

논문-04-09-1-07

3차원 메쉬 모델을 위한 강인한 워터마킹 기법

송한새*, 조남익*, 김종원**

Robust Watermarking Algorithm for 3D Mesh Models

Han Sae Song*, Nam-Ik Cho* and JongWeon Kim**

요 약

본 논문에서는 3차원 메쉬 모델에 적용되는 강인한 워터마킹 알고리즘을 제안한다. 제안하는 알고리즘에서 워터마크는 3차원 모델로부터 추출된 2차원 영상에 삽입된다. 이 2차원 영상의 화소값은 정해진 기준점들로부터 3차원 모델의 표면까지의 거리이며, 이를 거리 영상이라 한다. 워터마크는 거리 영상에 삽입되며, 워터마크된 거리 영상을 이용하여 3차원 모델의 꼭지점 좌표를 변경하면 워터마크가 삽입된 3차원 모델을 얻는다. 워터마크의 추출은 워터마크가 삽입된 모델로부터 거리영상을 얻고, 이 영상에서 워터마크를 추출한다. 워터마크 추출에는 원본 모델이 필요하며 원본과의 정합이 필요하다. 실험을 통해 제안하는 알고리즘이 회전, 병진, 비례조절, 가우스 잡음, 메쉬 간략화, 꼭지점 양자화에 강인함을 확인하였다.

Abstract

A robust watermarking algorithm is proposed for 3D mesh models. Watermark is inserted into the 2D image which is extracted from the target 3D model. Each pixel value of the extracted 2D image represents a distance from the predefined reference points to the face of the given 3D model. This extracted image is defined as "range image" in this paper. Watermark is embedded into the range image. Then, watermarked 3D mesh is obtained by modifying vertices using the watermarked range image. In extraction procedure, the original model is needed. After registration between the original and the watermarked models, two range images are extracted from each 3D model. From these images, embedded watermark is extracted. Experimental results show that the proposed algorithm is robust against the attacks such as rotation, translation, uniform scaling, mesh simplification, AWGN and quantization of vertex coordinates.

Keywords : Robust watermarking, 3Dmesh

I. 서 론

3차원 메쉬 모델은 3차원 물체를 기술하는 여러 가지 방식 중에서 현재 널리 쓰이고 있는 방식이다. 3차원 메쉬는

기본적으로 2가지 요소, 꼭지점의 좌표와 연결 정보로 구성된다. 꼭지점 좌표는 해당 꼭지점의 3차원 공간상의 위치를 기술하며, 연결 정보는 꼭지점 사이에 모서리들의 연결을 나타낸다. 이 두가지 요소가 결합되어 3차원 메쉬 모델의 면을 이루게 된다. 그리고 삼각형의 면을 모델의 표면을 구성하는 기본 단위로 사용하는 삼각형 메쉬가 간단하고 처리가 쉬워서 많이 사용되고 있다.

워터마킹은 대상체에 정보를 숨겨서 여러 가지 목적으로 사용하기 위한 기술이다. 3차원 메쉬 모델에 대한 워터마킹

* 서울대학교 전기공학부, 뉴미디어통신공동연구소
School of Electrical Engineering Seoul National University

** 마크애니 연구소
MarkAny Research Institute

기법도 여러 연구가 진행되어 왔다. Ohbuchi는 다각형 메쉬에 대한 간단한 워터마킹 기법과 NURBS에 대한 워터마킹 기법을 제안하였다^{[1][2]}. Benedens는 수직 벡터를 이용하여 메쉬 간략화에 강인한 워터마킹을 제안하였다^[3]. 또한 어파인(affine) 변환에 강인한 워터마킹을 제안하였고, 대용량 워터마킹 기법도 제안하였다^{[4][5]}. Wagner^[6]는 어파인 변환에 강인한 워터마킹을 제안하였고, Praun^[7]은 영상 워터마킹에서 널리 쓰이는 확산 대역 방식을 확장하여 메쉬 간략화 등 여러 가지 공격에 강인한 기법을 제안하였다. Yeo^[8]는 3차원 메쉬의 무결성 검사를 위한 기법을 제안하였다. 위의 방법들은 모두 공간 영역에서 워터마킹을 삽입하는 기법들이며, 주파수 영역에서 워터마킹을 하는 기법 역시 제안되었다. Kanai^[9]는 웨이블릿 변환을 이용한 워터마킹을 제안하였고, Ohbuchi^[10]는 라플라시안 행렬을 사용하여 주파수 영역 메쉬 워터마킹을 제안하였다.

워터마킹 기법 중에서 강인한 워터마킹 기법은 숨겨진 정보를 가지고 있는 대상체에 여러 공격이나 조작을 가한 후에도 숨은 정보가 사라지지 않도록 하는데 그 목적이 있다. 3차원 메쉬 모델에 가해질 수 있는 조작은 무수히 많으나 일반적인 조작들로는 어파인 변환, 메쉬 간략화, 꼭지점 좌표 양자화 등이 있다. 의도적으로 가해질 수 있는 공격으로는 잠음 첨가, 꼭지점 재배열, 재 워터마킹 등이 있다.

3차원 메쉬에 정보를 숨기기 위해서 다양한 기법이 존재할 수 있으나, 결국 꼭지점 좌표나 연결 정보를 변경해서 정보를 전달해야 한다. 이 두 가지 구성요소는 3차원 메쉬 모델이라면 항상 가지고 있는 정보이고, 가장 중요한 정보이기 때문이다. 그런데 워터마크를 단순히 꼭지점 좌표의 최하위 비트를 변경하거나, 각 꼭지점을 조금씩 변경하는 조작만으로는 강인한 워터마킹의 목적을 달성할 수 없다. 왜냐하면 꼭지점의 좌표나 연결정보는 간단한 조작만으로도 쉽게 변형되기 때문이다. 꼭지점 좌표는 회전 변환이나 병진 혹은 비례조절 등, 3차원 메쉬에 일상적으로 가해지는 조작에도 그 좌표값이 완전히 바뀌게 된다. 또한 메쉬 간략화의 경우에는 꼭지점의 좌표와 개수 그리고 연결정보까지 모두 바뀌게 된다. 메쉬 간략화 기법들이 최대한 원본의 모양을 유지하면서 꼭지점의 개수를 줄이는데 그 목적이 있기 때문에 3차원 메쉬의 두 구성 요소가 완전히 바뀌어도 불구하고 그 모양은 크게 손상되지 않는다. 따라서 메쉬 간략화는 강인한 워터마킹에 대한 강력한 공격 중에 하나라고 할 수 있다.

