기하학적 공격에 강인한 핑거프린팅 기술 개발

충남대학교 이홍로 · 신정섭 · 김형신 · 황치정

1. 서 론

컴퓨터 네트워크와 멀티미디어 기술 및 전자 상거래의 발전에 따라 다양한 디지털 콘텐츠가 개발되고 있으나 콘텐츠 저작권의 보호는 아직 미흡한 실정으로 국내외적으로 이에 대한 연구가 활발히 진행되고 있다. 콘텐츠 보호 기술은 콘텐츠 제작자의 저작권 관련 정보를 각종 영상 변환에 강인하도록 콘텐츠에 삽입하는 기술로 정의할 수 있다. 이는 크게 워터마킹 기술과 핑거프린팅 기술로 나눌 수 있다.워터마킹 기술은 디지털 콘텐츠 제작자의 저작권 정보를 워터마크로 변환시켜 비가시적으로 콘텐츠에 삽입하는 기술로써 콘텐츠 제작자의 소유권을 인증할 수 있는 기술이지만 콘텐츠의 불법 유통 과정을 알 수 없다는 단점이 있다. 핑거프린팅 기술은 이러한 워터마킹 기술의 단점을 보완하기 위한 기술로써 콘텐츠의 유통 시 소유자의 정보뿐만 아니라 구매자의 고유한 정보를 콘텐츠에 삽입하여 다중 복사 및 불법 유통 과정을 추적할 수 있는 체계를 제공해 주는 보호 기술이다. 서로 다른 구매자 정보를 콘텐츠에 삽입하기 때문에 핑거프린팅 된 콘텐츠에 서로 다른 워터마크가 삽입된다. 따라서 핑거프린팅 기술은 워터마킹 기술을 사용한 경우보다 불법 복제를 더 억제 할 수 있는 보다 적극적인 의미의 저작권 보호 방법이라 할 수 있다.

현재 핑거프린팅 코드의 삽입 및 추출에 대한 많은 연구가 진행되고 있으나 대부분의 연구가 특정 공격에 대해서만 강인한 면을 보이는 문제점을 가지고 있다. 따라서 본 논문에서는 기하학적 공격과 평균화 및 공모 공격에 강인한 핑거프린팅 알고리즘을 제안한다.

본 논문에서 제안한 핑거프린팅 알고리즘은 퓨리에 변환을 통해 얻어진 주파수 영역에 워터마크를 삽입하고, 웨이블릿 변환을 통해 얻어진 영역에 핑거프린트 코드를 각각 삽입한다. 워터마크와 핑거프린트가 삽입된 영상으로부터 워터마크를 먼저 추출하여 영상을 복원한 후 복원된 영상으로부터 핑거프린트를 추출해 낸다. 핑거프린트 코드를 추출 시 원본 영상이 필요하지 않는 넌블라인드(non-blind) 기법을 사용하고 저주파 영역에 코드를 삽입하는 방법을 고안하였으며 다양한 공격에 대해서 보다 강인함을 획득하면서 화질 열화를 최소화하기 위하여 왓슨(Watson)의 모델을 이용하여 삽입강도를 정하였다[1]. 핑거프린트 코드 자체도 중복성이 있으므로 ECC 중에서 적은 중복성을 가지는 해밍코드(Hamming code)를 사용하여 추출 결과에 대한 신뢰성을 높였다.

본 논문의 구성은 다음과 같다. 2장에서는 핑거프린팅 코드의 삽입 및 추출에 관련된 최근 연구 동향에 대해 분석하였으며, 3장에서는 기하학적 변형을 위한 워터마크와 핑거프린트 코드의 삽입 및 추출에 대해 설명하였다. 4장에서는 이에 대한 구현과 성능에 대한 평가를 서술하였다. 마지막으로 5장에는 본 연구의 결론 및 향후 연구 방향에 대해 기술하였다.

