

논문 2004-41C1-4-13

삼차원 메쉬 모델에 적용한 공모방지 핑거프린팅 기법

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

허 영*, 전 정 희**, 호 요 성***

(Yung Hur, Jeong-Hee Jeon, and Yo-Sung Ho)

요 약

본 논문에서는 공모방지 핑거프린팅 알고리즘에 의하여 생성된 핑거프린트를 삼차원 메쉬 모델에 효율적으로 삽입할 수 있는 새로운 공모방지 핑거프린팅 기법을 제안한다. 제안한 알고리즘은 유한 사영기하학(finite projective geometry)을 기반으로 고객의 수만큼 핑거프린트를 만들고 이 정보를 바탕으로 삼차원 메쉬 모델을 분할한 다음, 마크(mark)할 특정 분할메쉬(submesh)에 저작권을 나타내는 워터마크 신호를 은닉한다. 삽입할 워터마크 신호는 비인지성과 강인성을 고려하여 마크할 분할메쉬로부터 삼각형 스트립(triangle strips)을 생성하고 각 스트립에 포함된 꼭지점 값들을 DCT 영역의 계수 값들로 변환시킨 후 중간 주파수 대역에 삽입한다. 다양한 실험을 통해 제안한 기법이 무작위 잡음첨가, MPEG-4 SNHC의 삼차원 메쉬 꼭지점 좌표값 압축, 기하학 변환 및 공모에 의한 핑거프린트 공격에 대해 강인할 뿐만 아니라 생성된 핑거프린트의 비트 수를 기존의 방법보다 줄일 수 있었다.

Abstract

This paper presents a new collusion-secure fingerprinting scheme to embed fingerprints into three-dimensional(3-D) mesh models efficiently. In the proposed scheme, we make the same number of fingerprints as the number of customers based on the finite projective geometry, partition a 3-D mesh model related to the number of bits assigned to each fingerprint, and then embed a watermark representing copyright information into each submesh to be marked. Considering imperceptibility and robustness of the watermarking algorithm, we embed the watermark signal into mid-frequency DCT coefficients obtained by transforming vertex coordinates in the triangle strips which are generated from the submeshes to be marked. Experimental results show that our scheme is robust to additive random noises, MPEG-4 SNHC 3-D mesh coding, geometrical transformations, and fingerprint attacks by two traitors' collusion. In addition, we can reduce the number of bits assigned to each fingerprint significantly.

Keywords: 핑거프린팅, 디지털 워터마킹, 유한 사영기하학, 삼차원 메쉬 모델

I. 서 론

최근 인터넷 기술의 발달로 인해 오디오, 정지영상, 동영상, 삼차원 데이터 등과 같은 다양한 디지털 데이터를 여러 사람이 쉽고 빠르게 송수신을 할 수 있게 되

었다. 그러나 디지털 데이터는 원본과 거의 동일한 복사본이 불법적으로 급격하게 배포될 수가 있다는 문제점을 안고 있으며, 이런 사실은 저작권 보호 측면에서 심각한 위협이 되고 있다. 따라서 이런 불법적인 복제 및 유통을 억제할 수 있는 기술로 디지털 워터마킹(digital watermarking)이 대두되었으며, 1990년에 정지영상에 적용된 워터마킹 기술이 처음 Tanaka^[1]에 의하여 소개된 이후, 1990년 중반부터 급격하게 많은 연구가 이루어졌다.

디지털 워터마킹은 디지털 데이터에 소유권을 주장할 수 있는 정보를 삽입하여 배포한 후, 불법 복제된 디지털 데이터로부터 워터마크를 추출함으로써 소유권을

* 정회원, 국가보안기술연구소
(National Security Research Institute)

** 학생회원, (주)지오소프트
(Geosoft)

*** 종신회원, 광주과학기술원 정보통신공학과
(Dept. of Information and Communications,
Gwangju Institute of Science and Technology)
접수일자: 2004년2월10일, 수정완료일: 2004년6월14일

주장할 수 있는 기술이며 저작권 보호, 방송 모니터링, 인증, 복사 방지(copy protection), 불법 배포자 추적을 위한 핑거프린팅(fingerprinting) 등에 사용될 수 있다. 이 중 핑거프린팅은 사용자 식별정보인 핑거프린트(fingerprint)를 디지털 데이터에 은닉하여 불법적인 디지털 데이터의 복사본 배포에 가담한 사용자를 추적하는 기술이다. 핑거프린트가 삽입된 디지털 데이터는 서로 조금씩 상이하게 되어 몇몇의 사용자들이 서로 공모하여 소유하고 있는 핑거프린트가 삽입된 복사본을 비교, 분석함으로써 핑거프린트 정보를 삭제하거나 변경시킬 수 있는 공모공격(collusion attack)이 가능하다.

이러한 공모로 발생하는 문제점은 Blakley^[2]에 의해 최초로 제기되었다. Boneh와 Shaw는 공모 문제가 핑거프린팅 기법에서 가장 중요한 문제임을 지적하면서 공모자의 수가 많은 경우에 대한 해결책으로 c-방지 코드(c-secure codes)를 이용한 공모방지 핑거프린팅 기법을 제안했다^[3]. 그 후, 비대칭 핑거프린팅^[4]과 익명 핑거프린팅^[5]이 기존의 핑거프린팅 기법의 기능을 향상시키기 위하여 연구되었다.

한편, 정지영상에 적용한 공모방지 핑거프린팅 기법은 Dittmann에 의하여 처음 시도되었다. Dittmann은 유한 사영기하학(finite projective geometry)을 기반으로 생성된 핑거프린트가 삽입된 정지영상을 소유한 몇몇 사용자가 공모를 통해 삽입한 정보의 변경 및 삭제를 방지할 수 있는 공모방지 핑거프린팅 알고리즘을 개발하였으며, 이런 공모방지 핑거프린팅 알고리즘과 효율적으로 결합된 강인한(robust) 워터마킹 알고리즘을 제안했다^[6]. 또한 Domingo-Ferrer는 정지영상에 대해서 공모자의 수가 두 명일 경우에 대하여 이중이진 해밍코드(dual binary Hamming code)를 사용해 공모방지 핑거프린팅 알고리즘과 워터마킹 알고리즘을 결합한 방식을 제안했다^[7]. 최근에는 마이크로소프트사의 미디어 플레이어 플랫폼에 기존의 워터마킹 시스템의 보안성을 보완하고 여기에 핑거프린팅 기능까지 부여한 듀얼 워터마킹/핑거프린팅(dual WM/FP) 시스템을 적용한 기법이 제안되었다^[8].

하지만 기존의 디지털 데이터에 적용되는 핑거프린팅 기법은 최근 CAD(computer aided design)와 VR(virtual reality) 등에서 널리 이용되고 있는 삼차원 메쉬 모델에는 직접적으로 적용할 수 없다. 따라서 우리는 공모방지 핑거프린팅 알고리즘과 삼차원 메쉬 모델에 적용할 수 있는 워터마킹 알고리즘을 효과적으로 결합하여 공모방지 핑거프린트를 삼차원 메쉬 모델에

삽입할 수 있는 공모방지 핑거프린팅 기법을 제안한다.

