

CEGI를 이용한 3D 메쉬 워터마킹

이 석 환,* 김 태 수,* 김 승 진,* 권 기 룡,** 이 건 일*

3D Mesh Watermarking Using CEGI

Suk-Hwan Lee,* Tae-Su Kim,* Seung-Jin Kim,* Ki-Ryong Kwon,**
Kuhn-II Lee*, *Regular members*

요 약

본 논문에서는 CEGI (Complex Extended Gaussian Image)를 이용한 3D 메쉬 모델 워터마킹 알고리즘을 제안하였다. 제안한 알고리즘에서는 VRML 데이터의 3D 메쉬 모델을 6개 패치로 분할한 후, 각 패치의 CEGI 분포에서 복소 가중치의 크기가 큰 셀에 투영되는 메쉬의 법선 벡터 방향에 워터마크를 삽입한다. 그리고 각 패치의 중점 좌표 및 CEGI 크기 분포의 우선 순위 정보를 이용하여 워터마크를 추출한다. 또한 아핀 (affine) 변형된 모델에서는 패치의 초기 중점 좌표의 재배열 과정을 이용하여 원 모델의 방향으로 전환한 후, 워터마크를 추출한다. 본 논문에서 제안한 알고리즘의 성능을 평가하기 위한 실험에서 기하학적 및 위상학적 변형에 강인한 특성을 가짐을 확인하였다.

Key Words : Mesh watermarking, CEGI,

ABSTRACT

We proposed 3D mesh watermarking algorithm using CEGI distribution. In the proposed algorithm, we divide a 3D mesh of VRML data into 6 patches using distance measure and embed the same watermark bits into the normal vector direction of meshes that mapped into the cells of each patch that have the large magnitude of complex weight of CEGI. The watermark can be extracted based on the known center point of each patch and order information of cell. In an attacked model by affine transformation, we accomplish the realignment process before the extraction of the watermark. Experiment results exhibited the proposed algorithm is robust by extracting watermark bit for geometrical and topological deformed models.

I. 서 론

최근 웹 상에서는 평면 이미지뿐만 아니라 3D 그래픽을 구현할 수 있는 Web 3D 기술에 대한 관심이 증가되고 있다. 이들 중 가상 현실 구현 언어 (virtual reality modeling language, VRML)는 여러 가지 장점으로 인하여 특히 주목받고 있다. 첫째, 국제 표준 기구인 ISO/IEC가 웹 상에서 3D 그래픽을 표현하는 표준으로 VRML를 공인하고 있다. 둘째, VRML은 소스자체가 공개되어 있고 라이선스를 요구하지 않아 원천 기술 개발 뿐만 아니라 누구나

자유로이 웹 상에서 3D 그래픽을 표현 할 수 있다. 따라서, 3D 그래픽 모델에 대한 저작권을 보호하기 위하여 이들의 워터마킹 연구가 진행되고 있다.

일반적으로 3D 그래픽 모델은 연결성 및 기하학적 데이터에 의하여 정의되는 메쉬로 표현된다. 그러나 이들 메쉬 데이터들은 정지 영상, 오디오, 및 비디오 등의 데이터와는 다른 특징을 가지고 있다. 먼저 다양한 기하학적 (geometrical) 및 위상학적 (topological) 연산에 의하여 편집이 용이하다. 그리고 3D 그래픽 모델을 표현하는 방법이 유일하지 못하다. 즉 하나의 모델에 대하여 꼭지점, 에지, 및 표

*경북대학교 대학원 전자공학과 e-mail : skylee@palgong.knu.ac.kr

** 부산외국어대학교

논문번호 : 030327-0801, 접수일자 : 2003년 8월 1일

면들이 다르게 구성되어 있는 여러 메쉬들로 표현이 가능하다. 또한 모델 데이터의 정렬 순서가 절대적이지 않다. 그러나 공간상에서 모델의 위치 및 방위각에 대한 정보가 필요하다. 마지막으로 렌더링(rendering) 속도를 높이기 위하여 모델 데이터 양을 메쉬 간단화(mesh simplification)에 의하여 줄일 수 있다. 메쉬 간단화된 모델에서는 기존의 꼭지점 및 메쉬가 없어지고, 새로운 꼭지점 및 메쉬가 생성된다. 3D 메쉬 모델에 대한 워터마킹 기술은 위의 사항들이 고려되어야 한다.

3D 메쉬 모델의 대표적인 워터마킹 알고리즘으로 Ohbuchi 등[1],[2]은 3D 메쉬 모델의 메쉬 분광 영역(mesh spectral domain)에서 워터마크를 삽입하였다. 그러나 이 알고리즘은 메쉬 간단화(mesh simplification) 및 리메쉬(remesh)와 같이 메쉬 연결성을 바꾸는 공격에 대하여 견고하지 못하는 단점을 갖는다. Kanai 등[5]이 제안한 알고리즘에서는 3D 메쉬 모델의 다해상도 웨이블릿 분해를 이용하여 3D 웨이블릿 계수 벡터에 워터마크를 삽입한다. 이 알고리즘에서 적용된 웨이블릿 분해는 4개의 메쉬를 1개의 메쉬로 분할하는 정규적인(regular) 형태의 부분할 기법이므로, 비정규적인(irregular) 3D 메쉬 모델의 위상(topology)을 4-to-1의 연결성으로 바꾸어야 하는 단점이 있다. Praun 등[6]은 점차적 메쉬(progressive mesh)의 다해상도 해석을 이용하여 메쉬의 꼭지점 상에서 일련의 스칼라 기저 함수(scalar basis function) 집합을 구한 후, 기저 함수에 가중치를 둔 메쉬의 법선 벡터 방향에 따라 워터마크를 삽입하는 알고리즘을 제안하였다. 그러나, 이 알고리즘에서는 워터마크 검출시 원 모델의 동일한 기하학적 및 위상학적 특성을 가지는 재표본화(resampling) 과정이 필요하다. Benedens[3],[4]이 제안한 알고리즘에서는 3D 형상을 인식할 수 있는 요소이며 또한 위상학적 변형에 변하지 않는 메쉬 법선 벡터 분포를 이용하였다. 이 알고리즘에서는 미리 정의된 하나의 원점에 대한 메쉬들의 표면 법선 벡터 분포를 이용한다. 즉, 하나의 EGI(Extended Gaussian Image)[14],[15] 분포에서 랜덤하게 선택된 방향에 속해 있는 메쉬들의 법선 벡터 분포에 워터마크를 삽입한다. 그러나 절단과 같은 기하학적 공격이 가해질 경우, 그 방향에 속해 있는 메쉬들이 없어지므로, 삽입된 워터마크 정보가 없어진다. 그러므로 메쉬 간단화, 리메쉬, 및 절단과 같은 공격에 강하면서, 워터마크 검출시 재표본화 과정 및 원 모델이 필요없는 알고리즘이 필요하다.