2. 관련 연구

워터마크를 삽입하는 공간은 공간영역(spatial domain) 또는 주파수 영역(Frequency Domain)을 사용하는 크게 두 가지 방법이 있다. 공간영역을 이용하는 경우 영상을 공간적으로 분석하여 코드를 영상 전체에 고루 삽입하여 쉽게 구별할 수 없도록 한다. 공간영역 삽입 방법은 보통 화소값의 미세한 변화를 이용하지만 손실압축과 필터링과 같은 영상처리 기법에 약하다. 주파수영역을 이용하는 경우는 영상을 주파수 성분으로 변환하여 주파수 성분에 코드를 삽입하는 방법으로써 이 경우 코드가 삽입된 후 주파수 성분을 원래 영상으로 역변환 했을 경우 코드가 영상 전체에 고르게 분포하는 장점이 있다[2]. 주파수 성분으로 변환하는 방법은 이산 코사인 변환(Discrete Cosine Transform), 고속 퓨리에 변환(Fast Furierr Transform) 그리고 웨이블릿 변환(Wavelet transform)이 있다.

영상을 주파수 성분으로 변환하는 방법 중에서 JPEG, MPEG 1-2와 같은 표준 압축 방식의 근간인 DCT가 워

터마크 삽입에 많이 이용되어 왔으나[3-6] Meerwald [7]는 다중 해상도 분석을 기반으로 하는 웨이블릿 변환이 블록 기반의 DCT에 비해 HVS(Human Visual System)에 유리함을 보였고 Podilchuk[4]는 웨이블릿 변환을 사용하는 것이 DCT를 사용하는 것보다 JPEG 압축에 대한 강인성이 높음을 실험 결과로 제시하였다.

웨이블릿 주파수 영역에 코드를 삽입할 경우, 사용되는 주파수 영역의 특성에 따라 저주파 또는 고주파 영역을 이용한다. 저주파 영역에는 영상 정보 대부분이 몰려 있고 고주파 부분에는 나머지 성분들이 포함되어 있다. 저주파 영역에 수정을 가할 경우 역변환 후 영상의 변화가 크므로 상대적으로 화질저하가 적은 고주파 영역을 이용하는 경우가 많으나 그 경우 공격에 대한 강인함이 떨어진다. 그래서 영상처리에 강인한 저주파 영역에 워터마크를 삽입하는 연구가 소개되고 있다[8].

워터마크 삽입 알고리즘에서 워터마크 삽입 강도를 잘못 결정할 경우 영상의 화질저하가 뚜렷이 나타날 수 있다. 이유는 영상마다 그 특성이 다른데 그와 상관없이 동일한 삽입강도를 사용하거나 테스트 영상을 이용한 실험 결과를 이용하여 삽입 강도를 고정시키면 다른 영상에 적용했을 경우에 신뢰도를 보장받기 힘들기 때문이다. 이와 같은 문제를 극복하는 접근 방법으로 HVS 모델로 영상을 분석하여 시각적인 중요도를 계산하여 워터마크 삽입 시 강도 조절에 이용하는 접근 방식이 있다[4-6, 9].

삽입된 워터마크를 보다 정확하게 추출하기 위하여 ECC(error correcting code)를 이용하는 방법이 있다. ECC는 정보에 데이터를 추가하여 저장 또는 전송 중에 생길 수 있는 에러를 원 코드로 수정하기 위해 고안되었다. 이것을 워터마킹에 적용하면 추출 시 워터마크가 삽입된 영상이 여러 공격을 당한 상태라 가정하면 추출된 신호에 많은 잡음 또는 에러가 포함될 것이다. 이를 ECC를 이용하여 극복하려는 시도가 있어 왔다. Hongtao Ge[10]의 경우 구현이 간단하고 적은 계산 복잡도를 가지는 선형 블록 코드인 (7,4)-해밍 코드(Hamming code)를 워터마킹에 적용하였다. Ching-Tang Hsieh[11]는 BCH code를 사용하여 오류 정정 코드를 사용하지 않은 경우보다 사용하는 경우가 워터마크의 추출 결과가 좋아짐을 보여주고 있다. Natasa Terzija[12]의 경우 (15,7)-Reed-Solomon code과 (31,11)-Reed-Solomon code 중 (15,7)-Reed Solomon code가 워터마킹에 더 유리함을 보여주고 있다.

