

삼차원 폴리곤 메쉬를 위한 핑거프린팅 기법

허영, 안정환, 전정희, 호요성
광주과학기술원 정보통신공학과

A Fingerprinting Scheme for 3-D Polygonal Meshes

Yung Hur, Jeong-Hwan Ahn, Jeong-Hee Jeon and Yo-Sung Ho
Kwangju Institute of Science and Technology
{yunghur, jhahn, jhjeon, hoyo}@kjist.ac.kr

Abstract: 본 논문에서는 정보보호를 위해 삼차원 폴리곤 메쉬(3-D polygonal meshes)에 적용할 수 있는 새로운 핑거프린팅(fingerprinting)기법을 제안한다. 본 논문에서 제안한 핑거프린팅 기법은 유한 사영 기하학(finite projective geometries)을 토대로 고객의 수만큼 핑거프린트(fingerprint)를 만들고, 생성된 핑거프린트에 따라서 원래의 삼차원 폴리곤 메쉬 모델을 분할하고, 선택된 각각의 분할 메쉬에 핑거프린트의 삽입강도를 높이기 위한 추가 정보를 스펙트럼 영역에 삽입한다. 제안된 핑거프린팅 기법은 잡음공격, 압축공격 및 공모를 통한 핑거프린트 공격에도 강인하였다. 이 기법을 사용하면 불법으로 유통된 삼차원 폴리곤 메쉬 모델에서 삽입된 추가 정보를 바탕으로 핑거프린트를 추출한 후 공모자 검출 알고리즘을 적용하여 불법 유통에 책임이 있는 공모자들을 찾을 수 있다.

I. 서론

최근 멀티미디어 유통 기술의 발달로 오디오, 정지 영상, 동영상, 삼차원 데이터 등과 같은 디지털 콘텐츠(digital contents)를 여러 사람에게 쉽고 빠르게 분배하는 네트워크 시스템이 체계적으로 잘 발달되어 있다. 이런 기술은 디지털 콘텐츠의 불법적인 복제 및 유통과 같은 문제점을 유발할 수 있다.

디지털 워터마킹(digital watermarking)은 이러한 불법적인 복제 및 유통을 억제할 수 있는 기술이며, 사용하는 기법에 따라 강인한(robust) 워터마킹, 연약한(fragile) 워터마킹, 핑거프린팅 등의 분야로 나뉘어진다. 이 중 핑거프린팅은 디지털 콘텐츠의 불법적인 복제를 방지하기 위해서 고객의 고유한 ID 번호로써 서로 다른 핑거프린트를 디지털 콘텐츠에 삽입하는 기술이다. 결과적으로 고객들에게 서로 다른 복제물을 제공하게 된다. 그러나, 핑거프린트가 삽입된 복제물들은 이를 소유한 고객들이 서로 공모하여 비교 분석함으로써 쉽게 외부로부터 공격 받을 수 있다는 단점이 있다.

이러한 공모로 발생하는 문제점의 한가지 해결책으로 Dittmann은 정지영상에 대해 Boneh와 Shaw가 제안한 공모방지 핑거프린팅(collusion-secure fingerprinting) 기법 [3]과 이 기법에 효율적으로 적용할 수 있는 강인한 워터마킹을 결합한 방법을 제안하였다. 본 논문에서는 삼차원 폴리곤 메쉬 모델(3-D polygonal mesh models)에 적용할 수 있도록 Dittmann이 제안한 핑거프린팅 기법 [1]과 Ohbuchi가 제안한 워터마킹 기법 [2]을 결합한 새로운 핑거프린팅 기법을 제안한다.

II. 공모 방지 핑거프린팅 기법

1. 핑거프린팅 기법의 구성 요소

삼차원 폴리곤 메쉬에 적용하는 핑거프린팅 기법은 그림 1과 같이 핑거프린팅 알고리즘, 메쉬 분할, 워터마킹 알고리즘, 공모자 검출 알고리즘으로 이루어진다.