본 논문에서는 꼭지점 좌표 및 연결성 정보가 바뀌더라도 메쉬 법선 벡터들의 방향에 따른 분포는 바뀌지 않으며, EGI와는 달리 물체의 방향(orientation) 및 평행 이동(translation)을 결정할 수 있으며, 같은 EGI 분포를 가지는 convex 및 nonconvex의 물체를 구별할 수 있는 CEGI(Complex Extended Gaussian Image)[16] 크기 분포를 이용하여 워터마크를 삽입 및 추출하는 알고리즘을 제안하였다. 제안한 알고리즘에서는 3D 메쉬 모델을 6개의 패치로 분할한 후, 각 패치에 대한 CEGI 크기 분포에 따라 동일한 워터마크 정보를 삽입하므로써 절단과 같은 기하학적 변형 및 리메쉬 및 표면 간단화 등의 위상학적 변형에 강한 특성을 가진다. 그리고 워터마크 삽입 과정에서 구한 패치의 중점 좌표값 \mathbf{I} 및 CEGI 크기 분포의 순위 정보를 이용하여 워터마크를 검출한다. 그리고 아핀(Affine) 변형된 모델에서는 패치의 중점 좌표값 \mathbf{I} 에 대한 재배열(realignment) 과정을 이용하여 원 모델의 방향으로 전환한 후, 워터마크를 검출한다.

본 논문에서 제안한 알고리즘의 성능을 평가하기 위한 실험 결과에서, 제안한 알고리즘이 메쉬 간단화 및 리메쉬 등의 위상학적 공격과 절단 등의 기하학적 공격에 대하여 우수한 견고성을 가짐을 확인할 수 있었다. 또한 워터마크 삽입시 단계적 탐색 방법을 이용하여 최적의 꼭지점을 구하므로 우수한 비가시성을 확인할 수 있었다.

II. CEGI

3D 모델의 EGI[14],[15]에서 모델 표면의 법선 방향에 관련된 각 가중치는 이들 방향에서 보여지는 모든 표면의 면적을 나타내는 스칼라 가중치이다. 그러나 CEGI[16]에서는 그림 1에서와 같이 원점에 대한 이들 표면과의 법선 거리가 반침 함수

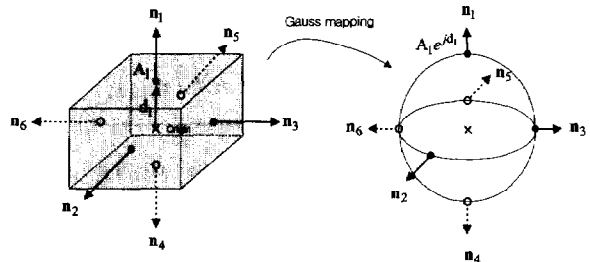


그림 1. 육면체 모델을 가우스 변환한 CEGI.

(support function)의 위상 성분으로 나타낸 복소 가중치를 가지며, 이는 기존의 EGI 표현이 확장된 개념이다. 즉, CEGI에서 특정한 법선 방향 \mathbf{n}_k 에 대한 가중치는 이 방향에서 보여지는 면적 A_{n_k} 를 크기 성분으로 가지며, 원점에서 \mathbf{n}_k 방향에 대한 표면과의 거리 d_k 를 위상 성분으로 가지는 복소 가중치 $A_{n_k} e^{jd_k}$ 이다.

본 논문에서는 3차원 공간상에서의 법선 방향을 240개의 방향으로 표본화하여, 하나의 방향으로 투영되는 모든 메쉬의 가중치를 하나의 셀에 포함한다. 그러므로 240개의 법선 방향 중 \mathbf{n}_k ($0 \leq k < 240$) 방향에 대한 셀의 가중치 $C_{\mathbf{n}_k}$ 는

$$C_{\mathbf{n}_k} = \sum_{l=1}^{N_k} (A_{l,\mathbf{n}_k} e^{jd_{l,k}}) = \sum_{l=1}^{N_k} (A_{l,\mathbf{n}_k} \cos(d_{l,k})) + j \sum_{l=1}^{N_k} (A_{l,\mathbf{n}_k} \sin(d_{l,k})) \quad (1)$$

이며 크기 성분 $||C_{\mathbf{n}_k}||$ 는

$$||C_{\mathbf{n}_k}||^2 = \left(\sum_{l=1}^{N_k} (A_{l,\mathbf{n}_k} \cos(d_{l,k})) \right)^2 + \left(\sum_{l=1}^{N_k} (A_{l,\mathbf{n}_k} \sin(d_{l,k})) \right)^2 \quad (2)$$

이고 위상 성분 $\angle C_{\mathbf{n}_k}$ 는

$$\angle C_{\mathbf{n}_k} = \tan^{-1} \left(\frac{\sum_{l=1}^{N_k} (A_{l,\mathbf{n}_k} \sin(d_{l,k}))}{\sum_{l=1}^{N_k} (A_{l,\mathbf{n}_k} \cos(d_{l,k}))} \right) \quad (3)$$

이다. 제안한 알고리즘에서는 CEGI의 240개 셀 중에서 가중치의 크기 성분이 큰 셀에 투영되는 메쉬

의 법선 벡터 방향에 워터마크를 삽입한다.

III. 제안한 메쉬 워터마킹 알고리즘

본 논문에서 제안한 3D 메쉬 모델의 워터마크 삽입 및 추출 알고리즘의 전체 블록도는 그림 2에서와 같다.

1. 3D 다면체 메쉬 모델의 패치 분할

3D 메쉬 모델의 모든 꼭지점들은 6개 패치의 중점 \mathbf{I} 들과의 거리를 구한 후, 거리가 최소가 되는 패치로 분류된다. 먼저 6개의 단위 벡터 방향 \hat{x} , \hat{y} , \hat{z} 에 인접한 꼭지점 중 원점에서 거리가 가장 먼 위치에 있는 6개의 꼭지점을 $\mathbf{I} = \{I_1, I_2, \dots, I_6\}$ 의 초기 위치로 설정한다. 그리고 각 꼭지점 v 는 6개 패치의 초기 위치 $I_{i \in [1,6]}$ 와의 거리가 최소되는 패치 $P_{i \in [1,6]}$ 로 분류된다.

각 패치 $P_{i \in [1,6]}$ 에 속해 있는 꼭지점들의 무게중심을 구한 후, 이를 I_i 에 대입한다. 그리고 I_i 에 대하여 각 꼭지점을 위와 같은 방법으로 다시 분류한다. 이러한 과정을 거리 감소 비율 $(D_m - D_{m+1})/D_m$ 이 문턱값보다 작을 때까지 반복 수행한다. 여기서 D_m 은

$\sum_{i=1}^6 \sum_{v \in P_i} ||I_i - v||^2$ 이며, m 은 반복 횟수를 나타낸다. 이와 같은 방법으로 6개 패치로 분할된 Stanford Bunny 모델은 그림 3에서와 같다.

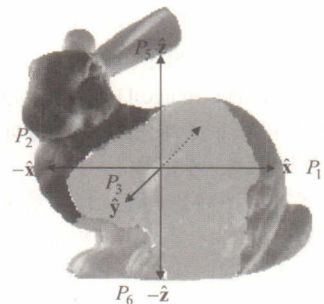


그림 3. 6개 패치로 분할된 Stanford Bunny 모델.