3. 제안 알고리즘

본 연구에서 제안한 알고리즘은 기하학적 공격을 위한 RST 모듈과 핑거프린팅 정보의 삽입 및 추출에 관한 핑거프린팅 모듈로 나누어진다. 핑거프린팅 시스템의 구조는 그림 1과 같다.

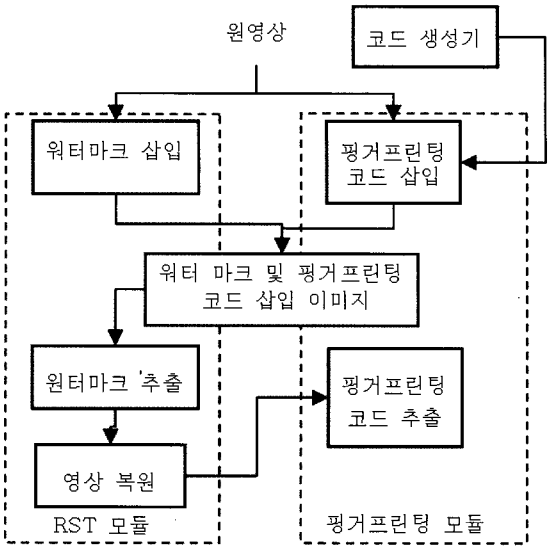


그림 1 핑거프린팅 시스템 구성도

핑거프린팅의 삽입은 워터마크의 삽입과 핑거프린팅 코드의 삽입 과정으로 이루어져 있다. 입력된 영상으로부터 기하학적 변환에 강인한 워터마크를 주파수 영역에 삽입한 후, 코드 생성기에 의해 생성된 핑거프린팅 코드를 웨이블릿 변환을 이용하여 삽입한다. 핑거프린팅의 추출은 핑거프린팅 코드가 삽입된 이미지로부터 기하학적 변환에 강인한 워터마크를 추출한 후, 영상을 복원한다. 복원된 영상으로부터 핑거프린팅 코드를 추출한다.

3.1 RST 모듈

본 모듈은 기하학적 변형에 강인한 워터마크를 영상에 삽입함으로써 핑거프린팅 코드의 추출이 가능한 영상으로 복원하는데 그 목적이 있다. 대부분의 기하학적 변형은 공간 영역에서 이루어지기 때문에 본 알고리즘에서는 기하학적 변형에 영향이 적은 주파수 영역에 삽입하는 것을 기본으로 한다. 주파수 영역은 영상의 회전 및 크기 변환에 따라 영상의 크기값(magnitude)에서 그 공격 강도를 계산해 낼 수 있으며, 절삭의 경우 영상의 위상(phase)에 이동(shift)만이 나타나기 때문에 기하학적 변형에 강인한 워터마크를 삽입할 수 있는 공간이다.

따라서 워터마크의 삽입은 푸리에 변환으로 얻어진 주파수 영역에 템플릿(template)을 생성하여 삽입한다. 템플릿은 원영상의 변형이 최소화될 수 있고, 추출할 때 필요한 검색시간을 줄일 수 있는 구조체를 고안하였다. 또한, 템플릿을 주파수의 중간 영역(mid-frequency)에 삽입함으로써 영상 압축에도 강인하도록 생성하였다.

워터마크의 추출은 주파수 영역의 크기값에서 피크들을 검색하여 삽입한 템플릿을 추출해낸다. 추출과정에서 극부분의 피크들을 제거하기 위해 바틀렛 윈도우(Bartlett window)를 적용하였다. 또한 추출된 템플릿들의 후보들(candidates) 중 정확한 템플릿을 찾아내기 위해 변환행렬(transformation matrix)을 적용해 얻어지는 최소제곱오차(mean square error)를 이용하였다.