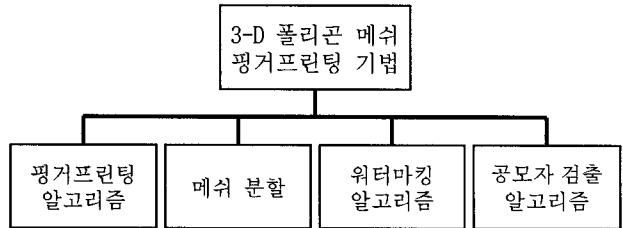


그림 1. 핑거프린팅 기법의 구성 요소

핑거프린팅 알고리즘은 고객에게 제공되는 삼차원 메쉬 모델에 삽입할 고객의 고유 정보인 핑거프린트를 생성한다. 메쉬 분할은 삼차원 메쉬 모델에 핑거프린트를 삽입할 위치를 정할 때, 핑거프린트를 삽입하기 전의 원래 메쉬(cover-메쉬)를 핑거프린트의 비트수에 따라 분할하여 비트의 정보에 따라 특정한 분할 메쉬를 선택하는데 사용된다. 워터마킹 알고리즘은 선택된 특정 분할 메쉬에 핑거프린트의 삽입 강도를 높이기 위해 추가 정보를 삽입하는데 이용된다. 마지막으로 공모자 검출 알고리즘은 외부로부터 공격받은 핑거프린트가 삽입된 메쉬(stego-메쉬)에서 핑거프린트를 추출해 공격에 기여한 사람 중 적어도 한 사람을 밝혀내는데 쓰인다.

2. d -detecting 핑거프린팅 기법

Dittmann은 최대 공모자의 수가 d 인 경우에 대해서 모든 공모자들을 색출할 수 있는 유한 사영 기하학을 기반으로 하는 d -detecting 핑거프린트 알고리즘을 제안하였다 [1]. d -detecting 핑거프린팅 기법을 적용하여 $q+1$ 개의 핑거프린트가 삽입된 stego-메쉬를 만들기 위하여 cover-메쉬로부터 분할메쉬가 $k=q^d+q^{d-1}+\dots+q+1$ 개 만큼 필요하다. 유한필드(finite fields) 상의 벡터공간에서 만들어지는 유한 사영공간(finite projective space) $PG(d, q)$ 를 생성하는 방법은 Hirschfeld가 그의 저서 [4]에서 자세히 언급하였으며 이 $PG(d, q)$ 를 토대로 길이가 k 비트인 핑거프린트가 만들어진다.

그림 2는 유한 사영 공간 $PG(2, 2)$ 에서 3개의 stego-메쉬를 만들 수 있는 2-detecting 핑거프린팅 기법을 보여준다.

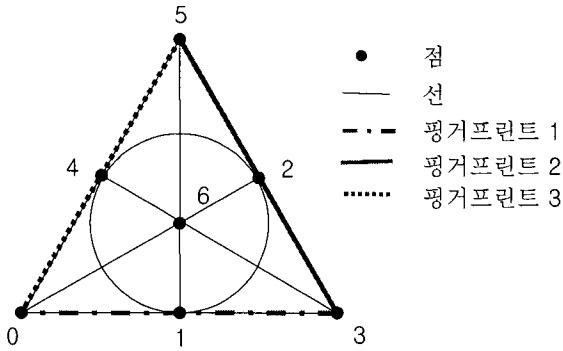


그림 2. 2-detecting 핑거프린팅 기법

PG(2,2)는 사영 공간 중 가장 작은 형태이며, 7개의 점과 7개의 선으로 이루어져 있다. 각각의 점은 핑거프린트 각각의 비트에 해당되며, 그림 2와 같이 PG(2,2)에서 핑거프린트를 생성할 경우에 핑거프린트1(FP1), 핑거프린트2(FP2), 핑거프린트3(FP3)은 각각 1101000, 0011010, 1001101이 된다. PG(2,2)에서 생성된 세 개의 핑거프린트들 중 어느 두개의 핑거프린트를 비교해도 서로 같은 위치에 1의 값을 갖는 비트가 항상 존재하게 된다. 이 위치의 비트가 공모를 한 경우에도 살아남은 정보가 되어 공모자 검출 알고리즘을 이용하여 공모자를 찾을 수 있다.