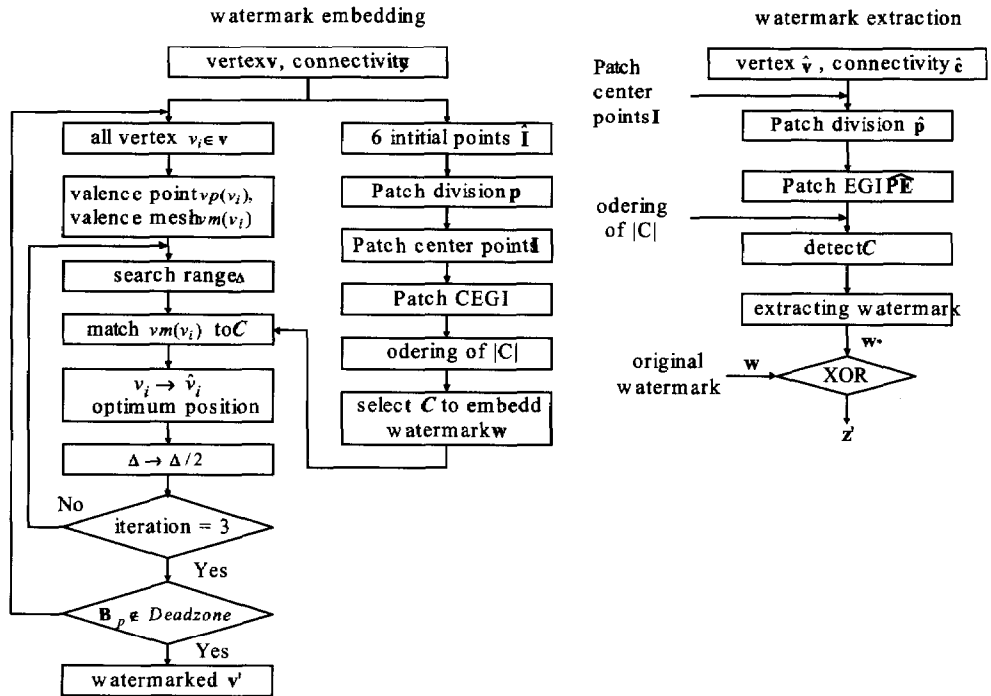


그림 2. 제안한 알고리즘의 워터마크 삽입 및 추출 블록도.

2. 각 패치별 CEGI 크기 분포

CEGI에서 k 번째 메쉬가 가지는 가중치는 복소 가중치 $A_{n_k} e^{jd_k}$ 이다. 여기서 크기 A_{n_k} 는 메쉬의 면적이고, 위상 d_k 는 기준점에서 메쉬의 중점까지의 정규화된 거리이다. 그리고 n_k 는 메쉬의 바깥쪽 방향으로 나가는 법선 벡터를 나타낸다. 본 논문에서는 각 패치의 CEGI 크기 분포를 구하기 위하여 단위 길이가 1인 구를 동일한 면적의 240개 셀로 표분화하였다. 그러므로 각 메쉬의 법선 벡터들은 240개의 방향 중 가장 가까운 방향을 가지는 셀로 투영된다.

먼저 각 패치별 메쉬의 법선 벡터를 구한다. 메쉬 법선 벡터는 면에 대하여 안쪽 방향 및 바깥쪽 방향의 두 가지 방향을 가진다. 시각적으로 보여지는 방향을 판단하기 위하여 원점에서의 벡터가 아닌 각 패치의 중점 I 에 대한 꼭지점들의 벡터에 대한 메쉬의 법선 벡터 및 면적을 구한다. 즉, i 번째 패치 P_i 에 포함되는 j 번째 메쉬의 법선 벡터 \vec{n}_{ij} 및 면적 A_{ij} 및 위상 d_{ij} 은

$$\vec{n}_{ij} = \frac{1}{||\vec{n}_{ij}||^2} (\vec{I}_i v_1 - \vec{I}_i v_2) \times (\vec{I}_i v_1 - \vec{I}_i v_3)$$

$$A_{ij} = ||(\vec{I}_i v_1 - \vec{I}_i v_2) \times (\vec{I}_i v_1 - \vec{I}_i v_3)|| / 2$$

$$d_{ij} = ||\vec{I}_i m_j||$$

(4)

이다. $\vec{I}_i m_j$ 은 패치의 중점 I_i 와 메쉬의 중점 m_j 을 잇는 벡터이다. 여기서 메쉬 연결성 공격에 강인하기 위하여 건설한 법선 벡터를 결정하여야 한다. 메쉬의 법선 벡터는 그림 4에서와 같이 바깥 방향 및 안쪽 방향에 대하여 \vec{n}_{ij} 및 $-\vec{n}_{ij}$ 이다. 두 벡터 중, $\vec{I}_i m_j$ 와의 내적이 큰 값을 이 메쉬의 법선 벡터 \vec{n}_{ij}^* 로 결정한다. 그러므로 이 메쉬의 CEGI 복소 가중치는 $A_{\vec{n}_{ij}^*} e^{j||\vec{I}_i m_j||}$ 이다. 이 메쉬는 P_i 에 속하므로 P_i 의 CEGI 240개 셀 $C_{ik: i \in [1,6], k \in [1,240]}$ 중에서, 이 메쉬의 법선 벡터 방향

과 가장 가까운 방향에 있는 셀로 이를 투영시킨다. 즉, \vec{n}_{ij}^* 과 240개 셀들의 단위 법선 벡터과의 내적이 최소가 되는 셀로

$$\vec{n}_{ij}^* \rightarrow C_{ik} : ||\vec{BC}_{ik} \cdot \vec{n}_{ij}^*||^2 < ||\vec{BC}_{ij} \cdot \vec{n}_{ij}^*||^2, \text{ all } k \neq j, 1 \leq j \leq 240 \quad (5)$$

와 같이 투영시킨 후, 이 벡터의 복소 가중치 $A_{\vec{n}_{ij}^*} e^{j\angle |\vec{I}_m|}$ 를 투영된 C_{ik} 에 더한다. 여기서, \vec{BC}_{ij} 는 P_i 의 j 번째 셀의 중점 벡터이며, 이는 원점에서 셀의 중점에 대한 단위 법선 벡터이다.

모든 메쉬들의 법선 벡터 및 이들이 투영되는 셀을 위와 같이 구한 후, 식 (2)를 이용하여 각 패치들의 CEGI 240개 셀의 크기를 각각 구한다. 즉, i 번째 패치의 CEGI 크기 분포 중, k 번째 셀의 복소 가중치 크기 $||C_{ik}||, i \in [1,6], k \in [1,240]$ 는

$$\begin{aligned} ||C_{ik}||^2 &= ||\sum_{j=1}^{N_j} (A_{\vec{n}_{ij}^*} e^{j\angle d_{ij}})||^2 \\ &= (\sum_{j=1}^{N_j} (A_{\vec{n}_{ij}^*} \cos(d_{ij})))^2 \\ &\quad + (\sum_{j=1}^{N_j} (A_{\vec{n}_{ij}^*} \sin(d_{ij})))^2 \end{aligned} \quad (6)$$

이다. 여기서, N_j 은 C_{ik} 에 투영되는 메쉬들의 개수이다.