워터마킹 기법이 공격에 강인하기 위해서 워터마크를

3차원 모델의 꼭지점 좌표나 연결정보에 직접 삽입하지 않고 3차원 모델에서 공격에 강인한 성질을 가지는 특징을 추출하여 워터마크를 삽입하거나 3차원 모델을 다른 영역으로 변환하여 워터마크를 삽입하게 된다. 예를 들면, Benedens^[3]의 알고리즘에서는 3차원 메쉬의 수직 벡터들이 메쉬 간략화 전후에 크게 변하지 않는 성질을 이용하여 워터마크를 삽입하며, Kanai^[9]의 알고리즘에서는 3차원 모델을 주파수 영역으로 변환하여 워터마크를 삽입한다.

본 논문에서 제안하는 기법도 공격에 강인한 성질을 가지는 특징을 추출하여 워터마크를 삽입하는 방식을 사용한다. 구체적으로, 제시한 방법에서는 3차원 모델로부터 2차원 영상을 추출하고 여기에 워터마크를 삽입한다. 이 영상의 화소값은 정해진 기준점들로부터 3차원 모델의 표면까지의 거리값이며 이 영상을 거리영상이라고 정의하였다. 거리영상은 기준점들로부터 3차원 모델의 표면까지의 거리값들로 이루어져 있으므로 3차원 메쉬가 공격을 받아서 꼭지점이나 연결이 변하더라도 그 모양이 심하게 손상되지 않는다면 거리영상 역시 크게 변하지 않으며 삽입된 워터마크도 사라지지 않게 된다.

그림 1은 전체 알고리즘의 흐름도이다. 먼저 3차원 메쉬 모델로부터 거리영상을 얻고 여기에 워터마크를 삽입한다. 그 후에 워터마크가 삽입된 거리영상을 이용하여 워터마크를 3차원 메쉬의 꼭지점에 반영하게 된다. 워터마크의 추출 과정은 원본을 필요로 하며, 원본과의 정합이 필요하다.

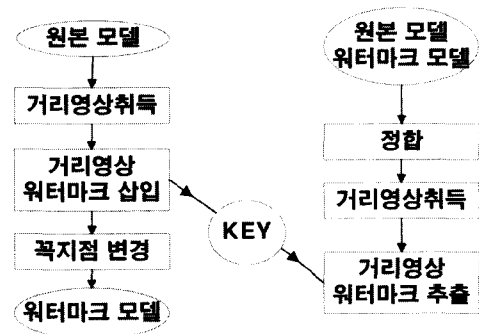


그림 1. 알고리즘 블록도
Fig 1. Block diagram of the algorithm

알고리즘은 크게 세 부분으로 나뉜다. 3차원 메쉬로부터 거리영상을 얻는 과정, 거리영상에 워터마킹을 하고 이를 3차원 모델의 꼭지점에 반영하는 방법, 그리고 워터마크 추

출 방법이며 제 2장에서 다루게 된다. 제 3장에서는 실험결과가 소개되고, 제 4장에서 결론을 맺는다.

II. 제안된 방법

1. 거리영상

거리영상의 화소값은 3차원 모델 주위의 기준점들로부터 모델의 표면까지의 거리이다. 기준점들은 3차원 모델을 감싸는 원기둥 옆면에 일정한 간격으로 위치하게 된다. 먼저 원기둥의 위치, 반지름과 높이를 구하는 방법을 설명하고 기준점의 위치에 대해 설명한다. 그리고 각 기준점으로부터 3차원 모델의 면까지 거리를 구하는 방식을 설명한다. 앞으로의 수식에서 굵은 글자체의 소문자는 모두 3차원 열벡터를 의미한다.

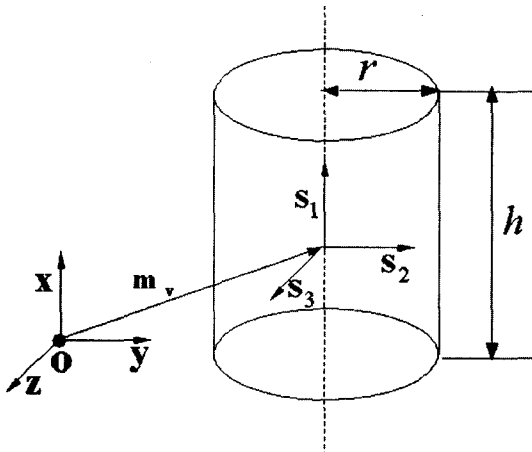


그림 2. 거리측정 기준점을 위한 원기둥
Fig 2. Cylinder for reference points

1.1 원기둥의 위치 설정

원기둥은 그림 2와 같이 3차원 상의 한 점 m_v 를 중심으로 하여 원기둥의 축은 s_1 방향이며 반지름은 r , 높이는 h 이다. m_v 는 3차원 메쉬의 중심점으로 꼭지점 좌표들의 평균이다. 원기둥의 세 기준축은 3차원 모델로부터 꼭지점 좌표의 공분산 행렬을 구하고, 이 행렬의 고유 벡터 (eigen-vector)를 구하여 이를 기준축으로 한다. 공분산 행렬은 다음과 같이 구한다.