본 논문의 II장에서는 삼차원 메쉬 모델에 적용할 수 있는 공모방지 핑거프린팅과 관련된 용어, 요구사항, 공모방지 핑거프린팅 기법의 구성 요소 및 본 논문에서 중요하게 적용되는 Dittmann이 제안한 공모방지 핑거프린팅 알고리즘에 대해서 설명한다. III장과 IV장에서는 공모방지 핑거프린트를 생성하여 삼차원 메쉬 모델에 삽입하고 추출하는 알고리즘을 제안한다. V장에서는 삼차원 메쉬 모델에 삽입된 핑거프린트에 대한 강인성 실험 결과와 이 결과를 토대로 삼차원 메쉬 모델에 적용한 공모방지 핑거프린팅 기법에 대한 성능을 평가한 후에 VI장에서 결론을 맺는다.

II. 공모방지 핑거프린팅

1. 용어 및 요구사항

우선 본 논문에서 사용되는 공모방지 핑거프린팅과 관련된 용어는 다음과 같다^[3,9]. 마크(mark)는 디지털 데이터의 일부분으로써 여러 상태에 있을 수 있으며, 핑거프린트는 이러한 마크의 집합체이다. 배급업자(distributor)는 핑거프린트가 삽입되어 있는 디지털 데이터를 사용자에게 제공할 수 있는 권한이 있는 공급자이다. 사용자(user)는 핑거프린트가 삽입되어 있는 디지털 데이터를 사용할 수 있는 권한이 있는 고객으로서 배급업자에게 데이터에 대한 사용 허용자로서 등록되어 있다. 공격자(attacker)는 핑거프린트가 삽입되어 있는 디지털 데이터에 불법적으로 접근한 사용 권한이 없는 사람이며, 공모자(traitor)는 핑거프린트가 삽입되어 있는 디지털 데이터를 불법적으로 배포한 사용자이다. 또한, 핑거프린트를 삽입하기 전의 원 모델을 커버 메쉬(cover-mesh), 핑거프린트를 삽입한 후의 모델을 스테고 메쉬(stego-mesh)로 정의하였다. 외부로부터 공격받은 핑거프린트가 삽입된 메쉬도 스테고 메쉬에 해당한다. 핑거프린팅은 일반적으로 강인한 워터마킹 기법에서 요구되는 비인지성(imperceptibility)과 강인성(robustness) 이외에 공모를 통한 공격에 강인할 수 있는 공모방지(collusion secure) 성질을 추가로 요구한다. 따라서 스테고 메쉬에는 이러한 성질을 모두 내포하도록 핑거프린트가 삽입되어야만 한다.

2. 구성 요소

삼차원 메쉬 모델에 적용하는 공모방지 핑거프린팅 기법은 핑거프린트 생성, 마크 위치 선정, 워터마킹, 공

모자 추적 알고리즘으로 이루어진다. 핑거프린트 생성 알고리즘은 사용자에게 제공하는 삼차원 메쉬 모델에 삽입할 사용자의 고유한 식별정보인 핑거프린트를 생성한다. 마크 위치 선정 알고리즘은 삼차원 메쉬 모델에 핑거프린트를 삽입할 위치를 정할 때, 커버 메쉬를 핑거프린트의 비트 수에 따라 분할한 후, 각각의 비트 정보에 의하여 특정한 분할메쉬를 선택하는데 사용된다. 즉, 삼차원 메쉬 모델의 마크 위치(marking position)를 선택하는 역할을 마크 위치 선정 알고리즘이 수행하게 된다. 워터마킹 알고리즘은 선택된 특정 분할메쉬를 마크하기 위하여 저작권 정보인 워터마크 신호를 삽입하는데 사용된다. 마지막으로 공모자 추적 알고리즘은 외부로부터 공격받은 핑거프린트가 삽입된 스테고 메쉬에서 핑거프린트를 추출하여 공격에 기여한 사용자 중 적어도 한 사람을 밝혀내는데 쓰인다.

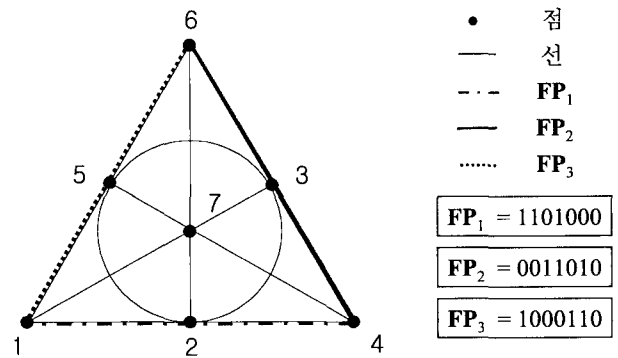


그림 1. 2-detecting 핑거프린팅 기법
Fig. 1. 2-detecting fingerprinting scheme.

3. Dittmann의 공모방지 핑거프린팅 알고리즘

본 논문에서는 핑거프린트 생성 알고리즘으로 Dittmann이 제안한 방법을 응용하였다. Dittmann은 최대 공모자의 수가 d 인 경우에 대해서 모든 공모자들을 검출할 수 있는 유한 사영공간(finite projective space) $PG(d, q)$ 을 기반으로 d -detecting RNC(rational normal curves) 기법을 제안하였다^[6].

$PG(d, q)$ 는 유한체(finite field) $GF(q)$ ($q=p^a$, p 는 소수, $a \geq 1$) 상의 d 차원 사영공간이다. 이때 RNC R 은 식 (1)과 같이 초평면(hyperplane)의 집합체이다.

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

식 (1)을 기반으로 핑거프린트를 생성하는 방법은 다음과 같다. 첫째, 사용자 i 에 대해서, $i \in GF(q)$ 와 $H_i = [1:i^2:\dots:i^d]$ ($H_i \in R$)을 선택하고 둘째, 방정식 $1 \cdot x_0 + i \cdot x_1 + \dots + i^d \cdot x_d = 0$ 을 만족시키는 $PG(d, q)$ 상의 점 $(a_1, a_2, \dots, a_i, \dots, a_d)$, $a_i \in GF(q)$ 들과 연계되어 있는 모든 마크들을 선택하여 이 마크들과 대응되는 비트를 1로 설정하여 핑거프린트를 생성한다.

그림 1은 유한 사영공간 $PG(2, 2)$ 에서 최대 2명의 공모자를 검출할 수 있는 2-detecting 핑거프린팅 기법을 보여준다. $PG(2, 2)$ 는 사영 공간 중 가장 작은 형태이며, 7개의 점과 7개의 선으로 이루어져 있다. $PG(2, 2)$ 상의 7개의 점은 각각 $(0,0,1)$, $(0,1,0)$, $(1,1,0)$, $(0,1,1)$, $(1,0,0)$, $(1,0,1)$, $(1,1,1)$ 이며 이 점들에 대한 인덱스는 그림 2에서처럼 각각 1, 2, 3, 4, 5, 6, 7로 표현하였다. 우선 FP_1 의 생성만을 고려해 보자. i 의 값이 0일 때 초평

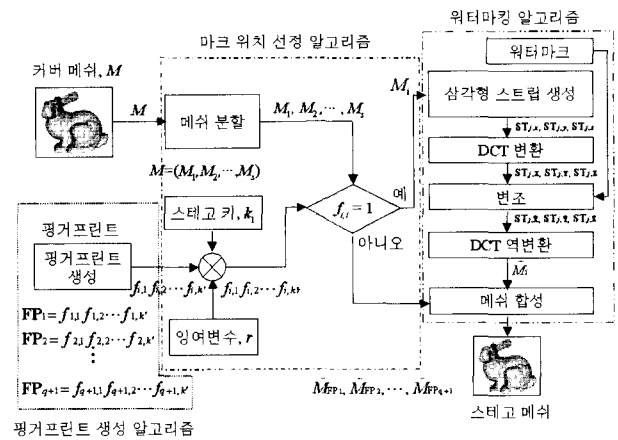


그림 2. 핑거프린트 삽입 과정
Fig. 2. Procedure for embedding collusion-secure fingerprints.