3.1.1 워터마크의 생성과 삽입

주파수 영역에 삽입 할 워터마크 W_n는 n개의 점 (u_n, v_n)로 이루어진 템플릿 구조체이다. 수식 (1)과 같이 기울기 $\tan\theta$를 가진 직선의 형태로 생성되어진다. 그림 2는 생성된 워터마크의 형태를 나타내고 있다

$$v_n = (\tan\theta)\times u_n,\ \frac{u}{3} \le u_n \le \frac{2_u}{3} \qquad (1)$$

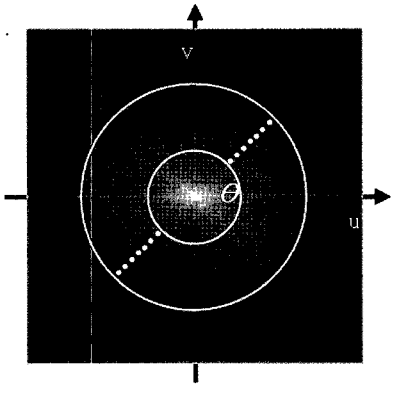


그림 2 워터마크의 생성

워터마크의 삽입 과정은 원 영상에 대해 1024x1024의 크기로 제로 패딩(zero padding)을 한 후, 푸리에 변환으로 얻어진 주파수 영역에 템플릿을 삽입하는 과정을 거친다. 마지막으로 역푸리에 변환을 통하여 공간영역의 템플릿이 삽입된 영상을 얻는다(그림 3 참조). 템플릿의 삽입은 영상 압축에 영향을 적게 받는 주파수의 중간 영역에 삽입된다. 또한 원 영상의 변형을 최소화하기 위해 마스크를 이용하여 주변 크기 값에 대한 평균값(mean)과 표준분산(standard deviation)의 합을 강도(strength)로 사용한다.

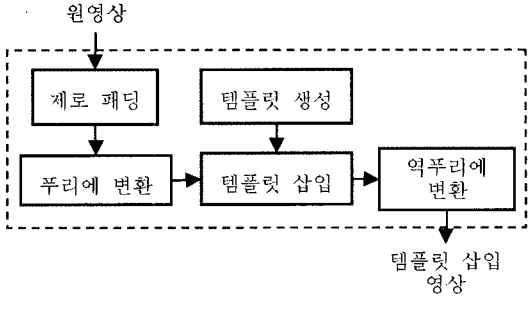


그림 3 워터마크 삽입

3.1.2 워터마크의 추출 및 영상 복원

워터마크의 추출은 워터마크가 삽입된 영상에 바틀렛 윈도우를 적용한 후, 1024×1024의 크기로 제로 패딩을 하고, 푸리에 변환을 통해 주파수 영역으로 변환한 다음 변환된 주파수 영역의 크기값으로부터 피크를 추출하여 각도 θ에 따른 직선으로 분리한다. 직선의 분리와 동시에 템플릿을 생성하고 분리한 직선과의 템플릿 매칭 과정을 가진다. 이렇게 해서 얻어진 영상의 기하학적 변환 데이터를 이용해 워터마크가 삽입된 영상을 복구해낸다(그림 4 참조).