$$C = \frac{1}{n} \sum_{i=1}^n (v_i - m_v)(v_i - m_v)^T \quad (1)$$

v_i 는 i 번째 꼭지점 좌표이며, n 은 3차원 모델의 꼭지점의 개수이다. s_1 벡터는 C 의 고유 벡터 중 가장 큰 고유치 (eigen value)를 갖는 벡터이며, s_2 가 두 번째로 큰 고유치를 갖는 벡터이다. 그런데 고유 벡터의 계산 방법에 따라 방향이 반대가 될 수도 있으므로 일정한 방향을 유지하기 위해 3차원 모델의 꼭지점 좌표의 분포도를 이용하여 방향을 결정해 준다^[11]. 집합 X 에 속한 꼭지점 좌표들의 분포도는 다음과 같이 정의한다.

$$D_x = \sqrt{\frac{1}{n} \sum_{v \in X} \|v - m_v\|^2} \quad (2)$$

먼저, s_1 의 방향을 정하기 위해서 3차원 모델의 꼭지점을 두개의 집합 A와 B로 나눈다.

$$\begin{aligned} A &= \{v : s_1^T (v - m_v) > 0\} \\ B &= \{v : s_1^T (v - m_v) < 0\} \end{aligned} \quad (3)$$

이때 각 집합의 분포가 $D_A < D_B$ 이면 s_1 의 방향을 바꾸고 아니면 그대로 둔다. s_2 축은 식 (3)에서 s_1 대신 s_2 를 사용하여 집합 A와 B를 구하고 마찬가지로 방향을 정한다. $s_3 = s_2 \times s_1$ 으로 한다. 그리고 s_1, s_2, s_3 는 크기를 1로 정규화한다. 원기둥의 반지름과 높이는 원기둥이 3차원 메쉬를 완전히 포함하면서 가장 작은 크기가 되도록 정해진다. 식 (4)과 같이 반지름 방향으로 가장 먼 꼭지점까지의 거리를 r 로 하며 s_1 방향과 그 반대방향으로 가장 먼 꼭지점으로부터 높이를 구한다.

$$\begin{aligned} r &= \max_{v \in V} \sqrt{s_2^T (v - m_v)^2 + s_3^T (v - m_v)^2} \\ h &= |h_{max}| + |h_{min}| \\ h_{max} &= \max_{v \in V} s_1^T (v - m_v) \\ h_{min} &= \min_{v \in V} s_1^T (v - m_v) \end{aligned} \quad (4)$$

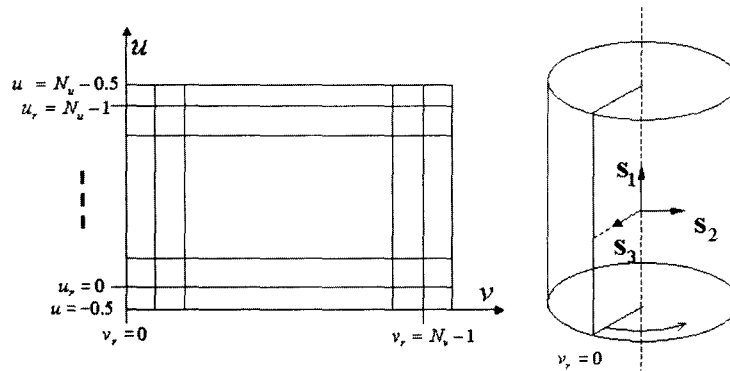


그림 3. $u-v$ 좌표계
Fig 3. $u-v$ coordinate

1.2 거리측정 기준점

거리측정 기준점은 위에서 정의한 원기둥의 옆면에 좌표계를 부여하고 좌표계 위의 정수 좌표들을 기준점으로 선택한다. 그림 3은 원기둥의 옆면의 전개도와 그 위에 부여된 좌표계를 나타낸다. s_1 방향으로 u 좌표, 둘레를 따라 v 좌표를 설정한다. 거리측정 기준점은 v 방향으로 N_v 개, u 방향으로 N_u 개가 선택되며 그림 3에서 격자의 교차점에 해당된다. 각 기준점의 $u-v$ 좌표는 (u_r, v_r) 로 표시하며 $(u_r, v_r) \in \{(0,0), \dots, (N_u-1, N_v-1)\}$ 이다. 각 기준점에서는 3차원 메쉬의 표면까지의 거리가 계산되며 거리값을 모으면 $N_v \times N_u$ 크기의 거리영상을 얻게 된다.

1.3 거리 계산

각 기준점으로부터 3차원 모델의 면까지의 거리를 계산

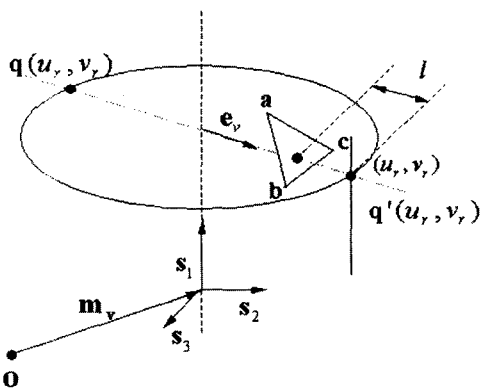


그림 4. 거리 계산
Fig 4. Range calculation

하려면 먼저 각 기준점의 3차원상의 위치를 계산하고, 계산된 위치로부터 반지름 방향으로 직선을 그어 3차원 모델의 면과의 교점까지의 거리를 구한다. 그림 4는 임의의 기준점 (u_r, v_r) 에서 거리를 계산하는 것을 보여준다. 그림 4에서 $q'(u_r, v_r)$ 를 기준점 (u_r, v_r) 의 3차원 좌표라 할 때, 직선 qq' 와 3차원 모델의 면 abc 와의 교점까지의 거리 l 을 구하게 된다.

