삼차원 메쉬 모델에 적용한 공모방지 핑거프린팅 기법

허 영*, 전 정 희*, 호 요 성*, 문 경 애**
*광주과학기술원, **한국전자통신연구원
전화: 062-970-2263 / 핸드폰: 011-326-6567

A Collusion-secure Fingerprinting Scheme for Three-dimensional Mesh Models

Yung Hur, Jeonghee Jeon*, Yo-Sung Ho*, and KyungAe Moon**
*Kwangju Institute of Science and Technology (K-JIST)
**Electronics and Telecommunications Research Institute (ETRI)
E-mail : {yunghur, jhjeon, hoyo}@kjist.ac.kr, kamoon@etri.re.kr

Abstract

This paper presents a new collusion secure fingerprinting scheme for 3-D meshes to protect copyright of data. In the proposed scheme, we make the same number of fingerprints as the total number of customers based on the finite projective geometries, partition a 3-D mesh model related to the total number of bits for each fingerprint, and then embed a certain additive information into each selected sub-mesh in the DCT domain. Experimental results show that the proposed scheme is robust to additive random noise, geometric transform, 3-D mesh compression, and fingerprint attack by two traitors' collusion. Using this scheme, we can find traitors responsible for the collusion attack causing unauthorized copy and distribution.

I. 서 론

최근 멀티미디어 유통 기술의 발달로 오디오, 정지영상, 동영상, 삼차원 그래픽스 등과 같은 디지털 데이터를 여러 사람에게 쉽고 빠르게 전송하는 네트워크 시스템이 체계적으로 잘 발달되어 있다. 이러한 전송로를 통해 전달되는 디지털 데이터는 다양한 이점들을 제공해 주지만, 이와 동시에 원본과 동일한 대량의 복사본이 불법적으로 쉽고 빠르게 배포될 수 있다는 사실은 저작권 보호에 심각한 위협이 되고 있다.

디지털 워터마킹은 이러한 불법적인 복제 및 유통을 억제할 수 있는 기술로써 대두되었으며, 저작권 보호, 인증, 복사 보호, 방송 모니터링, 핑거프린팅에 이용되고 있다. 이 중 핑거프린팅은 불법적으로 배포되고 있는 디지털 데이터의 복사본을 감시하거나 추적하기 위하여 사용자의 고유정보인 핑거프린트를 디지털 데이터에 삽입하는 기술이다. 여기서 핑거프린트는 인간의 지문처럼 어느 데이터를 다른 비슷한 동류의 데이터와 구별지을 수 있는 특징을 내포한다. 하지만, 핑거프린트가 삽입된 복사본은 이를 소유한 사용자들이 서로 공모하여 비교, 분석함으로써 외부로부터 공격받을 수 있는 단점이 있다.

이러한 공모로 발생하는 문제점을 해결하는 방법으로 Boneh와 Shaw는 c-secure 코드를 이용한 공모방지 핑거프린팅 기법을 제안하였다[1]. 구체적인 응용은 Dittmann에 의해서 제안되었으며, 정지영상에 대해 유한사영기학학(finite projective geometries)에 기반한 공모방지 핑거프린팅 알고리즘과 이 기법에 효율적으로 적용할 수 있는 강인한 워터마킹 알고리즘을 결합한 방법을 사용하였다[2]. 하지만 정지영상에 적용되는 핑거프린팅 기법은 최근 CAD(computer aided design)와 가상현실(virtual reality) 등에서 널리 이용되고 있는 삼차원 메쉬 모델에는 적합하지 않다. 따라서 본 논문은 삼차원 메쉬 모델에 효율적으로 적용할 수 있는 공모방지 핑거프린팅 기법을 제안한다.