III. 핑거프린팅 알고리즘

1. 알고리즘 개요

본 논문에서 제안한 핑거프린팅 기법에서는 그림 3과 같이 핑거프린트를 삽입하고 추출하여 공모자를 찾을 수 있다

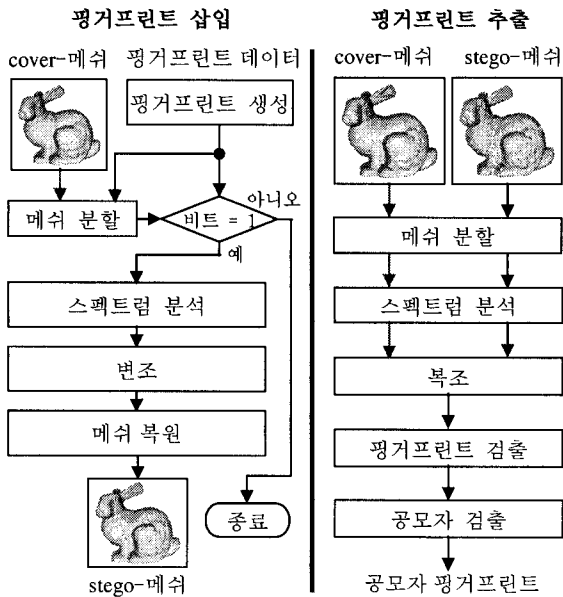


그림 3. 핑거프린트 삽입 및 추출

2. 핑거프린트 삽입

핑거프린트 생성: PG(d,q)을 토대로 d-detecting 핑거프린팅 기법을 적용하여 k비트로 구성된 q+1개의 핑거프린트를 생성한다. 이렇게 생성된 각각의 핑거프린트는 공

모 공격에 대하여 강인하도록 잉여 변수 r만큼 반복함으로써, cover-메쉬에 삽입할 핑거프린트의 전체 비트수는 kxr이 된다. 잉여 변수 r은 cover-메쉬의 꼭지점 개수 v와 분할 메쉬에 삽입될 추가 정보의 비트수 m에 따라 최대 kx2.mxr=v까지 정할 수 있다.

메쉬 분할: cover-메쉬에 공모방지 핑거프린팅 기법을 효율적으로 적용하기 위해 MeTis [5]를 이용하여 메쉬를 분할하였다. 분할 메쉬의 개수 s는 잉여 변수 r과 같은 방법으로 최대 kx2.mxs=v (s ≥ r) 까지 정할 수 있다. s개의 분할 메쉬 중 kxr개를 랜덤하게 선택한 후 핑거프린트의 kxr비트와 선택된 분할 메쉬를 서로 일대일 대응시킨다. 분할 메쉬와 대응된 비트가 1인 경우의 분할 메쉬만을 선택하여 다음 과정을 수행한다.

스펙트럼 분석: 분할된 메쉬는 삼차원 폴리곤 메쉬의 기하학 정보를 압축하기 위하여 Karni와 Gotsman [6]이 제안한 방법을 Ohbuchi[2]가 수정한 방법을 이용하여 스펙트럼 분석을 한다. 분할 메쉬가 n개의 꼭지점을 가지고 있을 때 이 메쉬의 연결성 정보를 가지고 nxn 크기의 Kirchhoff 행렬 K를 식 (1)과 같이 구한다.

$$K = D - A \quad (1)$$

행렬 D는 i번째 꼭지점의 valence 값 di를 대각선 원소 Di로 가지는 nxn 대각선행렬이다. 행렬 A는 i번째 꼭지점과 j번째 꼭지점이 서로 이웃하여 연결되어 있는 경우 1을, 그렇지 않은 경우에는 0의 값을 가지는 원소 aij로 구성되는 nxn 인접 행렬이다.