임의의 셀의 경계에 투영되는 메쉬 법선 벡터는 공격에 의하여 이웃하는 셀로 투영될 수 있다. 그러므로 제안한 방법에서는 이들 메쉬들을 그림 5에서와 같이 셀의 중심 단위 벡터와의 사이각을 이용하여 CEGI 분포 및 워터마크 삽입 대상에서 제외한다. 셀 C_k 에 투영된 메쉬 법선 벡터 \vec{n}^* 와 C_k 의 중점 벡터 \vec{BC}_k 와의 사이각 $\theta^* = \cos^{-1}(\vec{n}^* \cdot \vec{BC}_k)$ 은 $0 \leq \theta^* \leq \theta_{max}$ 의 범위를 갖는다. 여기서 θ_{max} 은 서로 인접한 C_k 및 C_{k+1} 들의 중점 벡터 \vec{BC}_k 및 \vec{BC}_{k+1} 와의

사이각의

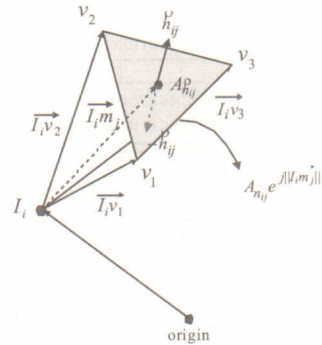


그림 4. i 번째 패치 P_i 로 분류된 j 번째 메쉬의 법선 벡터 \vec{n}_{ij}^* ($-\vec{n}_{ij}^*$) 및 CEGI의 복소 가중치.

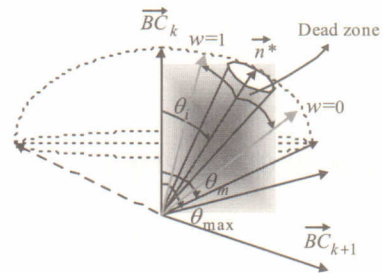


그림 5. 중점 벡터 \vec{BC}_k 를 가지는 셀 C_k 에 투영된 메쉬 법선 벡터 \vec{n}^* 및 \vec{BC}_k 와의 사이각의 범위 $0 < \theta^* < \theta_m$.

0.5배이다. θ_{max} 근처에 있는 \vec{n}^* 는 공격시 다른 셀로 투영될 수 있으므로, 제안한 알고리즘에서는 그림 5에서와 같이 실험적으로 구한 θ 범위

$$0 \leq \cos^{-1}(\vec{n}^* \cdot \vec{BC}_k) \leq \theta_m, \theta_m < \theta_{max} \quad (7)$$

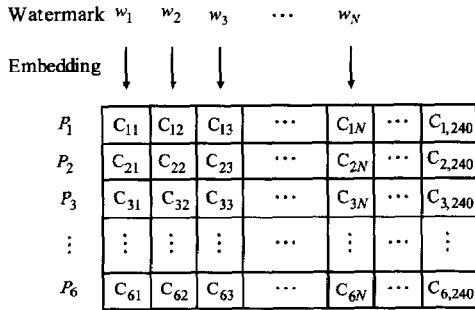
$$\theta_m = 8.6$$

안에 있는 \vec{n}^* 만을 C_k 로 투영시킨다. 6개 패치

의 CEGI 240개 셀에 대한 복소 가중치 크기를 위와 같은 방법으로 각각 구한다.

3. CEGI 셀의 복소 가중치 크기에 따른 워터마크 위치 결정

CEGI 분포에서 셀의 복소 가중치 크기가 크다는 것은 이 방향에 대한 메쉬들이 많이 분포함을 나타낸다. 다양한 공격에 의하여 변형된 모델에서는 원



C_{ij} : cell with j -th. complex weighting magnitude in P_i

그림 6. 패치별 CEGI 분포 중에서 복소 가중치의 크기에 대하여 내림차순으로 정렬된 셀에 동일한 워터마크 삽입.

모델에 비하여 메쉬의 꼭지점들의 위치가 변하거나, 또는 메쉬가 없어진다. 그러나, 변형된 모델이 원 모델의 형상을 유지하므로, CEGI 분포에서 투영되는 메쉬의 개수가 많은 셀들은 일정한 복소 가중치의 크기를 가진다. 제안한 방법에서는 이러한 셀들에 속하는 모든 메쉬 법선 벡터들의 평균 사이각을 이용하여 워터마크를 삽입한다.

각 패치별 CEGI 분포에서 셀의 복소 가중치 크기에 따라 순위를 정한 후, 워터마크 길이만큼 셀을 선택한다. 즉, n 개의 워터마크 삽입시 각 패치별 n 개의 순위 정보가 필요하며, 총 $6 \times n$ 개의 셀에 투영된 모든 법선 벡터들이 워터마크에 의하여 이동하게 된다. 제안한 방법에서는 1 비트의 워터마크를 6개 패치별 CEGI 분포 중 동일한 순위에 있는 6개의 C에 삽입한다. 즉, j 번째 워터마크 w_j 를 그림 6에서와 같이 각 패치의 CEGI 분포에서 j 번째 크기 순위를 가지는 셀

$\{C_{1j}, C_{2j}, \dots, C_{6j}\}$ 에 각각 삽입한다.

4. 워터마크 삽입

제안한 알고리즘에서는 셀에 투영된 모든 메쉬 법선 벡터들을 1비트의 랜덤 시퀀스 (random sequence)으로 이루어진 워터마크 정보에 의하여 이동시킨다. 즉, i 번째 패치의 CEGI 분포에서 j 번째 길이 순위인 C_j 에 투영된 모든 메쉬 법선 벡터 \vec{n}^* 와 C_j 의 중점 벡터 $\overrightarrow{BC_j}$ 와의 사이각 θ 가 j 번째 워터마크 w_j 에 따라 바뀌어지도록 각 메쉬의

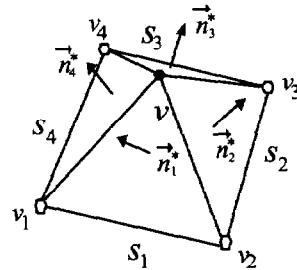


그림 7. 꼭지점 v 에 연결된 4개의 꼭지점 v_i 과 이들 3개의 꼭지점으로 연결된 각 메쉬 S_i 및 법선 벡터 \vec{n}_i^* , $i \in [1, 4]$.

연결된 꼭지점 v 의 위치를 이동시킨다. 여기서, 워터마크 w_j 가 1일 경우, 그림 5에서와 같이 θ 를 0° 의 방향으로 이동시키고, 워터마크 w_j 가 0일 경우 θ 를 θ_m 방향으로 이동시킨다. 그러나 하나의 꼭지점에는 그림 7에서와 같이 여러 개의 메쉬가 연결되어 있고, 이들 메쉬의 법선 벡터들이 다른 셀로 투영될 수 있다. 그러므로 하나의 꼭지점에 연결된 메쉬의 법선 벡터들이 워터마크 정보를 모두 만족하는 꼭지점의 위치를 찾아야 한다. 이를 위하여 제안한 알고리즘에서는 단계적 검색 방법을 이용하여 각 꼭지점의 탐색 범위 내에 비용 함수가 최소가 되는 최적의 위치를 찾는다.