면 $H_0 = [1:0:0]$ 이 선택되어 $PG(2, 2)$ 상의 점들 중 $1 \cdot x_0 + 0 \cdot x_1 + 0 \cdot x_2 = 0$ 을 만족시키는 세 점 $(0,0,1)$, $(0,1,0)$, $(0,1,1)$ 이 구해진다. 이 점들의 인덱스 번호의 순번에 해당되는 비트 값을 1로 설정하고 나머지 비트를 0으로 설정하면 $FP_1 = 1101000$ 이 생성된다. i 에 대해서도 동일한 방법을 적용시키면 $FP_2 = 0011010$ 이 생성된다.

그림 1에서처럼 파란 선위의 점을 선택하여 핑거프린트를 생성하면 $FP_3 = 1000110$ 이 된다. $PG(2, 2)$ 에서 생성된 세 개의 핑거프린트들 중 어느 두개의 핑거프린트를 비교해도 서로 공유하는 마크 위치가 존재한다. 이 마크 위치가 공모를 한 경우에도 살아남은 정보가 되어 공모자 검출 알고리즘을 통하여 공모자를 색출할 수 있다.

III. 핑거프린트 삽입 과정

그림 2는 삼차원 공모방지 핑거프린팅 기법의 구성 요소 중 핑거프린트 생성 알고리즘, 마크 위치 선정 알

고리즘 및 워터마킹 알고리즘을 순차적으로 나타낸 전체적인 핑거프린트 삽입 과정을 보여준다. 먼저 공모방지 핑거프린트를 생성하여 그 정보에 따라 삼차원 메시지를 분할하고 워터마크 신호를 삽입할 분할메쉬를 결정한다. 이렇게 마크할 분할메쉬가 결정되면 그 메쉬에 대한 삼각형 스트립을 생성하여 스트립에 속한 좌표값들을 DCT 영역으로 변환시켜 워터마크를 삽입하여 스테고 분할메쉬를 생성한다. 한편 선택되지 않는 커버 분할메쉬는 워터마크를 삽입하지 않고 메쉬 합성 단계에서 스테고 분할메쉬와 결합하여 하나의 스테고 메쉬를 만들게 된다. 따라서 핑거프린트 정보에 따라 마크할 각각의 분할메쉬에 워터마크를 삽입함으로써 결국은 삼차원 메시 모델에 핑거프린트를 삽입하게 된다.

1. 핑거프린트 생성 알고리즘

공모방지 핑거프린트를 생성하기 위하여 본 논문에서는 Dittmann이 제안한 방법을 수정하였다. 공모자의 수가 2명인 경우만을 고려한 공모방지 핑거프린트 생성 과정은 다음과 같다. 먼저, 입력 변수인 사용자 수 q 에 따라 $PG(2, q)$ 를 생성하여 Dittmann의 방식에 따라 $k=q^2+q+1$ 비트로 구성되는 $q+1$ 개의 핑거프린트를 생성한다. 생성된 모든 핑거프린트에서 같은 위치에 있는 각각의 비트를 서로 비교하여 비트값이 모두 0인 경우가 이 비트를 각각의 핑거프린트에서 제거한다. 이렇게 제거된 비트 수를 δ 라 할 때, 새롭게 생성된 핑거프린트의 비트 수 k' 는 $k-\delta$ 가 된다. 따라서 $PG(2, q)$ 공간에 2-detecting 핑거프린팅 기법을 적용하여 k' 비트로 이루어진 $q+1$ 개의 핑거프린트 $FP_1, FP_2, \dots, FP_i, \dots, FP_{q+1}$ 가 생성되며, 여기서 FP_i 는 i 번째 사용자에게 부여되는 핑거프린트를 의미한다. 그림 1에서 생성된 세 개의 핑거프린트는 각각 $FP_1=110100$, $FP_2=001101$, $FP_3=100011$ 가 된다.

2. 마크 위치 선정 알고리즘

우리는 이렇게 만들어진 핑거프린트를 외부 공격으로부터 강인하게 만들기 위해 홀수 잉여변수 r 만큼 반복하여 커버 메쉬에 삽입한다. 따라서 핑거프린트 전체 비트 수는 $k \times r$ 이다. 잉여변수 r 은 커버 메쉬의 꼭지점 개수 v 와 추후 분할메쉬에 삽입할 워터마크 정보의 비트 수 m , 반복회수(chip rate) c 에 따라 $m \times c$ 만큼 DCT 영역에서 중간 주파수 대역만을 선택하여 삽입한다. 이때 잉여변수 r 은 식 (2)와 같이 최대치 r_{max} 까지 설정할 수 있다.

$$r_{max} = \left\lfloor \frac{v}{k' \times 2m \cdot c} \right\rfloor \quad (2)$$

여기서 $\lfloor \cdot \rfloor$ 은 정수만을 취하는 연산자다. 그러므로 FP_i 는 잉여변수 r 에 의하여 $k \times r$ 비트의 핑거프린트가 생성된다. 마찬가지로 다른 모든 핑거프린트에 대해서도 동일하게 적용된다. 커버 메쉬에 공모방지 핑거프린팅 기법을 효율적으로 적용하기 위해 MeTis^[16]라는 메쉬 분할 소프트웨어를 이용해 삼차원 메시지를 분할한다. 분할메쉬의 개수 s 는 식 (3)과 같이 최대치 s_{max} 까지 설정할 수 있다.

$$s_{max} = \left\lfloor \frac{v}{2m \cdot c} \right\rfloor \quad (3)$$

여기서 분할되는 메쉬의 개수 s 는 항상 $s \geq k \times r$ 을 만족한다.

스테고 키(stego-key) k_1 으로 생성되는 PRNS (pseudo random number sequence)에 따라 총 s 개의 분할메쉬 중 $k \times r$ 개의 분할메쉬를 선택한 후, FP_i 의 $k \times r$ 비트와 선택된 분할메쉬를 서로 일대일 대응시킨다. 이때 k_1 은 공모 공격에 의하여 각각의 분할메쉬와 핑거프린트 비트들간의 대응관계를 알지 못하도록 해 준다. 따라서 공모 공격으로 알 수 있는 것은 스테고 메쉬들 사이의 차이를 발생하게 하는 꼭지점과 면의 위치일 뿐이다. 핑거프린트 FP_i 의 비트 중에 0일 때는 무시하고, 1의 값을 갖는 경우에만 그 비트와 대응하는 분할메쉬 M_i 를 선택하여 워터마킹 알고리즘을 적용하여 워터마크를 삽입함으로써 마크하게 된다.