크기가 nxn인 Kirchhoff 행렬 K를 고유치 분해를 하면 n개의 고유치 λi와 n개의 n차원 고유벡터 wi (1 ≤ i ≤ n)가 생성된다. 꼭지점 좌표값 vi = (xi, yi, zi) (1 ≤ i ≤ n)를 각각 x 좌표값, y 좌표값, z 좌표값으로 분리한 후 각각을 식 (2)와 같이 정규화가 된 고유벡터 ei에 사영시키면 n 개의 메쉬 스펙트럼 계수 벡터 ri = (rsi, rti, rui)를 얻을 수 있다. 첨자 s, t, u는 각각 공간 좌표축 x, y, z에 대응하는 메쉬 스펙트럼 영역에서의 좌표축이다.

$$e_i = w_i / \|w_i\| \quad (1 \leq i \leq n) \quad (2)$$

변조: 핑거프린트의 삽입강도를 높이기 위하여 m비트 길이의 추가 정보 a=(a1,a2,...,am), ai ∈ {0,1}를 Ohbuchi 방식[2]을 이용해 각각의 분할 메쉬에 삽입하였다. m차원 벡터 a의 각 비트 ai는 chip rate c 만큼 식 (3)과 같이 복제되어 mxcc차원 벡터 b=(b1,b2,...,bmc), bi ∈ {0,1},가 된다.

$$b_i = a_j, \quad jxc \leq i < (j+1)xc \quad (3)$$

벡터 b는 다시 식 (4)에 의하여 벡터 b'=(b'1,b'2,...,b'mc), bi' ∈ {-1,1}으로 변환된다.

$$b'_i = \begin{cases} -1, & \text{if } b_i = 0 \\ 1, & \text{if } b_i = 1 \end{cases} \quad (4)$$

스펙트럼 영역의 s축에 있는 i번째 스펙트럼 계수 rsi를 변조하는 경우에 대해서 먼저 고려해 보자. 스펙트럼 영역의 t, u축에 있는 다른 스펙트럼 계수들도 동일한 방법으로 변조된다. 스펙트럼 계수를 변조하기 위하여 stego-key kw로 생성되는 PRNS(pseudo random number sequency)인 pi ∈ {-1,1}와 변조 크기변수 α (α>0)가 필요하다. 추가 정보가 삽입된 스펙트럼 계수 r̂s,i는 식 (5)와 같이 계산된다.

$$\hat{r}_{s,i} = r_{s,i} + b'_i \cdot p_i \cdot \alpha \quad (5)$$

위와 동일한 방법을 t 축과 u 축에 있는 스펙트럼 계수에 적용하면 추가 정보가 삽입된 스펙트럼 계수 $\hat{r}_i = (\hat{r}_{s,i}, \hat{r}_{t,i}, \hat{r}_{u,i})$ 가 만들어진다. 여기서 사용한 stego-key k_w 값은 핑거프린트 추출과정에서 핑거프린트 삽입과정에서 사용된 동일한 PRNS를 만들기 위해 이용된다.

메쉬 복원: 식 (3)에서 구한 정규화된 고유벡터 e_i 를 식 (6)과 같은 방법으로 추가 정보가 삽입된 스펙트럼 계수 $\hat{r}_i = (\hat{r}_{s,i}, \hat{r}_{t,i}, \hat{r}_{u,i})$ 를 곱한 후 첨자 i 에 대하여 1부터 n 까지 더하면, 추가 정보가 삽입된 분할 stego-메쉬를 얻을 수 있다.

$$\begin{aligned} (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)^T &= \hat{r}_{s,1} e_1 + \hat{r}_{s,2} e_2 + \dots + \hat{r}_{s,n} e_n \\ (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n)^T &= \hat{r}_{t,1} e_1 + \hat{r}_{t,2} e_2 + \dots + \hat{r}_{t,n} e_n \\ (\hat{z}_1, \hat{z}_2, \dots, \hat{z}_n)^T &= \hat{r}_{u,1} e_1 + \hat{r}_{u,2} e_2 + \dots + \hat{r}_{u,n} e_n \end{aligned} \quad (6)$$

$k \times r$ 개의 핑거프린트 비트 중 1인 비트와 대응된 분할 메쉬에 추가 정보를 삽입한 후 분할 메쉬가 결합된 핑거프린트가 삽입된 하나의 stego-메쉬로 만들기 위해 분할 stego-메쉬가 서로 공유하는 꼭지점이 있는 경우에는 각 분할 stego-메쉬에 있는 공유 꼭지점 좌표값의 평균값을 계산하여 이용한다.