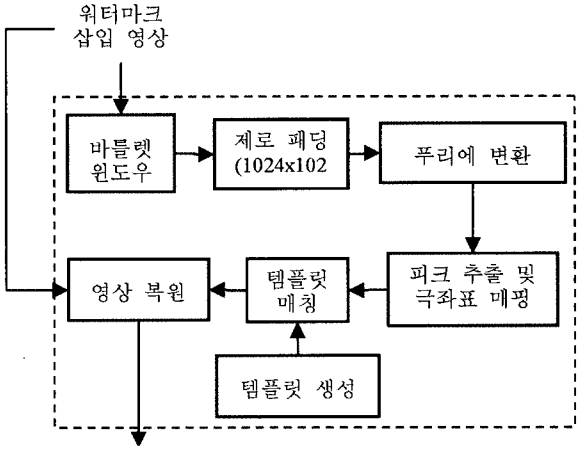


그림 4 워터 마크 추출 및 영상 복원

템플릿 매칭 시 발생하는 오차를 줄이기 위한 전처리 과정으로 워터마크가 삽입된 영상에 바틀렛 윈도우를 적용하여 주파수 영역에서의 극좌표상에 나타나는 높은 피크들을 제거하였다. 수식 (2)는 영상의 크기가 N일 때, 일반적인 바틀렛 윈도우의 계산식이다.

$$0 \le n \le \frac{N-1}{2}$$
$$\frac{N-1}{2} \le n \le N-1 \qquad (2)$$

영상 $f(x,y)$의 크기가 $0\le x<N_1, 0\le y<N_2$일 때, 바틀렛 윈도우를 적용하면 수식 (3)과 같이 쓸 수 있다.

$$bartlett(x,y) = f(x,y)\times w(N_1)\times w(N_2) \qquad (3)$$

피크점의 추출은 크기 $i\times j$를 가진 마스크 $w(i,j)$를 회선 처리하며 이웃점들 중 가장 높은 점을 선택하는 방식을 사용한다.

$$peak(u,v) = \begin{cases} 1 & If\ \ F(u,v) \ge w(i,j) \\ 0 & If\ \ F(u,v) < w(i,j) \end{cases} \qquad (4)$$

극 좌표의 매핑은 각각의 피크점들에 대한 반지름 r와 각도 θ를 계산하여 매핑 테이블을 생성함으로써 이루어진다. 수식 (5)는 반지름 r와 각도 θ를 구하는 식

을 나타낸다.

$$r = \sqrt{u^2 + v^2}$$
$$\theta = \frac{\tan^{-1}(u,v) \times 360}{2\pi} \quad (5)$$

이렇게 생성되어진 매핑 테이블에 대해 각도 θ로 정렬함으로써 피크점들을 직선 별로 구분할 수가 있다. 직선으로 구분되어진 피크점들을 이용해 템플릿을 선별한다. 템플릿이 될 수 있는 후보점들은 수 없이 많을 수 있다. 이들 중에서 삽입한 템플릿을 선별하기 위하여 직선에 포함된 피크점과 템플릿 간의 매칭기법을 이용한다.

직선 l에 포함된 n개의 피크점 $P_n = \{peak_0, \cdots peak_n\}$은 m개의 템플릿 $T_m = \{template_0, \cdots template_m\}$과 수식 (6)의 관계가 성립한다. 이때 특정 임계값을 이용하여 후보가 되는 템플릿을 선출해 낼 수 있다.

$$T_m = k \times P_n \leq Threshold \quad (6)$$

여기서 k는 크기 변환 비율을 의미하며, min≤k≤max값을 갖는다. $Threshold$는 위 식을 만족하는 후보점을 찾기 위한 임계치이다.

다음 과정은 위 과정을 통해 추출한 후보 템플릿들에 대해서 선형 변환 행렬(Linear Transform Matrix)을 구한 후, 이들을 변환시킨다. 변환시킨 후보 템플릿과 삽입한 원 템플릿 간의 최소 제곱 오차(Mean Square Error)를 구함으로써 최종 후보 템플릿을 선택할 수가 있다. 행렬 A는 수식 (5)와 수식 (6)에 의해 생성된 후보 템플릿들을 변환시키기 위한 선형 변환 행렬이다. CP는 템플릿 매칭 되어진 후보 템플릿 행렬이다. 그리고, T는 원영상에 삽입되어진 템플릿에 대한 행렬이다.