3. 워터마킹 알고리즘

본 논문에서 [10]에서 제안한 방법을 토대로 워터마크를 분할메쉬에 삽입하였다.

가. 삼각형 스트립 생성

분할메쉬로부터 삼각형 스트립을 생성하기 위하여 P. Terdiman에 의해 제안된 방법^[11]을 이용하였다. 이 방법으로 VRML(virtual reality modeling language) 파일의 연결성 정보와 꼭지점 좌표들로부터 삼차원 메시 모델을 표현하고, 이 모델로부터 삼각형 스트립을 생성한다. 본 논문에서는 FP_i 의 i 번째 비트가 1인 비트와 대응된 분할메쉬 M_i 가 n 개의 꼭지점을 가지고 있을 때 여러 개의 삼각형 스트립이 생성이 되고 이 스트립에 포함되어 있는 꼭지점의 기하학 정보를 바탕으로 DCT

변환을 한다.

나. DCT 변환

표현의 단순화를 위하여 j 번째 삼각형 스트립 ST_j 에 포함되어 있는 N 개의 꼭지점의 좌표값 $v_i = (x_i, y_i, z_i)$, ($1 \leq i \leq N$)에 대해서 DCT 변환을 한 후에 워터마크를 삽입하는 경우에 대하여 먼저 고려하자.

원래 삼각형 스트립을 이용해 DCT 변환을 하는 방식은 삼차원 메쉬 압축 분야에서 먼저 제안되었으며^[12], 본 논문에서는 스트립을 적절한 크기의 블록(block)으로 분할한 후, DCT 변환을 위해서 블록의 크기(block size)를 패딩(padding)을 통해 조절한 다음 DCT 변환을 하는 방식^[10]을 이용하였다.

DCT 변환을 위하여 우선 N 개의 꼭지점의 좌표값들을 x 좌표값, y 좌표값, z 좌표값끼리 각각 분리하여 벡터 $ST_{j,x}$, $ST_{j,y}$, $ST_{j,z}$ 를 생성하여 식 (4)와 같이 DCT 변환을 이용하여 각각의 좌표축의 꼭지점 값들로부터 계산된 DCT 계수들로 이루어진 DCT 계수벡터 $ST_{j,x}$, $ST_{j,y}$, $ST_{j,z}$ 를 생성한다.

$$\begin{aligned} X_{j,k} &= \sqrt{\frac{2}{N}} C_k \sum_{n=0}^{N-1} x_{j,n} \cos \frac{(2n+1)k\pi}{2N}, \quad k=0,1,\Lambda, N-1 \\ Y_{j,k} &= \sqrt{\frac{2}{N}} C_k \sum_{n=0}^{N-1} y_{j,n} \cos \frac{(2n+1)k\pi}{2N}, \quad k=0,1,\Lambda, N-1 \\ Z_{j,k} &= \sqrt{\frac{2}{N}} C_k \sum_{n=0}^{N-1} z_{j,n} \cos \frac{(2n+1)k\pi}{2N}, \quad k=0,1,\Lambda, N-1 \\ C_k &= \begin{cases} 1/\sqrt{2}, & k=0 \\ 1, & o/w \end{cases} \end{aligned} \quad (4)$$

워터마킹에서는 일반적으로 공간영역보다 변환된 영역에서 워터마크 신호를 삽입하고 추출하는데, 그 주된 이유는 변환된 영역에서 비교적 용이하게 워터마크 신호의 강인성과 비인지성을 고려할 수 있기 때문이다. 따라서 삼차원 메쉬 모델의 압축 등과 같은 공격에 강인하면서 삼차원 메쉬 모델의 외형적인 변화를 최소화 하기 위해 모든 AC 계수를 4등분하여 저주파 대역과 고주파 대역에 해당하는 AC 계수를 제외한 전체 주파수 대역의 1/2에 해당하는 중간 주파수 대역의 AC 계수에 워터마크를 삽입했다. 즉, 벡터 $ST_{j,x}$, $ST_{j,y}$, $ST_{j,z}$ 의 성분 중 중간 주파수 대역의 성분만을 선택하여 변조과정을 통해 워터마크를 삽입한다.

다. 변조

분할메쉬에 워터마크 신호를 삽입 방법은 기본적으로 Hartung의 방식과 유사하다^[13]. 먼저, d -detecting 핑거프린팅 방법에 의해 생성된 핑거프린트 신호를 얻어 변수 r 만큼 반복시켜 만들어진 신호 FP_r 비트 중에 0일 경우에는 무시하고 1인 경우에만 해당되는 분할메쉬 M_i 를 선택한다. 이 마크할 분할메쉬에 m 비트 길이의 워터마크 정보 $a=(a_1, a_2, \dots, a_i, \dots, a_m)$, $a_i \in \{0,1\}$ 를 삽입하게 되는데, 우선 워터마크 a 는 식 (5)와 같이 반복회수 c 만큼 신호를 확산시킴으로써 벡터 $b=(b_1, b_2, \dots, b_i, \dots, b_{mc})$, $b_i \in \{0,1\}$ 로 변환된다.

$$b_i = a_j, \quad j \times c \leq i < (j+1) \times c \quad (5)$$

벡터 b 는 다시 식 (6)에 의하여 벡터 $b'=(b'_1, b'_2, \dots, b'_i, \dots, b'_{mc})$, $b'_i \in \{0,1\}$ 로 변환된다.

$$b'_i = \begin{cases} -1, & \text{if } b_i=0 \\ 1, & \text{if } b_i=1 \end{cases} \quad (6)$$

따라서 DCT 계수 벡터 $ST_{j,x}$, $ST_{j,y}$, $ST_{j,z}$ 중 벡터 $ST_{j,x}$ 의 i 번째 DCT 계수 $X_{j,i}$ 의 변조는 식 (7)과 같이 수행된다.

$$\hat{X}_{j,i} = X_{j,i} + b'_i p_i \alpha \quad (7)$$

여기에서 p_i 는 스테고 키 k_2 에 의해 생성되는 PRNS 이고 α 는 변조강도(modulation amplitude)이다. 그러므로 삽입되는 워터마크 신호의 강도는 α 에 의해 조절될 수 있다. 마찬가지로 나머지 벡터 $ST_{j,y}$, $ST_{j,z}$ 의 DCT 계수들도 동일한 방법으로 변조하면 워터마크가 첨가된 DCT 계수 벡터 $ST_{j,x}$, $ST_{j,y}$, $ST_{j,z}$ 을 만들 수 있다.

라. IDCT 변환

워터마크가 삽입된 DCT 계수 벡터 $ST_{j,x}$, $ST_{j,y}$, $ST_{j,z}$ 로부터 i 번째 스테고 분할메쉬 γ 를 얻기 위하여 식 (8)과 같이 DCT 역변환(Inverse DCT)을 하면, 워터마크가 삽입이 된 공간 영역의 벡터 $ST_{j,x}$, $ST_{j,y}$, $ST_{j,z}$ 을 구할 수 있다.

$$\begin{aligned} x_{j,k} &= \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} C_k \hat{X}_{j,n} \cos \frac{(2n+1)k\pi}{2N}, \quad n=0,1,\Lambda, N-1 \\ y_{j,k} &= \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} C_k \hat{Y}_{j,n} \cos \frac{(2n+1)k\pi}{2N}, \quad n=0,1,\Lambda, N-1 \\ z_{j,k} &= \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} C_k \hat{Z}_{j,n} \cos \frac{(2n+1)k\pi}{2N}, \quad n=0,1,\Lambda, N-1 \end{aligned} \quad (8)$$

이와 같은 방법을 다른 삼각형 스트립에도 적용하여 최종적으로 스테고 분할매쉬 M 를 생성한다.