기준점의 3차원 좌표를 쉽게 표시 하기위해 다음과 같은 벡터 두개를 정의한다.

$$e_u = \frac{h}{N_u} s_1$$

$$e_v(x) = \cos(2\pi x/N_v) s_3 + \sin(2\pi x/N_v) s_2$$
(5)

e_u 는 s_1 방향의 벡터이며 3차원 상에서 u 방향으로 인접한 기준점사이의 거리를 그 크기로 갖는다. $e_v(x)$ 는 x 에 따라 변하는 벡터로 v 좌표 값이 x 인 지점을 향하는 반지름 방향의 벡터이다. 식 (5)의 두 벡터를 이용하여 $q'(u_r, v_r)$ 를 표현하면 다음과 같다.

$$q'(u_r, v_r) = m_v + h_{min} s_1 + (u_r + 0.5) e_u + r e_v(v_r)$$
(6)

거리값 l 을 구하는 문제는 점 $q'(u_r, v_r)$ 로부터 $-e_v(v_r)$ 방향으로 광선을 쏘아 3차원 상의 면과 교차점을 구하는 문제로서 이에 대한 많은 연구 결과들이 있다^{[12][13]}. 실제의 구현에서는 빠른 속도를 위해 균일 공간 분할(Uniform Space Subdivision)^{[16][17]}을 사용하여 구현하였다.

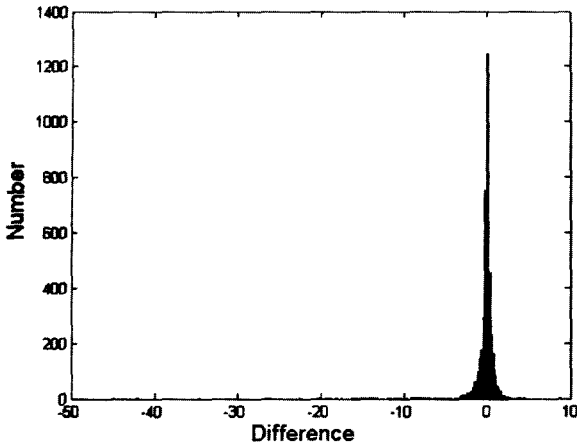


그림 5. 거리값 변화량 도수분포표
Fig 5. Histogram of the range value differences

그림 5는 원본 3차원 모델의 거리 영상과 1/8 메시 간략화를 수행한 후의 거리영상의 차이를 도수분포표로 그린 것으로서 차이 값이 0 근처에 집중되어 있는 모습을 볼 수 있다. 즉, 서론에서 언급한 바와 같이 거리영상의 화소값들이 메시 간략화와 같이 꼭지점의 좌표와 연결관계가 모두 변하는 공격을 당하더라도 공격 전후의 차이가 작음을 보여 주고 있다. 따라서 거리영상에 워터마크를 삽입한다면 공격에 대한 강인성을 얻게 된다.

2. 워터마크의 삽입

이 절에서는 위에서 제시한 방법에 따라 얻은 거리영상에 워터마크를 삽입하고, 워터마크된 거리영상을 이용하여 원본 모델의 꼭지점을 움직여서 워터마크된 3차원 모델을 얻는 방법을 설명한다.

2.1 거리영상 워터마킹

거리영상은 DCT를 기반의 간단한 워터마킹 기법을 이용한다. 일반적으로 위에서 구한 거리영상에서 거리값이 있는 부분은 사각형 모양이 아니다. 따라서 DCT를 사용하기 위해서 사각형 영역을 선택하고 이 부분에만 워터마크를 삽입한다. 선택된 사각형 영역의 좌표와 크기는 워터마크 열쇠로 추출기에 전송된다.

거리값은 3차원 모델에 따라 큰 범위로 변하기 때문에 거리영상에서 원기둥의 지름에 해당하는 거리값이 255가 되도록 선형비례조절한 후 워터마크를 삽입한다. 이렇게 하

면 거리영상이 0-255사이의 실수 값을 가지게 되고 일반적인 2차원 영상의 범위가 되므로 기존의 영상 워터마크 기법을 적용하는데 무리가 없게 된다.

DCT 영역에 워터마크를 삽입하기 위해 거리영상을 8×8 블록으로 나누고 각 블록을 DCT한 후, DC를 제외하고 그 다음 스캔으로 13개의 AC계수에 워터마크를 넣는다. 워터마크는 각 블록의 AC계수에 $c' = c \pm \alpha w$ 와 같이 삽입된다. w 는 1 과 -1이 동일한 확률로 발생하는 신호이고, α 는 워터마크 강도로서 삽입되는 정보가 비트 1 이면 +, 0 이면 -해준다. 이 워터마크는 DCT블록의 개수만큼의 비트를 삽입할 수 있다. 워터마킹을 삽입한 후에는 255가 다시 원기둥의 지름이 되도록 선형비례조절을 해준다.

2.2 3차원 모델 워터마킹

거리영상에 워터마크를 삽입하면 거리값이 변화하게 된다. 이를 3차원 모델에 반영하기 위해서는 각 거리값에 영향을 주는 꼭지점들을 워터마크로 인해 변화된 거리만큼 옮겨주어서 3차원 모델이 변화된 거리값을 나타내도록 해주어야 한다. 즉, 워터마크된 3차원 모델로부터 거리영상을 얻었을 때, 워터마크된 거리영상을 얻게 되도록 꼭지점들을 움직여 주어야 한다. 그러기 위해서 각 기준점에서 변경된 거리값을 기준점 주위의 꼭지점들에 반영하면 된다. 이를 위해서 먼저 꼭지점들을 원기둥의 옆면에 투영하고, 각 꼭지점의 $u-v$ 좌표를 구한다. 그 다음에 투영된 꼭지점과 가장 가까운 기준점에 꼭지점을 할당한다. 이렇게 각각의 기준점에 할당된 꼭지점은 해당 기준점에서 변경된 거리값을 반영하게 된다. 구체적으로 이 과정은 다음과 같이 3단계로 나누어 수행된다.