4.1 워터마크 삽입 조건

제한한 알고리즘에서는 각 꼭지점에 연결된 메쉬들의 법선 벡터 방향들이 각각 워터마크 정보를 모두 만족하는 방향으로 이동하기 위하여 각 꼭지점의 최적 위치를 찾는다. 그러나 견고성 있는 워터마크를 삽입하기 위하여 두 가지 워터마크 삽입 조건이 필요하다.

첫 번째에는 하나의 꼭지점 v 에 연결된 각 메쉬의 법선 벡터들은 각각 다른 셀로 투영될 수 있다. 그러므로 워터마크 삽입 대상에서 제외되는 셀로 투영되는 메쉬들은 최적의 꼭지점 v 위치를 찾기 위한 비유함수의 인자 (factor)에서 제외된다.

두 번째에는 여러 가지 공격에 의하여 워터마크가 삽입된 각 셀의 평균 법선 벡터 방향이 바뀌어질 수 있다. 본 논문에서는 Stanford Bunny 모델을 이용하여 대표적인 공격 형태인 메쉬 간단화에 의하여 각 셀의 평균 법선 벡터 방향이 바뀌어지는 범위를 그림 8에서와 같이 실험적으로 구하였다. 이 그림에서 240개 셀의 평균 사이각 $\Delta\bar{\theta}_{ij}$ 은

$$\Delta\bar{\theta}_j = 1/6 \sum_{i=1}^6 (\bar{\theta}_{ij} - \bar{\theta}'_{ij}) \quad (8)$$

$$\text{where } \bar{\theta}_{ij} = 1/N \sum_{k=1}^N \cos^{-1}(\vec{n}_k \cdot \vec{BC}_j),$$

$$\bar{\theta}'_{ij} = 1/N' \sum_{k=1}^{N'} \cos^{-1}(\vec{n}_k \cdot \vec{BC}_j)$$

이며, $\bar{\theta}_{ij}$ 는 원 모델의 i 번째 패치 P_i 에서 j 번째 크기 순위를 가지는 셀 C_{ij} 상에서, 이 셀로 투영되는 모든 메쉬 법선 벡터 \vec{n}_k 와 C_{ij} 의 중점 벡터 \vec{BC}_j 와의 평균 사이각이며, N 은 C_{ij} 로 투영되는 메쉬들의 개수를 나타낸다. $\bar{\theta}'_{ij}$ 는 메쉬 간단화된 모델에서 원 모델의 같은 크기 순위의 셀 C_{ij} 상에서의 평균 사이각이며, N' 은 메쉬 간단화된 모델에서 C_{ij} 로 투영되는 메쉬들의 개수를 나타낸다. 이 그림에서 살펴보면, $\|C_{ij}\|$ 가 클수록 $\Delta\bar{\theta}_{ij}$ 가 작아진다. 그리고, 50번째 셀까지의 평균 사이각이 최대 1 degree를 벗어나지 않음을 알

수 있다.

제한한 알고리즘에서는 길이가 50인 1 비트 워터마크를 이들 셀의 평균 사이각에 삽입한다. 1 비트 워터마크 정보인 0과 1 사이를 나누는 문턱값 θ_{th} 는

$$\theta_{th} = \frac{1}{6 \times N_w} \sum_{i=1}^6 \sum_{j=0}^{N_w} \bar{\theta}_{ij} \quad (9)$$

이며, N_w 은 워터마크 길이인 50이다. 두 번째 워터마크 조건에 의하여, 워터마크가 삽입된 셀의 평균 사이각 $\bar{\theta}_{ij}$ 이 $[\theta_{th} - \Delta\theta_{max}, \theta_{th} + \Delta\theta_{max}]$ 범위 내에 있을 경우 공격에 의하여 다른 워터마크가 검출될 수 있다. 여기서 $\Delta\theta_{max}$ 는 1이다. 본 논문에서는 이의 범위를 데드존 (dead zone)이라 하며, 평균 사이각이 데드존을 벗어나도록 각 꼭지점의 위치를 탐색한다.

4.2 워터마크 삽입 조건에 맞는 꼭지점의 위치 탐색

제한한 알고리즘에서는 1 비트의 워터마크 정보를 각 패치별로 선택된 셀의 평균 사이각에 삽입한다. 셀의 평균 사이각을 워터마크 정보에 따라 바꾸기 위하여 셀에 투영되는 모든 메쉬 법선 벡터들을 이동하여야 한다. 그리고 메쉬 법선 벡터들은 이들에 연결되어 있는 꼭지점의 좌표 위치에 따라 바뀌어

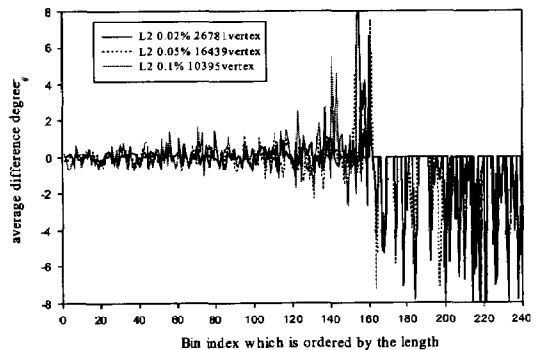


그림 8. Stanford Bunny 모델의 첫 번째 패치에서 각 셀의 평균 법선 벡터 방향 $\bar{\theta}$ 과 메쉬 간단화 공격된 모델에 대한 각 셀의 평균 법선 벡터 방향 $\bar{\theta}'$ 와의 차이.

진다. 그러므로 제안한 알고리즘에서는 3D 모델의 모든 꼭지점 좌표를 위의 워터마크 조건을 만족하는 위치로 이동시킨다.

먼저, 워터마크 삽입시 비가시성을 고려하기 위하여 현재 꼭지점 v 의 탐색 범위 내에서 최적의 꼭지점 v^* 를 찾는다. 여기서 탐색 범위는 현재 꼭지점 v 에 연결된 모든 꼭지점 $valence(v)$ 들 중 각 축의 최소값을 넘지 않도록 설정된다. 그림 7에서 $valence(v)$ 는 v_1, v_2, v_3, v_4 이다. 현재 꼭지점 v 의 좌표가 (x, y, z) 일 때, 각 축에 대한 탐색 범위는 $x \pm \Delta x, y \pm \Delta y, z \pm \Delta z$ 이다. 여기서 $\Delta x, \Delta y, z$ 는 각각

$$\begin{aligned} \Delta x &= 0.5 \times \min |x - v_i^x|_{v_i \in valence(v)}, \\ \Delta y &= 0.5 \times \min |y - v_i^y|_{v_i \in valence(v)}, \\ \Delta z &= 0.5 \times \min |z - v_i^z|_{v_i \in valence(v)} \end{aligned} \quad (10)$$

