle mesh compression," in *Proc. Graphics Interface '98*, pp. 26-34, 1998.
- [3] J. Li and C.-C. J. Kuo, "A dual graph approach to 3D triangular mesh compression," in *Technical Report*, Contribution Document M3195, MPEG4 San Jose Meeting, 1998.
- [4] 심재영, 이상욱, "메쉬 구조에 기반한 3차원 데이터의 부호화 기법에 관한 연구," *서울대학교 대학원 석사학위 논문*, 2001.
- [5] H. Hoppe, "Progressive meshes," in *Proc. Siggraph '96*, pp. 99-108, 1996.
- [6] J. Li and C.-C. J. Kuo, "Progressive coding on 3D graphic models," in *Proc. IEEE*, 86:1052-1063, 1998.
- [7] G. Taubin, A. Guezic, W. Horn and F.



(a) 'bounceball' 모델..



(b) 'chicken' 모델.

그림 9. 2가지 실험 시퀀스에 대한 성능. Ns는 양자화 비트이고, 'w/ep'와 'wo/ep'는 각각 'with error vector prediction'과 'without error vector prediction'을 나타낸다.

Lazarus, "Progressive Forest Split Compression," in *Proc. Siggraph'98*, pp. 123-132, 1998.

[8] A. Khodakovsky, P. Schroder and W. Sweldens, "Progressive Geometry Compression," in *Proc. Siggraph'2000*, pp. 271-278, 2000.

[9] P. Alliez and M. Desbrun, "Progressive Compression for Lossless Transmission of Triangle Meshes," in *Proc. Siggraph'2001*, 2001.

[10] J. E. Lengyel, "Compression of Time-Dependent Geometry," in *Proc. 1999 ACM Symposium on Interactive 3D Graphics*, pp. 89-96, 1999.

[11] B. Guenter, C. Grimm, and D. Wood, "Making Faces," in *Proc. Siggraph'98*, pp. 55-66, 1998.

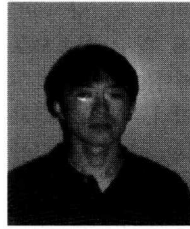
[12] M. Alexa, and W. Muller, "Representing Animations by Principal Components," in *Eurographics'2000*, vol.19, 2000.

[13] P. J. L. Van Beek, A. M. Tekalp, and A. Puri, "2-D Mesh geometry and motion compression for efficient object-based video representation," in *Proc. ICIP*, pp. 440-443, vol.3, 1997.

[14] 안정환, 임동근, 호요성, "삼차원 애니메이션 모델의 분할 및 부호화 방법," 제 13회 신호처리 합동 학술대회 논문집, 제 13권, 1호, pp. 153-156, 2000.

양 정 휴(Jeong-Hyu Yang)

준회원

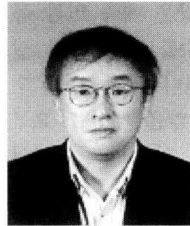


1998년 2월 : 서울대학교 전기공학부 졸업
 2000년 2월 : 서울대학교 전기컴퓨터공학부 석사
 2000년 3월~현재 : 서울대학교 전기컴퓨터공학부 박사과정

<주관심분야> 신호처리, 영상처리

이 상 옥(Sang-Uk Lee)

종신회원



1973년 : 서울대학교 전기공학과 졸업
 1976년:Iowa State University 전기공학 석사
 1980년:University of Southern California 전기공학 박사

1980년~1981년:General Electric Company
 1981년~1983년:M/A-COM Research Center
 1983년~현재 : 서울대학교 전기컴퓨터공학부 교수
 <주관심 분야> 신호처리, 영상처리, 컴퓨터비전