마. 매쉬 합성

핑거프린트가 삽입된 하나의 스테고 매쉬를 만들기 위해서는 각각의 분할매쉬가 이웃한 영역에서 서로 공유하는 꼭지점에 대하여 스테고 분할매쉬에 삽입되어 있는 워터마크 정보의 왜곡을 최소화하기 위한 방법이 고려되어야 한다. 스테고 분할매쉬들이 서로 이웃하고 있는 경우에는 공유 꼭지점 좌표값으로 두 매쉬의 공유 꼭지점의 좌표값의 평균치를 사용하고, 커버 분할매쉬와 스테고 분할매쉬가 서로 이웃하고 있는 경우에는 공유 꼭지점의 좌표값을 스테고 분할매쉬의 공유 꼭지점의 좌표값으로 대체한다. 이렇게 하여 결과적으로 커버 매쉬에는 사용자 정보인 핑거프린트와 저작권 정보인 워터마크가 동시에 삽입된다.

IV. 핑거프린트 추출 과정

그림 3은 핑거프린트 추출 과정을 간략하게 표현한 것이다. 입력으로 커버 매쉬, 핑거프린트가 삽입된 스테고 매쉬나 외부로부터 공격받은 스테고 매쉬 및 핑거프린트 삽입 과정에서 사용되었던 입력 변수들이 필요하며 마크 위치 선정 알고리즘과 워터마킹 알고리즘에서는 기본적으로 앞서 설명한 핑거프린트 삽입 알고리즘의 일부분이 포함되어 있다.

1. 마크 위치 선정 알고리즘

커버 매쉬와 스테고 매쉬를 핑거프린트 삽입 과정에 포함되어 있는 매쉬 분할 과정과 동일한 조건으로 두 매쉬를 각각 분할한다.

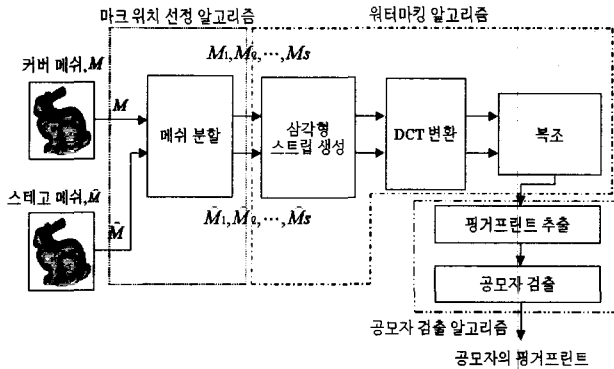


그림 3. 공모자 핑거프린트 추출 과정
Fig. 3. Procedure for extracting traitors' fingerprints.

2. 워터마킹 알고리즘

두 매쉬로부터 분할된 각각의 분할매쉬에 대하여 핑거프린트 삽입 과정과 동일한 조건으로 삼각형 스트립을 생성한 후 DCT 변환을 한다. 커버 매쉬와 스테고 매쉬의 i 번째 분할 매쉬로부터 생성된 j 번째 삼각형 스트립을 $ST_{j,i}, ST_{j,i}$ 라고 할 때 이 삼각형 스트립에서 DCT 변환으로 생성된 DCT 계수 벡터는 각각 $ST_{j,i,x}, ST_{j,i,y}, ST_{j,i,z}$ 와 $ST_{j,i,x}, ST_{j,i,y}, ST_{j,i,z}$ 이 만들어진다.

가. 복조

$ST_{j,i,x}, ST_{j,i,y}, ST_{j,i,z}$ 와 $ST_{j,i,x}, ST_{j,i,y}, ST_{j,i,z}$ 로부터 스테고 매쉬의 i 번째 분할 매쉬의 워터마크의 삽입 유무를 알아보기 위해 우선 핑거프린트 생성 과정에서 사용했던 동일한 스테고 키 k_2 를 사용하여 PRNS $p_i \in \{-1, 1\}$ 를 생성한다. 이 PRNS값을 $ST_{j,i,x}, ST_{j,i,y}, ST_{j,i,z}$ 와 $ST_{j,i,x}, ST_{j,i,y}, ST_{j,i,z}$ 의 차에 곱한 뒤에 식 (9)과 같이 계산하여 q_i 을 구한다.

$$q_i = \frac{1}{3} \left(\sum_{i=c}^{(H+1)-c-1} \{ (X_{j,i} - X_{j,i}) + (Y_{j,i} - Y_{j,i}) + (Z_{j,i} - Z_{j,i}) \} p_i \right) \quad (9)$$

q_i 로부터 식 (10)을 이용하여 삽입된 워터마크 정보의 i 번째 비트 a_i 를 구할 수 있으며 이와 동일한 방법을 모든 분할매쉬에 적용하여 워터마크 정보의 나머지 비트들을 구한다.

$$a_i = \begin{cases} 1, & \text{if } q_i > 0 \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

이렇게 추출된 워터마크 a' 는 핑거프린트의 검출에 사용된다.

3. 공모자 추적 알고리즘

가. 핑거프린트 추출

i 번째 스테고 분할매쉬로부터 구한 m 비트의 워터마크 정보 a' 와 핑거프린트를 삽입할 때 사용한 m 비트의 워터마크 정보 a 를 서로 비교하여 동일한 경우 i 번째 스테고 분할매쉬와 일대일 대응되는 핑거프린트의 비트를 1로 설정하고, 그렇지 않은 경우에는 0으로 설정한다. 이와 동일한 방법을 모든 분할 매쉬에 적용하여 스테고 매쉬에 존재하는 $k \times r$ 비트를 추출한다. 추출한 $k \times r$ 비트를 스테고 키 k_3 을 이용하여 재배열한 후 잉여변수 r 만큼 반복되는 각각의 반복 비트들이 1이 많은 경우는

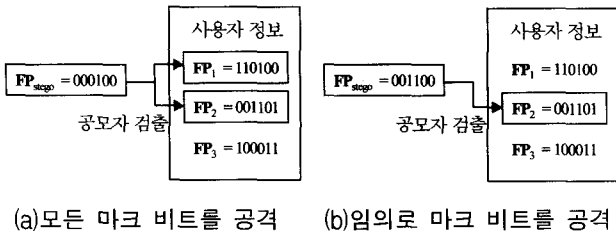


그림 4. 공모자 추적 과정
Fig. 4. Traitor tracing procedure.

1의 값을, 0이 많은 경우에는 0의 값을 k 개의 해당 비트 값으로 설정하여 k 비트 길이의 핑거프린트 FP_{stego} 를 생성한다.