이며, v_i^x, v_i^y, v_i^z 는 v 에 연결된 꼭지점 v_i 의 각 축에 대한 좌표값이다. 현재 꼭지점 v 는 탐색 범위 내에서 비용 함수 $cost(v)$ 가 최소가 되는 최적의 꼭지점 v^* 위치로 이동한다. 그러나 탐색 범위 내에서 존재하는 모든 꼭지점 위치에 대한 비용 함수를 구하려면 처리 시간이 길어진다. 즉, 꼭지점의 좌표가 소수 6자리의 실수인 VRML 파일에서는 탐색 범위 내에 무수히 많은 꼭지점의 위치가 존재한다. 그러므로 제안한 알고리즘에서는 단계적 검색 방법을 이용하여 최적의 꼭지점을 구한다. 첫 번째 단계에서는 현재 꼭지점 v 에서 탐색 범위의 최종 위치에 있는 27개의 꼭지점 v'

$$\begin{aligned} v = \{ (x', y', z') \mid x' \in [x - \Delta x, x, x + \Delta x], \\ y' \in [y - \Delta y, y, y + \Delta y], \\ z' \in [z - \Delta z, z, z + \Delta z] \} \end{aligned} \quad (11)$$

들의 비용 함수 값을 각각 구한다. 그리고 현재 꼭지점 v 를 v' 위치에 대한 비용 함수 값들 중에서 최소의 값을 가지는 위치의 꼭지점 v^*

$$v^* = arg [min \{ cost(v') \}]_{v'} \quad (12)$$

으로 이동시킨 후, $\Delta x, \Delta y, z$ 를 각각 $\Delta x/2, \Delta y/2, z/2$ 으로 줄인다. 두 번째 단계에서는 줄어든 범위 내에서의 27개 꼭지점 중 최소의 비용 함수를 가지는 꼭지점으로 다시 이동시킨다. 이러한 과정을 모든 꼭지점들에 대하여 세 단계까지 반복적으로 수행한다.

4.3 비용 함수

제안한 알고리즘에서는 각 단계별로 꼭지점의 탐색 범위 내의 27개 위치에 따라 이에 연결된 메쉬들의 비용 함수를 각각 구한다. 4.1 절에서 나타낸 두 가지의 워터마크 삽입 조건에 의하여 비용 함수 $cost(v)$ 는

$$cost(v) = \sum_{m_i \in S(v)} a_{m_i} |\theta_{w_i} - \theta_{m_i}| \quad (13)$$

와 같다. 여기서 $S(v)$ 는 v 에 연결된 모든 메쉬들을 나타내며, θ_{m_i} 는 $S(v)$ 의 i 번째 메쉬 m_i 의 법선 벡터 \vec{n}_i 와 m_i 이 속해 있는 패치의 CEGI 분포 중 이 벡터가 투영된 셀의 중점 벡터와의 사이각을 나타낸다. \vec{n}_i 가 투영된 셀이 워터마크가 삽입되는 셀이면 a_{m_i} 는 1이고, 워터마크 삽입

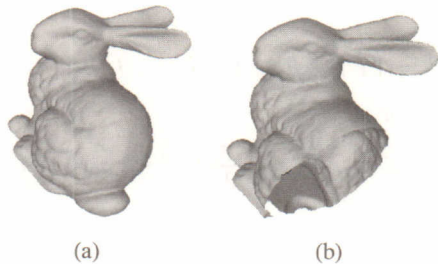


그림 9. $\Delta \phi = 45^\circ$ 및 $\Delta \theta = 45^\circ$ 만큼 회전된 (a) Stanford bunny 모델 및 (b) 절단된 Stanford bunny 모델.

대상 셀이 아니면 a_{m_i} 는 0이다. 그리고 \vec{n}_i 이 투영된 셀에 삽입될 워터마크 비트가 1이면 θ_{w_i} 가 0이고, 워터마크 비트가 0이면 θ_{w_i} 가 θ_{max} 이다.

여기서, 각 단계별 27개의 꼭지점 v' 에서 이에 연결된 메쉬 $S(v)$ 들 중, 원래의 셀로 투영되지 않는 메쉬가 존재하면 이 위치는 비용 함수에서 제외된다. 그리고 이들 꼭지점 v' 중에서 $S(v)$ 의 모든 메쉬가 두 번째 워터마크 삽입 조건인

$$\begin{aligned} w_i &= 1 \text{ if } 0 \leq \theta_i \leq \theta_{th} - \Delta\theta \\ w_i &= 0 \text{ if } \theta_{th} + \Delta\theta \leq \theta_i \leq \theta_{max} \end{aligned} \quad (14)$$

을 만족하면, 이 위치를 최적의 위치로 한다. 그러나 이 조건을 만족하는 위치가 없으면 위의 비용함수가 최소가 되는 위치로 이동한다.

모든 꼭지점들이 이에 연결된 메쉬 법선 벡터들의 워터마크 정보를 각각 만족하는 위치로 이동한다. 그리고 워터마크 삽입된 셀의 평균 사이각이 데드존 안에 있으면, 이들 셀에 속한 꼭지점들을 다시 위와 같은 방법으로 최적의 위치로 이동한다.

5. 재배열 (Realignment) 과정

회전, 이동 등과 같은 아핀 변환 (Affine transform) 공격에 대하여 워터마크 검출하기 전에 재배열 과정이 필요하다. 이는 아핀 변환 모델을 원 모델의 방향으로 이동하는 과정이다. 제안한 알고리즘에서는 패치 분할할 때 초기 패치 중점 \mathbf{I} 를 구하는 방법과 동일한 방법 및 \mathbf{I} 외의 유클리드 (Euclidian) 거리 d

$$\begin{aligned} d &= |I_i(\phi, \theta, \varphi) - I_j(\phi, \theta, \varphi)|, \\ &\text{for all} \\ &|\varphi| \leq \pi, |\theta| \leq \frac{\pi}{2}, |\phi| \leq \pi, \text{ and } i \neq j \end{aligned} \quad (15)$$

를 이용하여 아핀 변환 공격된 모델을 원 모델의 위치로 회전시킨다. 여기서 φ, θ, ϕ 는 각각 xy 평면, yz 평면, 및 xz 평면에서 회전각이다. 이는

회전된 모델의 각을 각각 $\pm \Delta\phi, \pm \Delta\theta, \pm \Delta\varphi$ 방향으로 이동하면서 이에 대한 초기 패치 중점 \mathbf{I}' 과 원 모델의 초기 패치 중점 \mathbf{I} 과의 MSE (Mean square error)가 최소화 되는 각을 찾음으로써 수행된다.

$$\begin{aligned} I &= \{(\Delta\phi, \Delta\theta, \Delta\varphi) : \\ &\min \left[\frac{1}{N} \sum_{k \in \{i, j\}} |d_k(\Delta\phi, \Delta\theta, \Delta\varphi) - d_k(0, 0, 0)|^2 \right] \end{aligned} \quad (16)$$

6. 워터마크 검출

제안한 알고리즘에 의하여 삽입된 워터마크의 검출 과정은 삽입 과정과 유사하며, 기존의 알고리즘과는 달리 원 모델이 필요 없다. 워터마크가 삽입되었다고 추정되는 모델에 대하여 워터마크 삽입 과정에서 얻은 각 패치의 중점 I 를 이용하여 6개 패치로 분류한다. 그리고 각 패치별 메쉬의 법선 벡터 및 CEGI 분포를 구한다. 각 패치별 워터마크가 삽입된 셀의 정렬 순서를 이용하여 셀에 투영되는 모든 메쉬 법선 벡터들과 셀의 중점 벡터와의 평균 사이각에 의하여 워터마크를 검출한다.