그러나, 추가 정보가 삽입되지 않은 분할 메쉬와 분할 stego-메쉬가 이웃한 꼭지점 좌표값의 경우는 분할 stego-메쉬의 꼭지점 좌표값을 그대로 이용한다. 이 방법으로 메쉬 복원으로 인한 삽입된 추가 정보의 왜곡을 최소화하였다.

3. 핑거프린트 추출

핑거프린트가 삽입되어 있는 stego-메쉬나 외부로부터 공격받은 핑거프린트가 삽입된 stego-메쉬로부터 핑거프린트를 추출하기 위해서는 cover-메쉬와 핑거프린트 삽입 과정에 사용되었던 입력 변수들이 필요하다.

메쉬 분할: Cover-메쉬와 Stego-메쉬를 핑거프린트 삽입 과정에서와 동일한 수의 메쉬로 분할한다.

스펙트럼 분석: 핑거프린트를 삽입할 때와 동일한 방법으로 분할 Cover-메쉬와 분할 Stego-메쉬로부터 스펙트럼 계수를 각각 구한다.

복조: i 번째 분할 메쉬로부터 이 메쉬에 삽입했던 추가 정보를 구하기 위하여 우선 i 번째 분할 cover-메쉬와 분할 stego-메쉬로부터 구한 스펙트럼 계수 $r_{s,i}, \hat{r}_{s,i}$ 에 대하여 추가 정보를 삽입할 때 사용했던 k_w 을 이용하여 PRNS를 생성한다. 이 PRNS값을 $r_{s,i}$ 와 $\hat{r}_{s,i}$ 의 차에 곱한 뒤에 식 (7)과 같이 계산하여 q_j 를 구한다.

$$\begin{aligned} q_j &= \frac{1}{3} \sum_{l \in \{s,t,u\}} \sum_{i=j-c}^{(j+1)c-1} (\hat{r}_{l,i} - r_{l,i}) \cdot p_i \\ &= \frac{1}{3} \sum_{l \in \{s,t,u\}} \sum_{i=j-c}^{(j+1)c-1} b'_i \cdot \alpha \cdot p_i \end{aligned} \quad (7)$$

q_j 로부터 식 (8)를 이용하여 삽입된 추가 정보의 j 번째 비트 a_j 를 구할 수 있다.

$$a_j = \text{sign}(q_j) \quad (8)$$

핑거프린트 검출: i 번째 분할 stego-메쉬로부터 구한 m 비트의 추가 정보 a' 와 핑거프린트를 삽입할 때 사용한 m 비트 추가 정보 a 를 비교하여 동일한 경우 i 번째 분할 stego-메쉬에 대응하는 핑거프린트의 비트를 1로 설정하고 그렇지 않은 경우에 0으로 설정한다. 위와 동일한 방법을 모든 분할 stego-메쉬에 적용하여 cover-메쉬에 삽입한 $k \times r$ 비트의 핑거프린트를 추출한다.

공모자 검출: 공모 공격에 가담한 공모자를 검출하기 위하여 stego-메쉬에서 추출한 $k \times r$ 비트의 핑거프린트를 핑거프린트를 생성할 때 만들어진 $q+1$ 개의 모든 핑거프린트와 비교하여 적어도 한사람 이상의 공모자를 찾을 수 있다. 잉여 변수 r 값이 클수록 공모 공격에 강인하게 되므로 정확한 공모자를 검출할 확률을 높일 수 있다.

IV. 실험 및 결과

그림 4는 실험에 사용된 삼차원 메쉬 모델인 Bunny (15095 꼭지점, 30019 면), Fandisk (6475 꼭지점, 12946 면), Horse (15002 꼭지점, 30000 면) 모델과 각각의 모델에 대하여 30, 15, 22개로 분할된 메쉬 분할을 보여준다. Bunny 모델과 Horse 모델은 각각 Stanford 대학과 Cyberware Inc.에서 제공하는 모델을 단순화시켰다.