나. 공모자 검출

공모 공격에 가담한 공모자를 검출하기 위하여 스테고 메쉬에서 추출한 핑거프린트 FP_{stego} 와 핑거프린트를 생성할 때 만들어진 $q+1$ 개의 핑거프린트 $FP_1, FP_2, \dots, FP_{q+1}$ 를 서로 비교하여 다음과 같은 방법으로 적어도 한 사람 이상의 공모자를 찾을 수 있다. 만약, 공모자들이 서로 공모하여 서로 다른 부분인 마크된 분할메쉬들을 모두 공격한다면, 스테고 메쉬에서 추출한 핑거프린트 FP_{stego} 에는 공모자들의 핑거프린트에 여전히 존재하는 마크 비트 즉, 공모 공격에도 여전히 살아남는 비트 정보를 바탕으로 모든 공모자들을 검출할 수 있다.

또한 공모자들이 서로 공모하여 발견한 마크된 분할메쉬들 중 몇 개의 분할메쉬만을 임의로 공격한다면, 스테고 메쉬에서 추출한 핑거프린트 FP_{stego} 에는 1로 설정되어 있는 마크 비트 정보를 바탕으로 핑거프린트를 생성할 때 만들어진 $q+1$ 개의 핑거프린트들과 서로 비교하여 마크 비트 정보를 가장 많이 포함하고 있는 핑거프린트를 찾아 공모자를 검출할 수 있다. 이때 잉여 변수 r 값이 클수록 공모 공격에 강인하게 되므로 정확한 공모자를 검출할 확률을 높일 수 있다.

그림 4는 $PG(2, 2)$ 를 기반으로 생성된 세 개의 핑거프린트를 할당받은 사용자들 중 공모자가 두 명인 경우의 공모 공격에 대하여 공모자를 검출할 수 있는 예를 보여준다.

그림 4(a)는 사용자 중 FP_1 과 FP_2 를 할당받은 사용자 둘이 서로 공모하여 발견한 모든 상이한 분할메쉬를 공격함으로써 4번째 비트를 제외한 1번째, 2번째, 3번째, 6번째 마크 비트 정보를 모두 잃은 FP_{stego} 를 스테고 메쉬로부터 추출할 수 있으며 이 정보를 바탕으로 사용자 정보에 있는 핑거프린트 정보와 비교함으로써 4번째 비트가 1인 핑거프린트는 FP_1 과 FP_2 이므로 공모자 둘

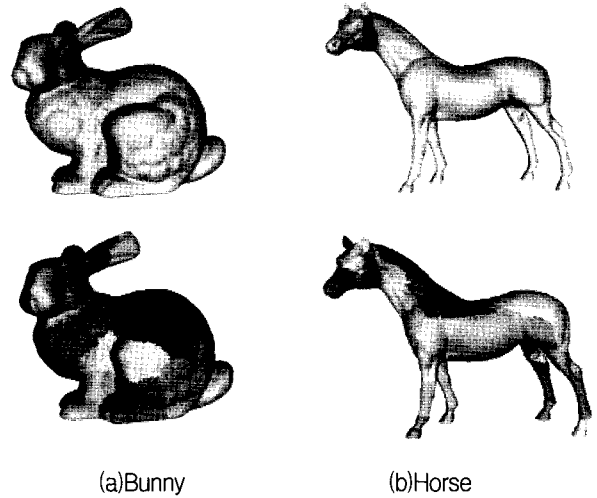


그림 5. 실험 메쉬 모델과 메쉬 분할
Fig. 5. Test models and their mesh partitioning.

표 1. 스테고 키 k_1 에 따라 선택된 분할 메쉬
Table 1. Submeshes selected by stego-key k_1 .

분할메쉬	5	9	8	10	21	13	14	12	6	29	20	28	11	4	18	30	2	16
FP_1	1	1	0	1	0	0	1	1	0	1	0	0	1	1	0	1	0	0
FP_2	0	0	1	1	0	1	0	0	1	1	0	1	0	0	1	1	0	1
FP_3	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1

을 모두 검출할 수 있다. 반면, 그림 4(b)와 같이 1번째, 2번째, 6번째 비트에 해당하는 분할메쉬만을 임의로 공격하는 경우 스테고 메쉬로부터 추출한 FP_{stego} 에는 3번째와 4번째 비트가 1로 설정되어 있으므로 이를 바탕으로 이 두 비트를 모두 포함하고 있는 핑거프린트를 사용자 정보에서 찾아 공모자를 검출할 수 있다.

V. 실험

그림 5는 실험에 사용된 삼차원 메쉬 모델인 Bunny (15095 꼭지점, 30019면), Horse (15002 꼭지점, 30000면) 모델과 각각의 모델에 대하여 30개로 분할된 메쉬 분할을 보여준다. Bunny 모델과 Horse 모델은 각각 Stanford 대학과 Cyberware Inc. 에서 제공하는 모델을 단순화시켰다.

본 논문의 실험을 위해 $PG(2, 2)$ 를 기반으로 하여 생성된 세 개의 핑거프린트 $FP_1=110100$, $FP_2=001101$, $FP_3=100011$ 를 삼차원 메쉬 모델에 적용하였다.

일례로 Bunny에 핑거프린트를 삽입하기 위하여 사용된 입력 변수 값들은 $v=15095$, $k=6$, $r=3$, $s=30$, $m=7$, $c=3$, $\alpha=0.0001$ 이다. 30개로 분할된 지역 분할 메쉬를 표 1과 같이 스테고 키 k_1 값에 의하여 임의로 선택하여

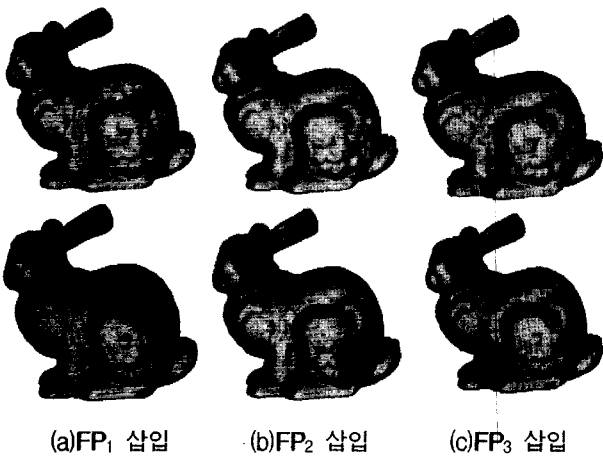


그림 6. 핑거프린트를 삽입한 Bunny 모델
Fig. 6. Fingerprinted Bunny models.

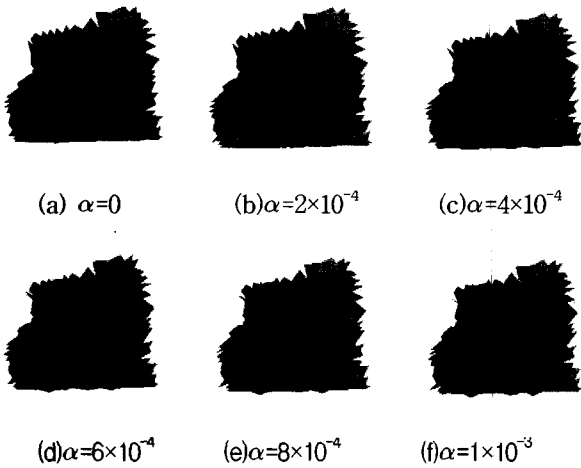


그림 7. 변조크기 변수 α 에 따른 분할메쉬이 왜곡
Fig. 7. Distortion of submeshes with different modulation amplitude.