j 번째 비트의 워터마크 w_j 은 각 패치별 CEGI 분포에서 j 번째 순위를 가지는 셀 C_{ij} 에서의 평균 사이각 θ_{ij} 들의 전체 평균값 $\bar{\theta}_j$ 이

$$\bar{\theta}_j = 1/6 \sum_{i=0}^5 \theta_{ij} \quad (17)$$

특정 문턱값 θ_{th} 보다 작으면 w_j 은 1이고, θ_{th} 보다 크면 w_j 은 0으로 검출된다. 여기서 i 번째 패치의 j 번째 순위를 가지는 C_{ij} 에서 이에 투영된 메쉬의 법선 벡터들과 C_{ij} 의 중점 벡터 \overline{BC}_{ij} 와의 평균 사이각 θ_{ij} 은

$$\theta_{ij} = 1/N \sum_{k=0}^N \cos^{-1}(\overline{BC}_{ij} \cdot \vec{n}_k) \quad (18)$$

이며, N 은 C_{ij} 에 투영되는 메쉬 법선 벡터들의 수를 나타낸다. 여기서 문턱값 θ_{th} 은 식 (9)에 의하여 결정된다.

IV. 실험결과

제안한 3D 워터마킹 알고리즘의 성능을 평가하기 위하여 컴퓨터 모의 실험을 행하였다. 본 실험에서는 Stanford Bunny 및 Knots의 VRML 데이터를 사용하였고, 이 모델들은 그림 10에서와 같다. 제안한 알고리즘에서 사용되는 워터마크는 길이가 50인 가우시안 랜덤 시퀀스 (Gaussian random sequence)를 1비트로 변환하여 사용하였다. 그러므로 각 패치별 CEGI 분포에서 50개의 셀들이 필요하며, 한 모델에서 총 300개의 셀들에 워터마크가 삽입된다. 제안한 알고리즘에 의하여 워터마크가 삽입된 모델들은 그림 10에서와 같다. 이 그림에서 볼 수 있듯이 시각적으로 워터마크의 삽입 여부를 구분하기 어려우므로, 워터마크의 비가시성을 확인할 수 있었다.

제안한 워터마킹 알고리즘의 견고성을 평가하기 위하여 본 실험에서는 워터마크가 삽입된 모델에 대하여 메쉬 간단화, 절단, 및 꼭지점 잠음 첨가 등의 공격을 행하였다. 이들 공격에 대한 실험 결과는 표 I에서와 같다. 이 표에서 살펴보면 6개 패치에 대한 총 300개 셀에 각각 워터마크 1 비트를 삽입한다. 그리고 총 셀 에러 비트수는 300 비트 중에서 손실이 발생한 비트 수이며, 셀 당 오류 확률은 300 비트에 대한 비트 에러 수의 퍼센트이다. 그리고 1 비트의 워터마크는 동일한 길이 순위를 가지는 각 패치의 6개 셀들에 동일하게 삽입되므로, 300 비트 중 추출된 워터마크는 총 50 비트이다. 그러므로 워터마크 비트 에러 수는 50 비트 중에서 손실이 발생한 비트 수이며, 이에 대한 퍼센트는 워터마크 비트의 오류 확률로 나타내었다. 제안한 알고리즘에서는 패치별로 워터마크를 삽입하므로, 여러 셀에 대한 비트 손실이 발생하더라도 워터마크 비트 손실은 거의 발생되지 않음을 표에서 알 수 있다.

표면 간단화에 대한 실험에서는 MeshToSS[8]를 이용하여 워터마크가 삽입된 모델의 꼭지점의 개수를 다양한 비율로 줄였다. 표에서 살펴보면, 꼭지점

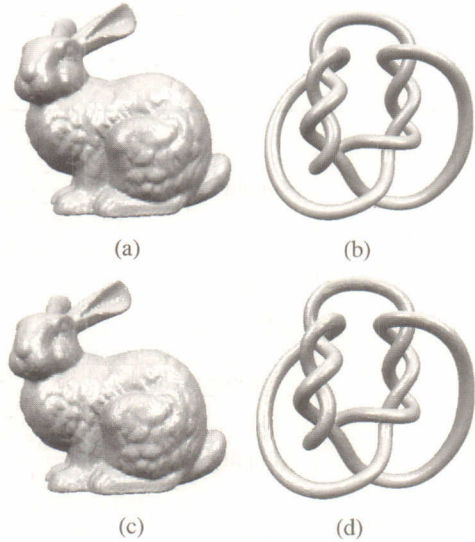


그림 10. (a) Stanford bunny (35947 꼭지점, 69451 면), (b) Knots (23232 꼭지점, 46464 면)와 제안한 알고리즘에 의하여 워터마크가 삽입된 (c) Stanford bunny 및 (d) Knots.

의 개수가 30 %까지 간단화된 모델에 대하여 워터마크 비트 손실이 거의 없음을 알 수 있다. 그리고 14 %까지 매우 많이 간단화된 모델에서도 워터마크의 92 %가 손실없이 검출되었음을 알 수 있다. 메쉬 간단화에서는 꼭지점의 개수가 줄여지면서 연결성 정보가 바뀌어진다. 그러므로 원 모델의 메쉬가 없어지거나 새로운 메쉬가 생성된다. 그러나 메쉬 간단화된 모델에서는 원 모델의 형상을 유지하므로, 이에 대한 CEGI 크기 분포는 원 모델의 CEGI 크기 분포와 유사하다. 꼭지점 랜덤화에 대한 실험에서는 랜덤하게 표본화된 꼭지점 v 의 각 좌표를 $v \times (1 + \alpha \times \text{uniform}(\))$ 로 바꾸었다. 여기서 변조도 α 는 0.008이며, $\text{uniform}(\)$ 는 [-0.5 0.5]의 범위를 가지는 균등한 랜덤 함수이다. 전체 꼭지점에 대하여 50 % 및 100 %로 랜덤하게 표본화된 꼭지점에 대하여 랜덤 잠음을 첨가한 모델에 대하여 견고성 실험을 행하였다. 메쉬 간단화 실험에서와 같이 꼭지점 랜덤화 공격에서는 메쉬의 연결성은 유지되지만, 꼭지점의 위치가 바뀌어지므로 메쉬의 표면 법선 벡터가 바뀌어진다. 그러나 제안한 알고리즘에서는 전체 셀 비트 손실이 4.6 % ~ 18.6 %이지만, 6개 패치의 각 셀에 삽입된 워터마크 비트가 손실없이 검출됨을 알 수 있다. 절단에 대한 실험에서는 워터마크가 삽입된 모델의 임의의 일정 부분을 제

Test 모델	Attacks	총 셀 오류 비트수 (300셀)	셀 당 오류 확률	위터마크 오류 비트수	위터마크 1개당 오류 확률	
Stanford bunny	Simplify	27875 (77.5 %)	6	2 %	0	0
		18427 (51.2 %)	21	7 %	0	0
		8964 (24.9 %)	55	18.3 %	1	2 %
		5330 (14.8 %)	77	25.6 %	4	8 %
		3594 (9.9 %)	97	32.3 %	10	20 %
	noise addition	50%	14	4.6 %	0	0
		100%	53	17.6 %	2	4 %
Cropping		17	5.6 %	0	0	
Cropping+Simplify (52.2 %)		31	10.3 %	1	2 %	
Knots	Simplify	14301 (61 %)	18	6 %	0	0
		9142 (39.35 %)	41	13.6 %	0	0
		6879 (29.6 %)	43	14.35 %	0	0
		5644 (24.29 %)	57	19 %	1	2 %
	noise addition	50%	28	9.3 %	0	0
		100%	56	18.6 %	1	2 %
	Cropping		8	2.6%	0	0
Cropping+Simplify (39.35 %)		43	14.3 %	3	6 %	

표 1. 다양한 공격에 대한 견고성 실험.