II. 핑거프린트 생성과 공모자 검출

본 논문은 기본적으로 Dittmann이 제안한 핑거프린트 생성 방식과 공모자 추적 알고리즘을 이용한다[2]. 우선 공모자의 최대 수가 d인 경우에 대해서 모든 공모자들을 검출할 수 있는 유한사영공간 PG(d,q)을 만들고, 생성된 PG(d,q)을 기반으로 공모자를 검출할 수

있는 d-detecting RNC(rational normal curves) 기법을 제안했다. PG(d,q)는 유한체(finite field) GF(q) ($q=p^a$, p는 소수, $a\geq 1$) 상의 d차원 사영공간이다. 이때, RNC R은 식(1)과 같이 초평면(hyperplane)의 집합체이다.

$$R=\{[1:t:t^2:\ldots:t^d] | t\in GF(q)\}\cup\{[0:0:0:\ldots:0:1]\} \quad (1)$$

식 (1)을 기반으로 핑거프린트를 생성하는 방법은 다음과 같다. 먼저, 사용자 i 에 대해서, $i\in$GF(q)와 $H_i=[1:i:i^2:\ldots:i^d]$ ($H_i\in$R)을 선택하고, 동차좌표가 $(a_0,a_1,\ldots,a_{d-1},(-a_0-a_1i-\ldots-a_{d-1}i^{d-1})/i^d)$인 점들과 연계되어 있는 모든 마크들을 선정하여 이 마크들과 대응되는 비트를 1로 설정하여 핑거프린트를 생성한다.

공모자를 검출하는 방식은 불법 복사본으로 의심되는 디지털 데이터에서 핑거프린트를 추출하여 1로 설정되어 있는 비트, 즉, 마크와 대응하는 PG(d,q) 공간의 점을 선정한다. 이 점들 중에서 임의의 한 점 $(1,a_1,\ldots,a_d)$을 선택하여 방정식 $1+a_1t+a_2t^2+\ldots+a_dt^d=0$을 $t\in$(GF(q)) 에 대해 계산한다. 핑거프린트를 생성할 때, 각각의 점과 대응하는 초평면에 대한 정보가 사용자 목록에 기록되므로 공모자의 초평면 인덱스도 포함된다. 따라서 공모자 검출은 공모자들이 서로 공모해서 찾은 모든 마크를 삭제할 경우, 사용자 목록에서 모든 공모자들이 서로 공유하는 마크가 탐지되기 때문에 모든 공모자들을 찾아낼 수 있다. 또한 공모자들이 서로 공모하여 발견한 마크들 중 몇 개의 마크만을 삭제한다고 할지라도 검출된 마크들에 다수결의 원칙을 적용하여 공모자를 검출할 수 있다.

III. 핑거프린트 삽입과 추출

공모방지 핑거프린팅 알고리즘에 의해서 생성된 핑거프린트를 삼차원 메쉬 모델에 삽입할 때 비지각성(imperceptibility)과 강인성(robustness)을 갖는 디지털 워터마킹 알고리즘을 적용하는 일은 매우 중요하다. 본 논문에서는 이 두 요건을 고려하여 핑거프린트의 비트 정보에 따라 선택된 분할 메쉬의 기하학 정보와 연결성 정보를 바탕으로 저작권에 관한 데이터인 추가 정보를 DCT 영역에서 삽입하였다.

3.1 핑거프린트 삽입 알고리즘

그림 1은 핑거프린트를 삽입하는 과정을 보여준다. PG(d,q)를 토대로 최대 공모자의 수가 d인 경우, 이를 검출할 수 있는 d-detecting RNC 핑거프린팅 기법을 적용하여 $k=q^d+q^{d-1}+\cdots+q+1$ 비트로 구성된 $q+1$개의 핑거프린트 FP_1, FP_2,…, FP_{q+1}를 생성한다.

이렇게 생성된 각각의 핑거프린트는 외부 공격으로부터 강인하도록 홀수인 잉여변수 r만큼 반복한다. 잉여변수 r은 cover-메쉬의 꼭지점 개수 v, 추후 분할 메쉬에 삽입할 추가 정보의 비트수 m, chip rate c 에 따라 $m\times c$ 만큼 DCT 영역에서 중간주파수 대역만을 선택하여 삽입함으로써 최대치 r_{max} = $\lfloor v/(k\times 2mc)\rfloor$ 까지 정할 수 있다. 따라서 각각의 핑거프린트의 비트수는 $k\times r$ 이며, 이때 i번째 핑거프린트를 FP'_i라고 하자.