- 단계 1: 꼭지점들을 $u-v$ 좌표평면으로 투영하여 각 꼭지점의 $u-v$ 좌표를 구한다. 임의의 꼭지점 v 의 $u-v$ 좌표는 다음과 같다.

$$u(v) = \frac{\mathbf{s}_1^T (\mathbf{v} - \mathbf{m}_v) - h_{\min}}{h/N_u} - 0.5 \quad (7)$$

$$v(v) = \frac{\theta_1}{2\pi/N_u} \quad \text{or} \quad \frac{\theta_2}{2\pi/N_u}$$

θ_1, θ_2 는 그림 6에서 보인 각도이다. 한 꼭지점은 $u-v$ 평면(원기둥 옆면)에 두 가지 방향으로 투영 될 수 있으

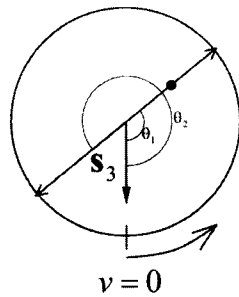


그림 6. 꼭지점 투영
Fig 6. Vertex projection onto u-v plane

므로 각 꼭지점의 수직벡터와 내적이 양수인 방향을 선택해서 투영한다.

- 단계 2: 집합 $\mathbb{B}(u_r, v_r)$ 를 정의한다. 이 집합은 $u-v$ 평면에서 임의의 기준점 (u_r, v_r) 을 중심으로 1×1 크기 영역 안에 있는 꼭지점의 집합이며, 기준점 (u_r, v_r) 에서의 거리값의 변화에 따라 변경해야하는 꼭지점의 집합이다.

$$\begin{aligned} \mathbb{B}(u_r, v_r) = \{ \mathbf{v} \mid \mathbf{v} \in \mathbb{V} \\ \text{and } |u(\mathbf{v}) - u_r| < 0.5 \\ \text{and } |v(\mathbf{v}) - v_r| < 0.5 \} \end{aligned} \quad (8)$$

- 단계 3: 위와 같이 $\mathbb{B}(u_r, v_r)$ 가 구해지면, 기준점 (u_r, v_r) 에서의 거리값 변화만큼 $\mathbb{B}(u_r, v_r)$ 에 속한 꼭지점을 움직여 준다. 교차점 (u_r, v_r) 에서 원래의 거리값을 $l(u_r, v_r)$, 워터마크된 거리값을 $\tilde{l}(u_r, v_r)$ 이라고 하자. 이때 $\mathbb{B}(u_r, v_r)$ 에 포함된 꼭지점을 원래 거리값과 워터마크된 거리값의 차이인 $\Delta l(u_r, v_r) = \tilde{l}(u_r, v_r) - l(u_r, v_r)$ 만큼 $\mathbf{e}_v(v_r)$ 방향으로 옮겨주면, 3차원 모델이 워터마크된 거리값을 갖게 된다. 즉 모든 꼭지점을 아래와 같이 변경하면 된다.

$$\begin{aligned} \forall (u_r, v_r), \forall \mathbf{v} \in \mathbb{B}(u_r, v_r) \\ \mathbf{v}' = \mathbf{v} - \Delta l(u_r, v_r) \mathbf{e}_v(v_r) \end{aligned} \quad (9)$$

그런데 식 (9)과 같이 꼭지점을 움직여 주면 그림 7에서 보듯이 서로 다른 집합 $\mathbb{B}(u_r, v_r)$ 에 속해있는 꼭지점들의 변화량이 달라서 그 경계에서 불연속성이 나타나게 되고, 눈에 거슬리게 된다. 따라서 꼭지점을 변화시킬 때, 가중치

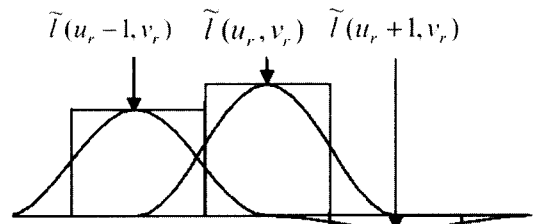


그림 7. 가중치의 유무에 따른 꼭지점 변경
Fig 7. Vertex perturbation with and without weights

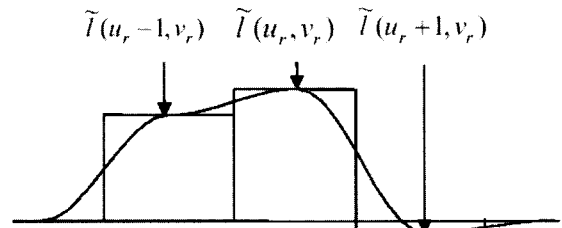


그림 8. 가중치로 부드럽게 된 경계
Fig 8. Smoothed boundary with weighted perturbation

를 주어 꼭지점이 자신이 속한 기준점으로부터 멀어질수록 작게 움직여주고, 주변의 변화량도 고려해주어야 부드럽게 연결된다. 현재 기준점의 변화량을 주위의 8개의 변화량에 영향을 주도록 하기위해서 그림 7에서처럼 인접한 거리값 사이에 서로 영향이 겹쳐지도록 가중치를 준다. 최종적으로 현재 기준점 주위 8개가 미치는 영향을 모두 더하여 그림 8과 같이 부드럽게 연결된 곡선에 맞추어서 꼭지점이 움직이게 된다. 여기서, 가중치는 아래와 같이 정의한다.

$$\begin{aligned} w(u_r, v_r, u, v) = \\ \begin{cases} \{1 + \cos(\pi(u - u_r))\} \{1 + \cos(\pi(v - v_r))\} / 4 & \text{if } |u - u_r| < 1 \text{ and } |v - v_r| < 1 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (10)$$

위 식은 기준점 (u_r, v_r) 에서 1이 되고 이와 인접한 기준점들 $(u_r \pm 1, v_r \pm 1)$ 에서는 0을 가지며 그 사이는 코사인함수로 부드럽게 연결되는 곡면이다. 기준점 (u_r, v_r) 에 속한 꼭지점들, 즉 $\mathbb{B}(u_r, v_r)$ 의 원소들의 변화량은 자기