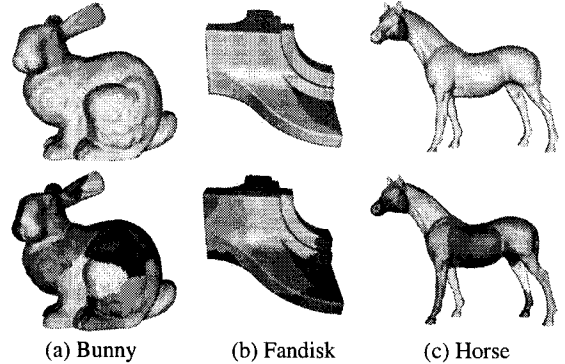


그림 4. 실험 메쉬 모델 및 메쉬 분할

본 논문의 실험을 위해 II-2절에서 언급한 세 개의 핑거프린트를 Bunny 모델에 적용하였으며, 핑거프린트를 삽입하기 위하여 사용된 변수 값은 $v=15095, k=7, r=3, 30$ -메쉬 분할, $m=35, c=3, \alpha=0.001$ 이다.

표 1. FP에 따라 임의로 선택된 분할메쉬

메쉬	5	9	8	10	21	13	22	14	12	6	29	20	28	19	11	4	18	30	2	16	27	
FP1	1	1	0	1	0	0	0	1	1	0	1	0	0	0	1	1	0	1	0	0	0	0
FP2	0	0	1	1	0	1	0	0	0	1	1	0	1	0	0	0	0	1	1	0	1	0
FP3	1	0	0	1	1	0	1	0	0	0	1	1	0	1	0	1	0	0	0	1	1	0

30개로 분할된 메쉬를 표 1과 같이 임의로 선택하여 35비트의 추가 정보를 삽입한 stego-메쉬를 다른 색으로 구별해서 그림 5에 나타내었다.

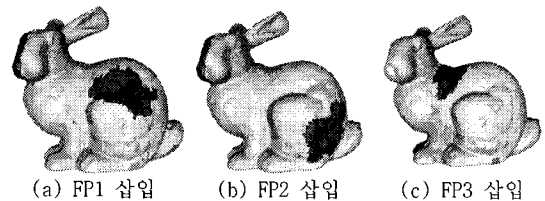


그림 5. Bunny 모델에 삽입한 핑거프린트

본 논문에서 제안한 핑거프린팅 기법의 성능을 평가하기 위해 추가로 삽입되는 정보에 적용된 워터마킹 방법이 외부 공격으로부터 얼마나 강한지를 실험했고, 삽입된 핑거프린트에 대한 공모 실험도 수행하였다.

우선 워터마킹 알고리즘에 대한 공격 실험으로 Praun이 사용한 잡음 공격[7]과 3-D 메쉬의 꼭지점 좌표값에 압축 공격을 사용하였다. 공모 공격으로는 두 사람이 서로 공모하여 두 메쉬의 꼭지점 좌표값의 평균값을 취하는 방법을 적용하였다.

그림 6은 Bunny 모델에 대한 잡음 공격, 압축 공격, 공모 공격에 대한 결과를 보여준다. 그림 6(c), 그림 6(f), 그림 6(i)의 메쉬에서 동일 비트 1값을 가지는 분할 메쉬는 다른 색으로 나타내었다.