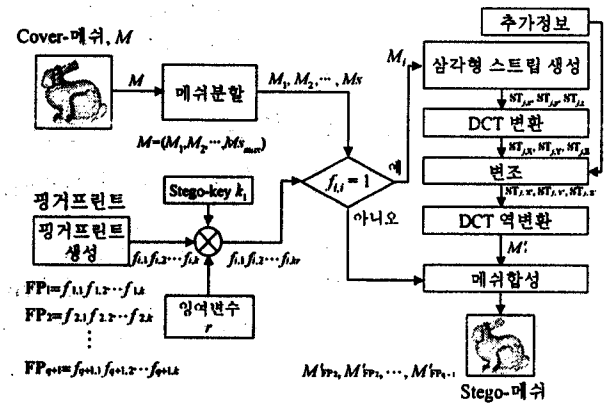


그림 1. 핑거프린트 삽입 알고리즘

cover-메쉬에 공모방지 핑거프린팅 기법을 효율적으로 적용하기 위해 MeTis[3]를 이용하여 메쉬를 분할하였다. 분할 메쉬의 개수 s는 조건 $s\geq k\times r$을 만족하도록 최대치 s_{max} = $\lfloor v/(2mc)\rfloor$ 까지 정할 수 있다.

stego-key k_1으로 생성되는 PRNS(pseudo random number sequency)에 따라 총 s개의 분할 메쉬 중 $k\times r$ 개의 분할 메쉬를 선택한 후, FP'_i와 선택된 분할 메쉬를 일대일로 대응시킨다. 여기서 stego-key k_1은 공모 공격에 의하여 각각의 분할 메쉬와 핑거프린트 비트들 사이의 대응 관계를 모르게 하는 역할을 담당한다.

핑거프린트의 i번째 비트가 1인 경우, 이 비트와 대응된 분할 메쉬 M_i을 선택하여 M_i의 기하학정보와 연결성정보를 이용하여 SGI 알고리즘을 이용한 스트립 생성방법[4]으로 삼각형 스트립을 생성한다. 표현의 단순화를 위하여 j번째 삼각형 스트립 ST_j에 포함되어 있는 N개의 꼭지점의 좌표값 $v_i=(x_i, y_i, z_i)$, $(1\leq i\leq N)$에 대해서 DCT 변환을 한 후에 추가 정보를 삽입하는 경우에 대하여 먼저 고려하자.

DCT 변환을 위해 우선 N개의 꼭지점 좌표값들을 x 좌표값, y 좌표값, z 좌표값끼리 각각 분리하여 N-차원의 벡터 $ST_{j,x}$, $ST_{j,y}$, $ST_{j,z}$를 생성한다. 식 (1)은 벡터 $ST_{j,x}$에 적용한 1차원 DCT 변환을 나타내며, $ST_{j,y}$, $ST_{j,z}$에도 동일하게 적용하여 DCT 계수 벡터 $ST_{j,X}$, $ST_{j,Y}$, $ST_{j,Z}$를 생성한다. (k = 0, 1, … , N-1)

$$X_{j,k}=\sqrt{\frac{2}{N}}\,C_k\sum_{n=0}^{N-1}x_{j,n}\cos\frac{(2n+1)k\pi}{2N} \quad (1)$$

여기서 C_k 값은 k = 0일 때 $1/\sqrt{2}$ 을 갖고, k = 1, 2, … , N-1일 때는 1의 값을 갖는다.

Hartung의 방식[5]을 이용하여 m 비트 길이의 추가 정보 $\mathbf{a}=(a_1,a_2,\cdots,a_m)$, $a_i\in\{0,1\}$를 벡터 $ST_{j,X}$의 i번째 DCT 계수 $X_{j,i}$에 식 (2)와 같이 삽입하였다.

$$X'_{j,i} = X_{j,i} + b'_i \cdot p_i \cdot \alpha \quad (2)$$

여기서 $b'_i \in \{-1,1\}$는 비트 a_i를 chip rate c만큼 복제하여 그 값이 0일 경우에는 -1로, 1일 경우에는 1로 설정한 값이다. 또한 DCT 계수를 변조하기 위해서 stego key k_2로 생성되는 PRNS인 $p_i \in \{-1,1\}$와 변조크기 변수 α(α>0)가 필요하다. 벡터 $ST_{j,Y}$, $ST_{j,Z}$의 DCT 계수들도 동일한 방법으로 변조하면 추가 정보가 삽입된 DCT 계수 벡터 $ST_{j,X'}$, $ST_{j,Y'}$, $ST_{j,Z'}$가 된다.