거하였다. 제한한 알고리즘에서는 패치별로 위터마크를 삽입하므로, 남아있는 부분에 속하는 패치에서는 위터마크가 여전히 남아 있다. 그러므로, 절단된 모델에서는 위터마크가 손실없이 검출됨을 표에서 알 수 있다.

위터마크가 삽입된 모델에 대하여 메쉬 간단화된 모델, 모든 꼭지점에 대하여 랜덤 잡음을 첨가한 모델, 그리고 임의의 일정 부분을 제거한 절단된 모델 들은 그림 11에서와 같다. 이 결과로부터 제한한 3D 위터마킹 알고리즘이 삽입한 위터마크에 대한 비가시성 및 다양한 공격에 대한 우수한 견고성을 가짐을 확인할 수 있었다.

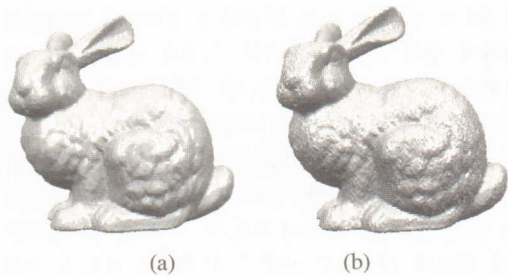
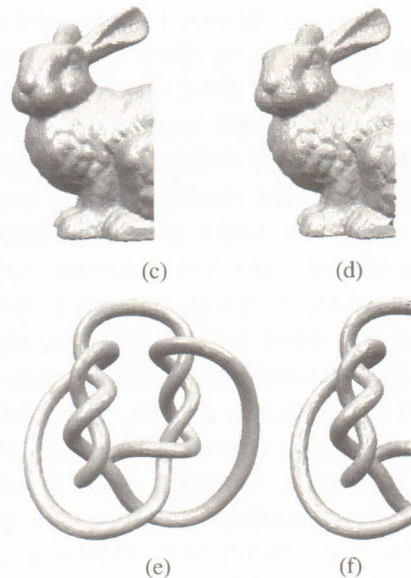


그림 11. (a) 꼭지점의 개수가 51.2 %으로 줄여진 메쉬 간단화, (b) 모든 꼭지점들에 대하여 잡음이 첨가, (c) 오른쪽 부분이 절단된, 및 (d) 절단 및 51.2%으로 메쉬 간단화된 Stanford bunny 모델 및 (e) 꼭지점의 개수가 39.35 %으로 줄여진 메쉬 간단화 및 (f) 절단 및 39.35 %으로 메쉬 간단화된 Knots 모델.

V. 결론

본 논문에서는 패치별 CEGI 크기 분포를 이용한 3D 메쉬 모델의 워터마킹 알고리즘을 제안하였다. 제안한 알고리즘에서는 기하학적 및 위상학적 변형에 대한 우수한 견고성을 가지기 위하여 동일한 워터마크 비트열을 6개로 분할된 패치의 CEGI 크기 분포에 각각 삽입한다. 그리고 워터마크 삽입에서는 인접한 꼭지점들 간의 최소 거리 내에 단계적 탐색 방법을 이용하여 최적의 꼭지점을 찾으므로 비가시성이 향상된다. 또한 워터마크 추출시 복잡한 재표본화 과정 및 원 모델이 필요 없다. 본 논문에서 제안한 알고리즘의 성능을 평가하기 위하여 VRML 3D 데이터의 Stanford Bunny 및 Knots 모델에 대하여 컴퓨터 모의 실험을 행하였다. 이 실험에서 제안한 워터마킹 알고리즘이 비가시성 및 견고성 면에서 우수함을 확인할 수 있었다.

제안한 알고리즘에서는 전체 형상이 비틀어지는 쉬어링 등의 공격에 대하여 견고성이 매우 약하다. 이와 같은 공격에 대하여 견고성이 우수한 알고리즘이 필요하다.

Acknowledgement

본 과제는 2003년도 Brain Busan 21 Project으로 수행한 연구결과입니다.

참 고 문 헌

- [1] R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modification," IEEE JSAC, pp. 551-560, 1998.
- [2] 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.
- [3] O. Benedens, "Geometry-Based Watermarking of 3D Models," IEEE CG&A, pp. 46-55, Jan./Feb. 1999.
- [4] Oliver Benedens, "Two high capacity methods for embedding public watermarks into 3D polygonal models," Proceedings of SPIE: Security and Watermarking of Multimedia Contents, SPIE, pp.329~340, 1999.
- [5] S. Kanai, 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, Dec. 1998.
- [6] E. Praun, H. Hoppe, A. Finkelstein, "Robust Mesh Watermarking," Proc. SIGGRAPH 99, pp. 49-56, 1999.
- [7] M. G. Wagner, "Robust Watermarking of Polygonal Meshes," Proc. Geometric Modeling & Processing 2000, pp. 201-208, Apr. 2000.
- [8] B-L. Yeo and M. M. Yeung, "Watermarking 3D Objects for Verification," IEEE CG&G, pp. 36-45, Jan./Feb. 1999.
- [9] T. Kanai, MeshToSS Version 1.0.1, <http://graphics.sfc.keio.ac.jp/MeshToSS/index.html>.
- [10] Boon-Lock, Y., Minerva. M, "Watermarking 3D Objects for Verification," IEEE Computer Graphics, Special Issue on Image Security, pp36-45, Jan./Feb. 1999,
- [11] H. S. Song, N. I. Cho, and J. W. Kim, "Robust Watermarking of 3D Mesh Models," Multimedia Signal Processing, 2002 IEEE Workshop, pp 332-335, 2002.
- [12] L. Zhang, R. Tong, F. Su, and J. Dong, "A mesh watermarking approach for appearance attributes," Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference. pp. 450-451, Oct. 2002.
- [13] S. Lee, T. Kim, B. Kim, S. Kwon, K. Kwon, and K. Lee, "3D Polygonal Meshes Watermarking Using Normal Vector Distributions," IEEE International Conference on Multimedia & Expo, Vol. III. pp. 105-108. July, 2003.
- [14] B. K. P. Horn, Robot Vision, The MIT Electrical Engineering and Computer Series, MIT Press, Cambridge, Mass., 1986.
- [15] B. K. P. Horn, "Extended Gaussian Image," Proc. IEEE, vol. 72. no. 12. pp. 1671-1686, Dec. 1984.
- [16] S. B. Kang and K. Ikeuchi, "The Complex