자신의 가중변화량과 인접한 영역의 영향을 모두 합한 만큼 움직이게 된다. 기준점(u_r, v_r)에 속한 임의의 꼭지점의 $u-v$ 좌표를 (u, v)라고 할 때, 그 변화량은 다음과 같다.

$$\Phi(u_r, v_r, u, v) = \sum_{j=1}^1 \sum_{k=1}^1 w(u_r + j, v_r + k, u, v) \Delta l(u_r + j, v_r + k) \quad (11)$$

이제 모든 $\mathbb{B}(u_r, v_r)$ 에 속한 꼭지점들에 대해, 위의 변화량 만큼 반지름 방향으로 꼭지점을 움직여 주어서 다음과 같이 워터마크된 꼭지점 \mathbf{v}' 를 얻는다.

$$\forall (u_r, v_r), \forall \mathbf{v} \in \mathbb{B}(u_r, v_r) \quad \mathbf{v}' = \mathbf{v} - \Phi(u_r, v_r, u(\mathbf{v}), v(\mathbf{v})) \mathbf{e}_v(v(\mathbf{v})) \quad (12)$$

2.3 워터마크 추출

앞의 두 절에서는 거리영상에 워터마킹을 하고 이를 이용하여 3차원 모델에 워터마크를 삽입하는 과정을 설명하였다. 이 절에서는 워터마크의 추출에 대해 설명한다. 워터마크 추출에는 원본 모델이 필요하다. 3차원 모델이 심하게 공격당하지 않아서 원기둥의 기준으로 사용되는 공분산 행렬의 고유 벡터가 거의 변화하지 않는다면 정합이 없이도 추출이 가능하나 그렇지 않은 경우 원본 모델과의 정합이 필요하다. 정합을 수행한 후에 원본 모델과 워터마크된 모델에서 각각 거리영상을 추출한 후에 8x8블록으로 나누어 DCT를 한다. 각각의 블록으로부터 추출된 워터마크는

$$w^* = \frac{c' - c}{\alpha} \quad (13)$$

이고, 이것과 원본 워터마크와의 유사도 ρ 를 각 블록마다 다음과 같이 비교한다.

$$\rho = \frac{\sum_i (w_i^* - \bar{w}_i^*)(w_i - \bar{w}_i)}{\sqrt{\sum_i (w_i^* - \bar{w}_i^*)^2} \sqrt{\sum_i (w_i - \bar{w}_i)^2}} \quad (14)$$

식 (14)에서 \bar{w} 와 \bar{w}^* 는 평균값을 의미한다. ρ 가 0보다 크다면 이 블록의 정보는 1로 복원하고, 0보다 작다면 0으로 복원한다.

III. 실험 결과

1. 강인성 실험

실험에 사용된 모델은 모두 5개로 그림 9와 같다. "행복한 부처" 는 Stanford 3D Scanning Repository 웹 사이트가 출처이고, 터빈날개는 Georgia Tech's Large Geometric Models Archive 웹 사이트에서 가져왔다. 다비드상과 두상은 Digital Michelangelo Project가 출처이다. 원본은 크기가 더 크나 실험을 위해 약 15만개 꼭지점을 가지도록 조정되었다.



행복한 부처

다비드

터빈날개

다비드 두상

비너스

그림 9. 실험용 3차원 메쉬모델

Fig 9. 3D mesh models for the experiments

표 1. 실험에 사용된 변수

Table 1. Parameters for experiments.

모델	행복한부처	터빈날개	다비드상	다비드두상	비너스
꼭지점수	148488	144650	162774	151855	50002
$N_u \times N_v$	80x66	64x57	127x69	69x102	35x68
모서리 (u, v)	(20,1)	(8,0)	(1,0)	(8,0)	(1,0)
사각영역 크기	56x56	56x56	64x64	40x96	32x64
비트수	49	49	64	60	32
강도 α	0.3	0.1	0.3	0.3	0.2

표 2. Benedens 알고리즘에 사용된 변수

Table 3. Parameters for Benedens' algorithm.

모델	행복한부처	터빈날개	다비드상	다비드두상	비너스
Bin 수	98	98	128	120	64
비트수	49	49	64	60	32
강도 α	0.3	0.1	0.3	0.3	0.2

표 3. 원본과 워터마크된 모델 사이의 하우스도르프 거리

Table 3. Hausdorff distances between original and watermarked models

모델	제안하는 기법	Benedens의 기법
행복한부처	0.118781	0.118967
터빈날개	8.754340	8.753471
다비드상	20.848623	20.848623
다비드두상	5.407834	5.407834
비너스	0.015362	0.028937

실험에 사용된 변수는 표 1 과 같다. 모서리의 좌표와 사각영역의 크기는 워터마크를 위해 선택된 영역의 시작 위치와 그 크기이다. 표 2는 Benedens 알고리즘에서 사용된 파라미터이다. 비교를 위해 같은 양의 비트를 삽입하였다. 삽입된 비트 수는 32-64비트이며 많은 정보를 전달할 수 있는 수준은 아니다. 그러나 제안하는 기법은 정보 전달의 목적이 아닌 공격에 워터마크가 살아남는 것이 목적이며 강인성을 위해 정보의 양이 제한된다. 강인성과 정보량의 관계는 정량적으로 분석하기가 어려우나, 정성적으로 반비례의 관계가 있다. 워터마크는 비가시성을 위해서 삽입되는 신호의 에너지가 제한되어있고, 제한된 에너지를 각 비트에 분배하게 되는데 비트량이 늘어날수록 각 비트에 할당되는 에너지가 줄어들어 워터마크의 강인성이 떨어지게 된다.