$$A = \begin{bmatrix} k_x \cos\theta & -\sin\theta \\ \sin\theta & k_y \cos\theta \end{bmatrix}$$
$$CP = \begin{bmatrix} x_{11} & y_{11} \\ \vdots & \vdots \\ x_{1m} & y_{1m} \\ x_{21} & y_{21} \\ \vdots & \vdots \\ x_{2m} & x_{2m} \end{bmatrix}, \; T = \begin{bmatrix} x'_{11} & y'_{11} \\ \vdots & \vdots \\ x_{1m} & y_{1m} \\ x_{21} & y_{21} \\ \vdots & \vdots \\ x_{2m} & x_{2m} \end{bmatrix} \quad (7)$$

모든 후보 템플릿에 대해서 최소 제곱 오차를 구한 후, 그 값이 최소가 되는 템플릿을 최종 템플릿 후보로 선택한다. 수식 (8)은 수식 (7)을 사용해 최소 제곱 오차를 구하는 식이다.

$$MSE(mean\,square\,error) = \frac{\sum [A(CP)^T - T]^2}{vmmatches} \quad (8)$$

영상의 복원은 템플릿 매칭에 의해 최종 선택되어진 템플릿의 크기 변환값과 회전값을 이용하여 주파수 영역에서의 특성을 공간 영역에 반영하는 과정을 거친다. 일반적으로 $0 \leq x < N_1, 0 \leq y < N_2$ 일 때, 공간 영역 $f(x, y)$의 푸리에 변환과 역푸리에 변환은 수식 (9)와 같이 정의할 수 있다.

$$F(u,v) = \sum_{x_1=0}^{N_1-1} \sum_{x_2=0}^{N_2-1} f(x,y) \cdot exp[-j2\pi xu/N_1 - j2\pi yv/N_2]$$
$$f(x,y) = \frac{1}{N_1 N_2} \sum_{x_1=0}^{N_1-1} \sum_{x_2=0}^{N_2-1} F(u,v) \cdot exp[j2\pi xu/N_1 + j2\pi yv/N_2] \quad (9)$$

또한, 크기값과 위상값은 수식 (10)과 같이 정의된다.

$$A(u,v) = |F(u,v)|$$
$$\Phi(u,v) = \angle F(u,v) \quad (10)$$

공간 영역과 주파수 영역의 크기변환, 회전값에 대한 특성은 수식 (11)과 같이 공간 영역 $f(x_1, x_2)$에서 영상의 크기 변환 a, b는 주파수 영역에서 $1/a$, $1/b$의 값을 갖는 성질이 있다. 즉, 공간영역에서의 크기 변환값은 주파수 영역에서의 크기 변환값과 반비례한다.

$$af(x,y) \leftrightarrow aF(u,v), f(ax,by) \leftrightarrow \frac{1}{|ab|} F(\frac{u}{a}, \frac{v}{b}) \quad (11)$$

또한 영상의 회전에 대해서는 주파수 영역의 회전값과 공간 영역에서의 회전값은 비례한다.

$$f(r, \theta + \theta_0) \leftrightarrow F(w, \phi + \theta_0) \quad (12)$$

따라서 주파수 공간에서 구해진 회전값과 크기변환값에 대해서 공격이 일어난 만큼 역변환을 해 줌으로써 영상을 복원할 수 있다.

3.2 핑거프린팅 모듈

핑거프린팅 코드 삽입 및 추출을 위한 알고리즘을 다음과 같이 제안한다. 삽입 과정은 웨이블릿 변환을 이용하여 원본 입력 영상에서 주파수 영역의 최저주파수 대역를 선택하여 왓슨의 HVS 모델을 이용하여 해밍 코드로 변환된 핑거프린트를 삽입한 후에 웨이블릿 역변환을 한다. 추출 과정은 원본없이 삽입 과정의 역순을 기본과정으로 하여 해밍코드를 추출한 후에 핑거프린팅 코드로 디코딩한다.