$ST_{j,X'}$, $ST_{j,Y'}$, $ST_{j,Z'}$로부터 식 (3)과 같이 DCT 역변환을 취하면, 추가 정보가 삽입이 된 공간 영역에서의 벡터 $ST_{j,x'}$, $ST_{j,y'}$, $ST_{j,z'}$을 구할 수 있다. (n = 0, 1, … , N-1)

$$x'_{j,n} = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} C_k \, X'_{j,k} \cos\frac{(2n+1)k\pi}{2N} \quad (3)$$

동일한 방법으로 다른 삼각형 스트립에도 적용하여 최종적으로 i번째 분할 stego-메쉬 M'_i를 만들 수 있다.

메쉬 합성(mesh synthesis) 단계에서는 하나의 stego-메쉬를 만들기 위해서는 각각의 분할 메쉬가 이웃한 영역에서 서로 공유하는 꼭지점에 대하여 추가 정보의 왜곡을 최소화하기 위한 방법이 고려되어야 한다. 분할된 stego-메쉬들이 서로 이웃한 경우에는 공유 꼭지점 좌표값으로 두 메쉬의 공유 꼭지점의 좌표값의 평균치를 사용하고, 분할 cover-메쉬와 분할 stego-메쉬가 서로 이웃하는 경우에는 공유 꼭지점의 좌표값으로 stego-메쉬의 공유 꼭지점 좌표값을 사용한다.

3.2 핑거프린트 추출 알고리즘

그림 2는 핑거프린트 검출 방법을 보여준다. 입력은 핑거프린트가 삽입되어 있는 stego-메쉬나 외부로부터 공격받은 stego-메쉬, cover-메쉬 및 핑거프린트 삽입 과정에서 사용되었던 입력변수들이 필요하다.

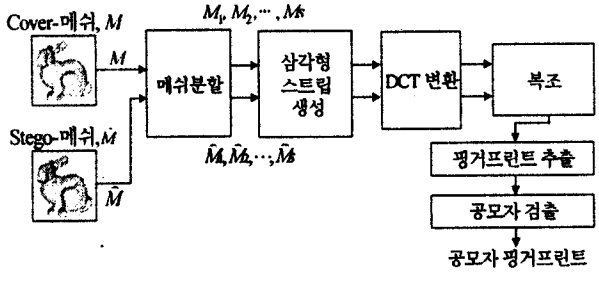


그림 2. 핑거프린트 검출 알고리즘

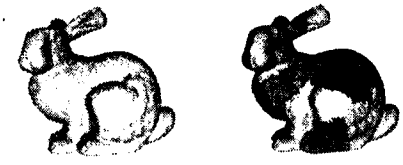
우선 cover-메쉬와 stego-메쉬를 핑거프린트 삽입과 동일한 방법으로 두 메쉬를 분할하여 각각의 분할 메쉬에 대하여 삼각형 스트립을 생성한다. 여기에 DCT 변환을 취하여 생성된 DCT 계수 벡터가 각각 $ST_{j,X}$, $ST_{j,Y}$, $ST_{j,Z}$와 $ST_{j,X'}$, $ST_{j,Y'}$, $ST_{j,Z'}$라고 할 때, stego-메쉬의 i번째 분할 메쉬의 추가 정보 삽입유무를 알아보기 위해 우선 핑거프린트 생성과정에서 사용했던 동일한 stego-key k_2를 사용하여 PRNS $p_i \in \{-1,1\}$를 생성한다. 이 PRNS값을 $ST_{j,X}$, $ST_{j,Y}$, $ST_{j,Z}$와 $ST_{j,X'}$, $ST_{j,Y'}$, $ST_{j,Z'}$의 차에 곱한 뒤에 식 (4)와 같이 계산하여 q_l를 구한다.