표 3은 원본과 워터마크가 삽입된 모델 사이의 하우스도르프(Hausdorff) 거리이며 제안하는 알고리즘과 Benedens

표 4. 제안된 알고리즘의 비트 에러율

Table 4. Bit error rate of the proposed algorithm

모델	간략화 1/2	간략화 1/8	가우스잡음0.03%	가우스잡음 0.1%	양자화 11비트
행복한부처	0.26%	18.6%	2.16%	5.40%	0.19%
터빈날개	6.28%	33.0%	1.18%	6.38%	3.38%
다비드상	7.80%	11.1%	1.09%	4.13%	1.90%
다비드두상	1.97%	11.3%	0.29%	1.30%	0.26%
비너스	0.03%	5.20%	0.16%	0.31%	0.66%
평균	3.27%	15.8%	0.98%	3.50%	1.28%

표 5. Benedens 알고리즘의 비트 에러율

Table 5. Bit error rate of the Benedens' algorithm

모델	간략화 1/2	간략화 1/8	가우스잡음0.03%	가우스잡음 0.1%	양자화 11비트
행복한부처	2.06%	41.6%	1.04%	45.7%	1.22%
터빈날개	22.9%	48.2%	4.27%	44.5%	5.56%
다비드상	4.45%	35.9%	9.11%	47.7%	13.2%
다비드두상	5.70%	38.8%	0%	35.3%	1.42%
비너스	2.94%	37.5%	0%	11.7%	0%
평균	7.61%	40.2%	2.89%	37.0%	4.28%

알고리즘에서 워터마크의 강도가 유사하도록 조절되었음을 보여준다. 모델마다 꼭지점 좌표의 크기가 다르므로 서로 다른 모델사이에서 표 3에 주어진 거리값의 비교는 의미가 없으며, 같은 모델에서는 유사함을 확인할 수 있다. 하우스도르프 거리는 P. Cignoni 등이 제안한 Metro^[15]를 사용하였다.

워터마크의 강인성을 테스트하기 위한 공격으로는 매쉬 간략화 공격과 가우시안 잡음 공격, 그리고 꼭지점 좌표의 양자화 공격을 사용하였다. 간략화는 꼭지점 좌표의 수가 원본의 1/2배, 1/8배 되도록 하였다. 꼭지점에 더해지는 가우시안 잡음은 3차원의 벡터로 각 좌표값이 서로 독립이고 분산이 1인 가우시안 잡음이다. 이렇게 구한 잡음 벡터의 크기를 조절하여 잡음의 강도를 다르게 한다. 잡음의 크기는 3차원 모델의 경계 상자(bounding box)의 최장 대각선 길이에 대한 비율이다. 꼭지점 양자화는 꼭지점 좌표 세 성분을 각각 11bit로 양자화 한 것이며, 3차원 압축 알고리즘에서 전처리로 쓰이는 기법이다. 3차원 모델의 경계 상자를 $2^{11} \times 2^{11} \times 2^{11}$ 의 격자로 나누고 꼭지점의 좌표를 가장 가까운 교차점의 좌표로 바꾸어 주었다.

표 4 는 각 공격에 대한 비트 에러율을 나타낸다. 각각의 에러율은 200번 워터마킹을 수행한 평균값이다. 표 5 는 비

교를 위한, 수직 벡터의 강인한 특징을 이용한 Benedens의 알고리즘^[3]의 결과이다. 표 4의 실험에서는 모델을 인위적으로 회전시킨 후에 원본과의 정합을 하였으며, 정합 알고리즘은 Chen의 알고리즘^[14]을 사용하였다. Benedens 알고리즘도 정합이 필요하지만, 원본의 자세를 그대로 유지하여서 이미 정합 되어있다고 가정하고 워터마크를 추출하였다. 실험 결과를 보면 대부분의 경우 제시한 방법이 Benedens 알고리즘보다 더 좋은 성능을 보여주었으며, 상대적으로 강력한 공격인 1/8간략화에 대해서도 Benedens 알고리즘보다 절반 이하의 에러율을 나타내었다.

2. 복잡도 실험

제안하는 알고리즘은 대부분의 수행시간이 거리 영상을 구하는데 걸린다. 거리값을 구하는 문제는 광선과 면의 교차점을 구하는 문제 이므로 실제 구현에서는 광선 추적(Ray-Tracing)을 빠르게 하는 기법중 하나인 공간 균일 분할(Uniform space subdivision)^{[16][17]}을 사용하였고, 약 10만개의 복셀을 사용하였다.

Benedens의 알고리즘의 경우, 워터마크 삽입에서는 비용 함수를 최소화하는데 사용되는 다차원 하강 단체법(Multi-dimensional downhill simplex)에서 대부분의 시간을 소모한다. 워터마크의 추출의 경우에는 EGI(Extended Gaussian Image)를 구성하는데 걸리는 시간이 대부분이 된다.

표 6은 두 알고리즘의 수행 시간을 나타낸다. 수행시간은 알고리즘의 계산시간만이 아닌 파일 입출력을 포함한 프로그램이 수행된 전체 시간이다. 워터마크를 삽입하는 경우 최적화 기법을 사용하는 Benedens의 알고리즘이 시간이 많이 걸려서 수분에서 수십 분의 시간이 소요된다. 제안하는 기법은 약 12-15초가 소요된다. 추출과정에서는 15만개의 꼭지점에 대해 제안하는 기법은 15-19초,

Benedens의 경우 18-24초가 걸린다. 5만개에 대해서 약 4-6초가 소요된다.

IV. 결론

본 논문에서는 거리 영상을 이용한 3차원 강인한 워터마킹을 제안하였다. 거리 영상은 기준점으로부터 3차원 모델까지의 거리를 화소값으로 하는 이차원 영상으로서 3차원 모델에 대한 공격에 대해 변화가 적은 특성을 이용하였다. 거리영상에 워터마크를 삽입한 후, 워터마크된 거리영상을 이용하여 3차원 모델의 꼭지점 좌표를 변경하여 워터마크된 모델을 얻게 된다. 거리영상은 이차원 영상이므로 DCT 등 기존의 영상 워터마킹 방법을 사용할 수 있다. 워터마크의 추출은 원본과의 정합 후에 거리영상을 구하고, 이로부터 워터마크를 추출하게 된다. 제안하는 알고리즘은 가우스 잡음, 양자화 같이 꼭지점 좌표에 대한 공격과 꼭지점 좌표와 연결정보를 모두 손상 시키는 메쉬 간략화에 강인함을 보인다.