커버 메쉬에 핑거프린트 정보를 마크하기 위하여 7비트의 워터마크 정보를 삽입한 스테고 메쉬와 마크된 분할 메쉬를 다른 색으로 구별해서 각각 그림 6에 나타내었다.

제안한 알고리즘으로 삽입한 신호의 비인지성을 정량적으로 평가하기 위해 커버 메쉬와 스테고 메쉬 간의 차이를 EPFL에서 제안한 MESH(Measure Error between Surface using the Hausdorff distance)^[14]를 이용했다.

MESH의 측정방법은 삼각형들을 정규격자(regular grid) 구조로 만들어 표본화 과정을 수행한 다음, Hausdorff 거리를 이용하여 평균 오류값과 평균 제곱근(RMS) 오류값을 계산한다. 이때 표본화 간격은 측정거리의 정확도에 중요한 역할을 하며 바운딩 박스(bounding box)의 대각선 길이에 0.5%나 0.4%이하의

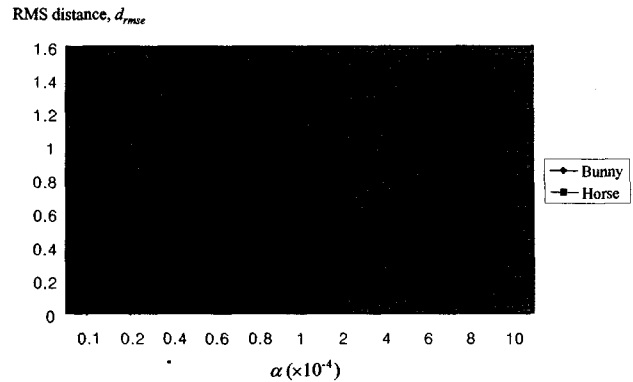


그림 8. 변조크기 변수 α 에 따른 RMS 거리
Fig. 8. RMS distance versus modulation amplitude α .



그림 9. Bunny 모델에 대한 다양한 공격
Fig. 9. Various attacks to Bunny.

값을 사용할 수 있다. 본 논문에서는 커버 메쉬, 공격 받지 않은 스테고 메쉬, 공격 받은 스테고 메쉬 사이의 평균 제곱근 오류값 d_{rms} 을 계산하기 위하여 표본화 간격으로 0.5%를 선택하였다.

그림 7과 그림 8은 각각 Bunny 모델의 왼쪽 앞발부분의 분할메쉬에 대해서 값을 다르게 워터마크를 삽입하였을 경우의 왜곡된 정도와 Bunny 모델과 Horse 모델에서 α 값에 따른 커버 분할메쉬와 스테고 분할메쉬 사이의 d_{rms} 값을 비교하여 그래프로 보여준다.

이 그래프로부터 워터마크가 삽입된 Bunny와 Horse 모델에서 커버 분할메쉬와 스테고 분할메쉬 사이의 차를 시각적으로 식별하기 어렵기 위해서는 d_{rms} 값으로 0.2이하의 값을 선택하는 것이 비교적 바람직하다는 것을 알 수 있다.

제안된 공모방지 핑거프린팅 기법이 다양한 공격으로부터 삽입한 신호가 살아남는지에 대한 실험을 위해, 무작위 잡음첨가 공격, MPEG-4 SNHC 표준의 꼭지점 좌표값 압축^[15], 평행 이동, 확대 및 축소, 셰어링(shearing)과 같은 기하학 변환을 하였으며 또한, 두 명의 공모자가 꼭지점 좌표값을 서로 평균하는 방식의 공모 공격을 적용시켰다.

그림 9는 무작위 잡음 첨가, 압축 공격, 기하학 변환, 공모 공격에 의해 공격을 당한 Bunny 모델을 보여준다.

표 2. Bunny 모델과 Horse 모델에 대한 실험 결과
Table 2. Test results for Bunny and Horse.

공격유형	실험 모델	Bunny		Horse	
		BER_w	BER_f	BER_w	BER_f
공격 없음		0	0	0	0
잡음 첨가 (%)	0.2	0	0	0	0
	0.4	0	0	0.015873	0
	0.6	0.031746	0	0.0409389	0.166
	0.8	0.063492	0	0.095238	0.166
좌표값 압축 (%)	80	0	0	0	0
	85	0	0	0	0
	90	0	0	0	0
기하학 변환	확대 / 축소	0	0	0	0
	평행이동	0	0	0	0
	색어영	0	0	0	0

표 3. 평균에 의한 공모 공격의 실험 결과
Table 3. Test results of collusion attack by average.

공격 유형	기법	Dittmann의 기법	제안한 기법
		CCR(%)	CCR(%)
평균 계산		70	100

위와 같은 다양한 공격을 Bunny 모델과 Horse 모델에 적용하여 제안한 핑거프린트 삽입 방법과 워터마킹 알고리즘의 강인성을 보여주는 실험 결과가 표 2에 있다.

여기서 무작위 잡음 첨가 비율(%)은 Bunny 모델의 바운딩 박스의 x, y, z 좌표축의 변위값 중 최대치와 모델의 기하학 정보에 더해질 잡음의 최대치와의 비율이다. 워터마크 비트 오류율 BER_w 은 워터마크가 삽입된 모든 스테고 분할메쉬에서 발생하는 오류 비트 수를 삽입한 워터마크의 총 비트 수로 나눈 값이며, 유사하게 핑거프린트 비트 오류율 BER_f 은 스테고 메쉬에서 추출한 핑거프린트에서 발생한 오류 비트 수를 핑거프린트의 비트 수로 나눈 값이다.

표 3은 커버 메쉬에 삽입한 핑거프린트에 대한 두 사람의 공모 공격에 대한 실험 결과를 보여준다. 비록 Dittmann이 정지영상에 대하여 공모 공격 실험을 하였지만, 실험 방법에 있어서 각각의 핑거프린트가 삽입된 영상을 평균하는 공격을 취하였기 때문에 본 논문에서 사용한 삼차원 메쉬 모델에서의 메쉬 평균에 의한 공모 공격 실험과 유사함으로 서로 비교하였다.

Dittmann의 핑거프린팅 기법에 비하여 제안한 핑거프린팅 기법이 정확한 공모자 검출율(CCR, Correct Customer Recognition)이 더 높은 이유는 Dittmann이 사용한 워터마킹 알고리즘이 평균 계산에 의한 공모에 강인하지 못하여 마크 비트의 오류율이 높았기 때문이다

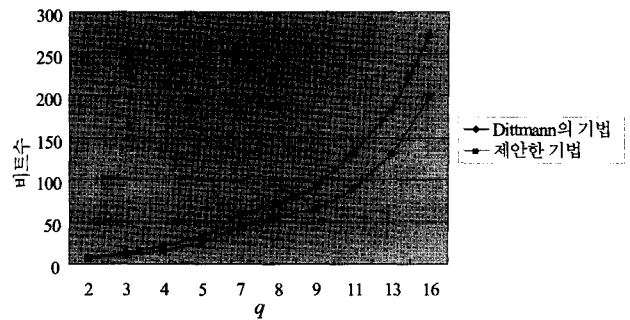


그림 10. $PG(2, q)$ 에서 생성된 핑거프린트의 비트 수
Fig. 10. The number of bits assigned to fingerprints in $PG(2, q)$.