$$q_l = \frac{1}{3}\left(\sum_{i=k_c}^{(l+1)\times c-1} \{(X'_{j,i} - X_{j,i}) + (Y'_{j,i} - Y_{j,i}) + (Z'_{j,i} - Z_{j,i})\}\right) \quad (4)$$

q_l 값이 양수일 때 삽입된 추가 정보의 l번째 비트 a_l를 1로 설정하고, 이 값이 음수일 때 0으로 설정한다. 이와 같은 방법을 모든 분할 메쉬에 적용하여 추가 정보의 나머지 비트들을 구한다. 이렇게 추출된 추가 정보 $\mathbf{a}'$를 $\mathbf{a}$와 비교하여 서로 동일한 경우 i번째 분할 stego-메쉬와 일대일로 대응하는 핑거프린트의 비트를 1로 설정하고, 틀린 경우에는 0으로 설정한다. 이와 같은 알고리즘을 모든 분할 메쉬에 적용하여 stego-메쉬에 존재하는 $k \times r$ 비트를 추출한다. 추출한 $k \times r$ 비트를 stego-key k_1을 이용하여 재배열한 후 잉여변수 r만큼 반복되는 각각의 반복 비트들이 1이 많은 경우에는 1을, 0이 많은 경우에는 0의 값을 k개의 해당 비트에 각각 설정하여 k비트의 핑거프린트 FP_{stego}를 생성한다. FP_{stego}와 핑거프린트를 생성할 때 만들어진 q+1개의 핑거프린트 정보를 이용하여 II절에서 언급한 Dittmann의 공모자 검출 알고리즘을 적용하여 공모에 가담한 최대 d만큼의 공모자를 찾을 수 있다.

IV. 실험 결과

본 논문에서 제안한 공모방지 핑거프린팅 기법의 성능을 평가하기 위해 삼차원 Bunny 모델을 15095 꼭지점과 30019 면을 갖도록 그림 3과 같이 단순화한 모델을 이용하여 컴퓨터 시뮬레이션을 수행하였다.



(a) 실험 모델 (b) 메쉬 분할

그림 3. 삼차원 Bunny 모델과 메쉬 분할

본 논문의 실험을 위해서 PG(2,2)로 생성된 세 개의 핑거프린트 FP_1=110100, FP_2=0011010, FP_3=100110를 사용하였으며, 세 개의 핑거프린트를 실험 모델에 삽입하기 위해 사용된 변수 값들은 v=15095, k=7, r=3, s=30, m=35, c=3, α=0.0002이다.

본 논문에서 제안한 알고리즘의 비지각성을 평가하기 위하여 cover-메쉬와 stego-메쉬의 차이를 측정할 수 있는 MESH[6]를 사용하였다. 본 논문에서는 표본화 간격으로 0.5%를 선택하여 cover-메쉬와 공격받지 않은 stego-메쉬와 공격받은 stego-메쉬 사이의 평균 제곱근

오류 d_{rmse}을 계산하였다. 그림 4는 실험 모델에서 α값에 따른 cover-분할 메쉬와 stego-분할 메쉬 사이의 d_{rmse}을 보여준다.

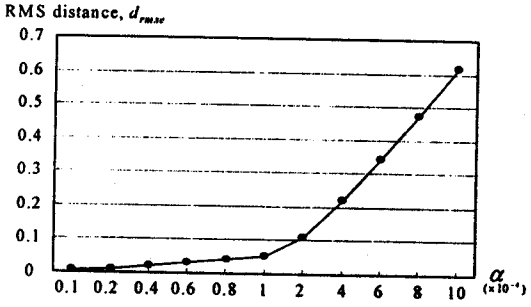


그림 4. α 값에 따른 d_{rmse} 값의 변화

Bunny 모델에서 추가 정보를 삽입한 후에 분할 cover-메쉬와 분할 stego-메쉬 사이의 차이를 시각적으로 식별하기 어렵게 하기 위해서는 d_{rmse}값을 0.2 이하로 선택하는 것이 바람직하였으며, 본 논문에서도 비지각성과 외부공격에 대한 강인성을 고려하여 α값을 0.2로 선택하였다.