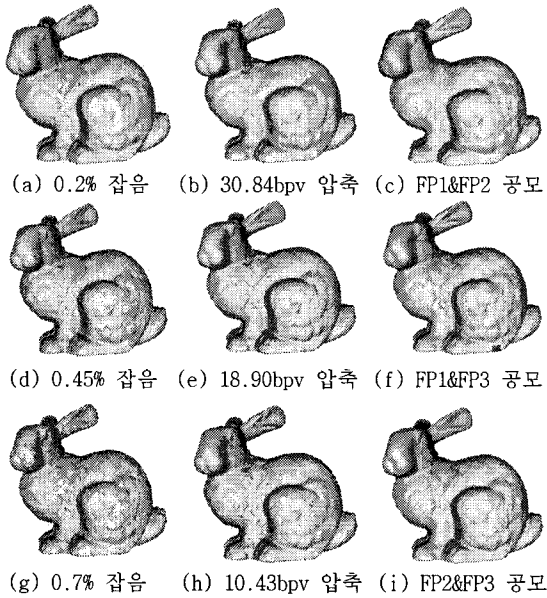


그림 6. Bunny 모델에 대한 공격

표 2는 Bunny, Fandisk, Horse 모델에 대한 그림 5에서 제시한 공격으로 실험한 결과를 보여주고 있다. 표 2의 수치는 Ohbuchi가 정의한 비트 오류율[2]을 이용하여 추가 정보가 삽입된 분할 stego-메쉬에서 구한 비트 오류율의 평균값이다. 공모자 검출율은 본 논문에서는 두 사람 공모에 의한 메쉬 평균 공격에 대하여 공모에 가담한 두 사람 중 적어도 한명을 검출할 확률을 의미한다. 실험 결과로부터 제안된 핑거프린팅 기법이 잡음 공격, 압축 공격 및 두 사람에 의한 메쉬 평균 공격에 대하여 강인함을 알 수 있다.

V. 결론 및 향후 과제

본 논문에서는 삼차원 폴리곤 메쉬에 적용할 수 있는 새로운 핑거프린팅 기법을 제안하였다. 실험을 통해 제안된 핑거프린팅 기법이 잡음 공격, 압축 공격 및 공모를 통한 핑거프린트 공격에도 강인함을 확인하였다. 향후 과제로는 사용되는 핑거프린트의 비트수가 d 값과 q 값에 따라 급격하게 증가하는 단점을 보완하여 핑거프린트의 비트수를 줄이는 연구와 메쉬 단순화와 같은 메쉬의 연결성 정보에 대한 공격에 대해서 강인한 방법에 대한 연구를 들 수 있다. 제안한 방법으로 추후 불법적으로 유통된 삼차원 폴리곤 메쉬 모델에서 삽입된 추가 정보로 핑거프린트를 추출하여 불법 유통에 책임이 있

는 공모자들을 찾을 수 있을 것이다.

표 2. Bunny, Fandisk, Horse 모델의 실험 결과

공격	3-D 메쉬 모델 ($\alpha=0.001$)		
	Bunny ($r=3, c=3$)	Fandisk ($r=2, c=3$)	Horse ($r=3, c=3$)
공격 없음	0	0	0
0.2% 잡음	0	0	0
0.45% 잡음	0	0.0428	0
0.7% 잡음	0	0.1571	0
약 70% 압축	0	0.0285	0
약 80% 압축	0	0	0
약 90% 압축	0	0.0285	0
0.7% 잡음과 약 90% 압축	0	0.1380	0
X	2 사람 공모에 의한 공격		
	메쉬 평균		
	Bunny	Fandisk	Horse
비트 오류율	0	0.0057	0
공모자 검출율	100%	100%	100%

감사의 글

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

참고문헌

- [1] J. Dittmann, "Combining Digital Watermarks and Collusion Secure Fingerprints for Customer Copy Monitoring," *Proc. IEE Seminar Sec. Image & Image Auth.*, pp. 128-132, March 2000.
- [2] R. Ohbuchi, S. Takahashi, T. Miyazawa and A. Mukaiyama, "Watermarking 3D Polygonal Meshes in the Mesh Spectral Domain," *Proc. Graphics Interface*, pp. 9-17, June 2001.
- [3] D. Bohch and J. Shaw, "Collusion-Secure Fingerprinting for Digital Data," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 1897-1905, Sept. 1998.
- [4] J.W.P. Hirschfeld: *Projective Geometries over Finite Fields*, Oxford Univ. Press, 2nd Ed. 1998.
- [5] G. Karypis and V. Kumar, MeTis: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-reducing Orderings of Sparse Matrices. Version 4.0, Univ. of Minnesota, Dept. of Comp. Sci., 1998.
- [6] Z. Karni and C. Gotsman, "Spectral Compression of Mesh Geometry," *Proc. SIGGRAPH*, pp. 279-286, July 2000.
- [7] E. Praun, H. Hoppe and A. Finkelstein, "Robust Mesh Watermarking," *Proc. SIGGRAPH*, pp. 49-56, Aug. 1999.