며, 제안한 방법에서는 평균 계산에 의한 공모 공격에 강인하여 마크 비트의 오류율이 없었다.

또한 제안한 핑거프린팅 기법에서 각각의 핑거프린트에 할당되는 비트 수를 Dittmann의 기법에 비교하여 현저하게 줄일 수 있었다. 그림 10은 $PG(2, q)$ 에서 변수 q 에 따라서 Dittmann의 기법과 제안한 기법에서 생성되는 비트 수를 비교한 것을 보여준다.

$PG(2, q)$ 를 토대로 생성되는 비트 수는 Dittmann의 기법에서는 q 에 따라서 q^2+q+1 에 의하여 증가하는 반면, 제안한 기법에는 $\sum_{i=1}^q i$ 만큼 증가한다. 따라서 q 값이 증가할수록 더 많은 비트 수를 줄일 수 있다.

VI. 결 론

본 논문에서는 공모자가 2명인 경우에 삼차원 메쉬 모델에 적용할 수 있는 공모방지 핑거프린트 기법을 제안했다. 제안된 공모방지 핑거프린팅 기법을 이용하여 삼차원 메쉬 모델에 대한 공모방지 뿐만 아니라 불법적인 배포로부터 소유권을 주장할 수가 있다. 먼저, 제안된 핑거프린트 삽입 방법으로 공모방지 핑거프린트를 생성하여 이 정보를 바탕으로 삼차원 메쉬 모델을 분할하였다. 마크할 특정 분할메쉬만을 선택하여 DCT 변환을 한 후 워터마크 신호의 비인지성과 강인성을 고려하여 중간 주파수 대역에 저작권 정보인 워터마크 신호를 각각의 좌표축에 삽입하였다. 따라서 워터마크 신호를 보다 강인하게 탐지해 낼 수 있기 때문에 마크된 분할 메쉬로부터 핑거프린트 신호를 추출해냄으로써 공모 공격에 보다 강인하게 공모자를 찾을 수 있었다. 실험 결과로부터 제안한 핑거프린팅 기법이 잡음 공격, 압축 공격, 기하학 변환 및 두 사람에 의한 메쉬 평균 공격에 대하여 강인함을 알 수 있었다. 또한 각각의 핑거프린트에 할당된 비트 수를 Dittmann의 기법에 비하여 현

저하게 줄일 수 있었다. 향후, 우리는 기존 핑거프린팅 알고리즘의 단점인 핑거프린트 비트 수의 급격한 증가에 대하여 비트 수를 줄이는 연구를 계속할 것이며, 메쉬 단순화와 같은 연결성 정보에 대한 공격에 대해 취약한 문제점을 개선해 나갈 예정이다.

참 고 문 헌

- [1] K. Tanaka, Y. Nakamura, and K. Matsui, "Embedding Secret Information Into a Dithered Multilevel Image", Proc. IEEE Military Communications Conference, pp. 216-220, 1990.
- [2] G.R. Blackley, C. Meadows and G.B. Purdy, "Fingerprinting Long Forgiving Messages," Proc. of CRYPTO '85, Vol. 218, pp. 18-22, 1986.
- [3] D. Boneh and J. Shaw, "Collusion-secure Fingerprinting for Digital Data," IEEE Trans. Inf. Theory, Vol. 44, No. 5, pp. 1897-1905, Sept. 1998.
- [4] B. Pfitzmann and M. Schunter, "Asymmetric Fingerprinting," Proc. of EUROCRYPT '96, Vol. 1070, pp. 84-95, 1996.
- [5] B. Pfitzmann and M. Waidner, "Anonymous Fingerprinting," Proc. of EUROCRYPT '97, Vol. 1233, pp. 88-102, 1997.
- [6] J. Dittmann, "Combining Digital Watermarks and Collusion Secure Fingerprints for Customer Copy Monitoring," Proc. IEE Seminar Sec. Image & Image Auth., pp. 9-17, 2000.
- [7] J. Domingo-Ferrer and J. Herrera-Joancomart, "Simple Collusion-secure Fingerprinting Schemes for images," IEEE International Conference on Information Technology: Coding and Computing, ITCC2000, pp. 128-132, Mar. 2000.
- [8] D. Kirovski, H.S. Malvar, and Y. Yacobi, "Multimedia Content Screening Using a Dual Watermarking and Fingerprinting System," ACM Multimedia, 2002.
- [9] S. Katzenbeisser and F.A.P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House, 2000.
- [10] 전정희, 호요성, "DCT 영역에서 3차원 다각형 메쉬 모델의 디지털 워터마킹 방법," 전자공학회 논문지, 제40권 CI편 제3호, pp. 30-37, 2003년 5월
- [11] Pierre Terdiman's web site: <http://codercorner.com/>
- [12] J. H. Ahn, C. S. Kim, C.-C. Jay Kuo and Y. S. Ho, "Motion Compensated Compression of 3-D Animation Models," SPIE Visual Communications and Image Processing, pp. 593-602, Jan. 2002.
- [13] F. Hartung, P. Eisert, and B. Girod, "Digital Watermarking of MPEG-4 Facial Animation Parameters," Computer and Graphics, pp. 425-435, 1998.
- [14] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "MESH: Measure Error between Surfaces using the Hausdorff distance," Proc. IEEE International Conference on Multimedia and Expo 2002, vol. I, pp. 705-708, Aug. 2002.
- [15] ISO/IEC 14496-2, Coding of Audio-Visual Objects: Visual (MPEG-4 video), Committee Draft, Oct. 1997.
- [16] 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. Available at: <http://wwwusers.cs.umn.edu/~karypis/metis/metis.html>

저 자 소 개

허 영(정회원)

2001년 2월 홍익대학교 전자전기공학부 졸업
(학사).
2003년 2월 광주과학기술원 정보통신공학과 졸업
(석사).
2003년 3월~현재 국가보안기술연구소 연구원.
<주관심분야: 디지털 워터마킹, 디지털 신호처리,
암호학>



호 요 성(종신회원)

1981년 2월 서울대학교 전자
공학과 졸업(학사).
1983년 2월 서울대학교 전자
공학과 졸업(석사).
1983년 3월~1995년 9월 한국전자
통신연구소 선임연구원.
1989년 12월 미국 University of California, Santa
Barbara Department of Electrical
and computer Engineering(박사).
1990년 1월~1993년 5월 미국 Philips 연구소
Senior Research Member.
1995년 9월~현재 광주과학기술원 정보통신
공학과 교수.
<주관심분야: 디지털 신호처리, 영상신호 처리 및
압축, 초저속 영상통신, 디지털 TV와 고선명 TV
방식, 멀티미디어 통신, 디지털 워터마킹>



전 정 희(학생회원)

1993년 2월 조선대학교 컴퓨터
공학과 졸업(학사).
1997년 8월 조선대학교 컴퓨터
공학과 졸업(석사).
2001년 2월 조선대학교 컴퓨터
공학과 졸업(공학박사).
2001년 2월~2003년 2월 광주과학기술원 정보
통신공학과 박사후 과정
(Post Doctor).
2003년 7월~현재 (주) 지오소프트 개발이사/
연구소장.
<주관심분야: 3차원 컴퓨터 비전, 그래픽 데이터
및 볼륨 데이터 신호처리, 3차원 모델 워터마킹,
모바일 GIS>