(a) 잡음 (b) 압축 (c) 기하학 변환 (d) 공모

그림 5. Bunny 모델에 대한 다양한 공격

그림 5에 보인 것처럼, 제안된 핑거프린팅 기법에 대한 공격 실험으로 삼차원 메쉬 모델의 기하학 정보에 대한 잡음 공격, MPEG-4 SNHC 표준을 이용한 삼차원 메쉬 모델의 기하학 정보 압축, 기하학 변환 및 두 사람 공모자에 의한 두 개의 메쉬 모델의 꼭지점 좌표값의 평균값을 취하는 공모 공격을 적용하였다.

표 1는 Bunny 모델에 대한 그림 5에서 제시한 공격으로 실험한 결과를 보여주고 있다. 잡음 공격에서의 수치는 Bunny 모델의 바운딩 박스의 x, y, z 축의 변위값 중 최대값과 모델의 기하학 정보에 더해질 잡음의 최대값의 비율이다. 비트 오류율(BER, bit error rate)은 추가 정보가 삽입된 모든 분할 stego-메쉬에서 발생하는 비트 오류를 삽입한 추가 정보의 총 비트수로 나눈 값이며, 검출한 공모자 수가 t일 때, 공모자 검출율(TDR, traitor detection rate)은 $(t/d)\times 100$으로 계산한다. 실험 결과로부터 제안한 핑거프린팅 기법이 잡음 공격, 압축 공격, 기하학 변환 및 두 사람에 의한 메쉬 평균 공격에 대하여 강인함을 알 수 있다.

V. 결 론

본 논문에서는 삼차원 메쉬 모델에 적용할 수 있는 새로운 핑거프린팅 기법을 제안하였다. 실험을 통해 제안된 핑거프린팅 기법이 잡음 공격, 압축 공격, 기하학 변환 및 공모를 통한 핑거프린트 공격에도 강인함을 확인하였다. 본 논문에서 제안한 방법은 불법적으로 유통되는 삼차원 메쉬 모델에서 삽입된 추가 정보를 바탕으로 핑거프린트를 추출하여 삼차원 메쉬 모델의 불법 유통과 관련된 공모자를 검출하는데 사용될 것이다.

표 1. Bunny 모델의 실험결과

공격		BER	d_{rmse}
없음		0	0.0140667
잡음(%)	0.2	0	0.0357517
	0.4	0	0.0653549
	0.6	0.031746	0.0988771
	0.8	0.063492	0.131353
	1.0	0.2063492	0.163298
압축(%)	80	0	0.021482
	85	0	0.0349104
	90	0	0.0658695
기하학 변환	1.5배 확대	0	11.4429
	평행이동	0	324.318
	Shearing	0	11.5655
공모	메쉬평균	TDR(%)	d_{rmse}
		100	0.0110824

감사의 글

본 연구는 광주과학기술원 (K-JIST) 초고속광네트워크 연구센터 (UFON)를 통한 한국과학재단 우수연구센터 (ERC)와 교육부 두뇌한국21 (BK21) 정보기술사업단의 지원에 의한 것입니다.

참 고 문 헌

[1] D. Boneh and J. Shaw, "Collusion-secure Fingerprinting for Digital Data," *IEEE Trans. Info. Theory*, vol. 44, no. 5, pp. 1897-1905, Sept. 1998.

[2] J. Dittmann, "Combining Digital Watermarks and Collusion Secure Fingerprints for Customer Copy Monitoring," *Proc. IEE Seminar Sec. Image & Image Auth.*, pp. 9-17, 2000.

[3] G. Karypis and V. Kumar, MeTis Version 4.0, Univ. of Minnesota, Dept. of Comp. Sci., 1998

[4] P. Terdiman, "Creating Efficient Triangle Strips," http://codercorner.com/Strips.htm.

[5] F. Hartung, P. Eisert and B. Girod, "Digital Watermarking of MPEG-4 Facial Animation Parameters," *Computer and Graphics*, vol. 22, no. 4, pp. 425-435, Elsevier, 1998.

[6] N. Aspert, D. Santa-Cruz and T. Ebrahimi, "MESH: Measuring Error between Surfaces using the Hausdorff distance," http://mesh.epfl.ch.