참고 문헌

[1] R. Ohbuchi, H. Masuda and M. Aono, "Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications," IEEE Journal on Selected Areas in Communications, Vol. 16, Issue 4, pp. 551 -560, May 1998.

[2] R. Ohbuchi, H. Masuda and M. Aono, "A Shape-preserving Data Embedding Algorithm for NURBS Curves and Surfaces," IEEE Computer Graphics International Proceedings, pp. 180 -187, 1999.

[3] O. Benedens, "Geometry-based Watermarking of 3D Models," IEEE Computer Graphics and Applications, Vol. 19, pp. 46-55, Jan.-Feb 1999.

[4] O. Benedens, "Affine Invariant Watermarks for 3D polygonal and NURBS Based Models," Springer, Information Security, Third International Workshop, ISW2000, vol. 1975, pp. 15-29, Australia, December 2000.

[5] O. Benedens, "Two High Capacity Methods for Embedding Public Watermarks into 3D Polygonal Models," Proceedings of the Multimedia and Security-Workshop at ACM Multimedia 99, pp. 95 - 99, Orlando, Florida, 1999.

[6] M. G. Wagner, "Robust Watermarking of Polygonal Meshes," Geometric Modeling and Processing 2000, pp. 201 -208, 2000.

[7] E. Praun, H. Hoppe and A. Finkelstein, "Robust Mesh Watermarking," ACM the 26th annual conference on Computer graphics and interactive techniques, pp. 49 - 56, New York, July

표 6. 수행시간 (Pentium 4 1.7Ghz, 512Mbyte RAM)
Table 2. Processing time

모델	제안하는 기법		Benedens의 기법	
	삽입	추출	삽입	추출
행복한부처	12.35	16.29	12:59.3	17.9
터빈날개	11.84	18.36	7:13.4	17.9
다비드상	14.17	15.03	33:27.3	24.3
다비드두상	13.50	17.71	2:38.6	21.9
비너스	4.16	5.57	2:38.6	4.3

*콜론 앞의 숫자는 분, 나머지 단위는 초

1999

[8] B.-L. Yeo and M. M. Yeung, "Watermarking 3D objects for Verification," *IEEE Computer Graphics and Applications*, Vol. 19, Issue 1, pp. 36-45, January/February, 1999.

[9] S. Kanae, H. Date, and T. Kishinami, "Digital Watermarking for 3D Polygons using Multiresolution Wavelet Decomposition," *Proc. Sixth IFIP WG 5.2 GEO-6*, pp. 296-307, Tokyo Japan, Dec., 1998.

[10] R. Ohbuchi, S. Takahashi, T. Miyazawa and A. Mukaiyama, "Watermarking 3D Polygonal Meshes in the Mesh Spectral Domain," *Proc. Graphics Interface 2001*, pp. 9-17, 2001.

[11] E. Paquet and M. Rioux, "Nefertiti: A Query by Content Software for Three-Dimensional Models Databases Management," *3-D Digital Imaging and Modeling, 1997. Proceedings., International Conference on Recent Advances in 1997*, pp. 345-352, 1997.

[12] J. Orourke, "Computational geometry in C : 2nd ed.," Cambridge, UK, : New York, NY, USA : Cambridge University Press, 1998.

[13] R.J. Segura and F. R. Feito, "An Algorithm for Determining Intersection Segment-Polygon in 3D," *Computers and Graphics* published by Elsevier Science, Vol.22, Issue 5, pp. 587-592, October 1998.

[14] Y. Chen, and G. Medioni, "Object modeling by registration of multiple range images," *Image and Vision Computing* 10, pp. 145-155, April 1992.

[15] P. Cignoni, C. Rocchini and R. Scopigno, "Metro : measuring error on simplified surfaces," *Computer Graphics Forum*, Blackwell Publishers, vol. 17(2), pp. 167-174, June 1998, "http://vcg.isti.cnr.it/activities/surfacegevis/simplification/metro.html"

[16] A. Fujimoto and K. Iwata, "Accelerated Ray Tracing," *Proc. CG Tokyo '85*, pp. 41-65, 1985.

[17] A. Fujimoto, T. Tanaka, and K. Iwata, "ARTS: Accelerated Ray-Tracing System," *IEEE Computer Graphics and Applications*, pp. 16-26, April 1986.

저 자 소 개



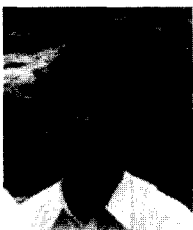
송 한 새

- 2001년 : 서울대학교 전기공학부 학사
- 2003년 : 서울대학교 전기컴퓨터공학부 석사
- 2003년~현재 : 서울대학교 전기컴퓨터공학부 박사과정



조 남 익

- 1986년 : 서울대학교 제어계측공학과 학사
- 1988년 : 서울대학교 제어계측공학과 석사
- 1992년 : 서울대학교 제어계측공학과 박사
- 1991년~1994년 : 제어계측신기술연구소 연구원
- 1994년~1998년 : 서울시립대학교 전자전기공학부 조교수
- 1999년~2001년 : 서울대학교 전기공학부 조교수
- 2001년~현재 : 서울대학교 전기공학부 부교수



김 종 원

- 1995년 : 서울시립대학교 대학원 전자공학과 공학박사
- 1995년~1996년 : KISTI 선임연구원
- 1996년~2000년 : 주성대학 정보통신학과 조교수
- 2000년~2004년 : (주)마크애니 연구소장
- 2004년~현재 : 상명대학교 소프트웨어대학 소프트웨어학부 초빙교수
- 주관심분야 : 디지털 워터마킹, 디지털 저작권 관리(DRM), 디지털 신호처리