3.2.1 핑거프린팅 코드의 삽입

코드 삽입을 위한 공간으로 영상의 RGB 칼라 성분을 YUV 영역으로 변환하여 Y 영역에 삽입한다. 그 다음 Y 영역을 1단계 웨이블릿 변환을 하여 생성되는 LL , LH, HL, HL의 중에서 LL 영역이 최저주파 영역에 해당된다. 이 영역은 원본영상의 정보 대부분을 포함하고

있으므로 코드를 부주의하게 삽입한다면 영상의 화질 저하가 우려된다. 따라서 이 성분에 대해 왓슨 모델을 적용하여 가중치 테이블로 생성하고, 핑거프린트 코드는 해밍 코드로 변환한 후 주어진 영역에 삽입한다.

삽입알고리즘에 전체적인 과정은 아래 그림 5와 같고 삽입 방법은 수식 (13)과 같다.

$$LL'(i,j)=\begin{cases} LL(i,j)+P(i,j)\cdot H_k, LL(i,j)>T \\ LL(i,j)-P(i,j)\cdot H_k, LL(i,j)<T \end{cases} \quad (13)$$

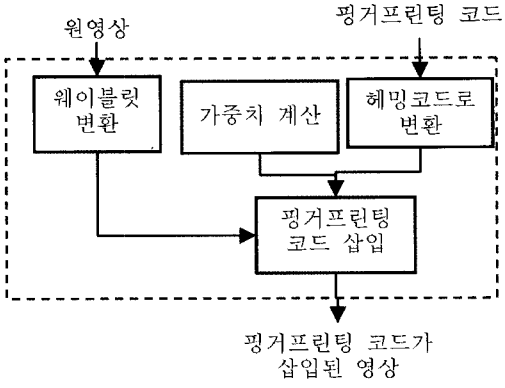


그림 5 핑거프린팅 코드 삽입 과정

P는 왓슨 모델로 분석된 가중치 테이블, H는 해밍 코드, T는 threshold이다. 삽입 과정이 완료된 이후에 코드가 삽입된 Y 성분과 저장된 U, V 성분을 이용하여 RGB 칼라 영상으로 복원한다.

3.2.2 핑거프린팅 코드의 추출

추출하는 방식은 기본적으로 삽입의 역순이며 원본 영상을 사용하지 않고 해밍코드를 추출하며 전체적인 과정은 그림 6과 같다.

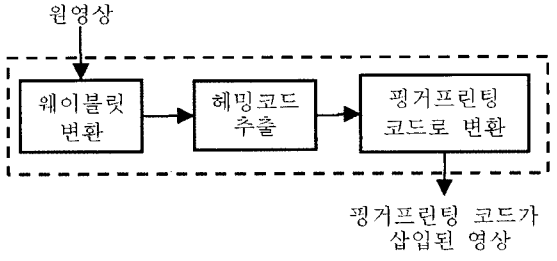


그림 6 핑거프린팅 코드 추출과정

코드가 삽입된 공간은 Y 성분으로 삽입과 동일한 변환 방식을 사용하였다. 그 다음 삽입 시와 동일한 웨이블릿을 이용하며 LL 성분을 분리한다. 그 후에 삽입된 해밍 코드를 수식 (14)를 사용하여 추출하고 그 코드를 디코딩하여 삽입된 원본 핑거프린팅 코드를 복원한다.

$$H_k=\begin{cases} 1, LL(i,j)>T \\ 0, LL(i,j)<T \end{cases} \quad (14)$$

3.2.3 핑거프린팅 코드의 검색

본 검색 알고리즘의 목표는 배포한 영상에 공모 공격을 가한 공모자를 추적하는 것이다. 본 알고리즘에서 대비한 공모 공격은 N 명의 사람이 모여서 자신들이 가진 영상을 합쳐서 N으로 나누어서 새로운 영상을 만드는 평균화 공격이다. 핑거프린팅 코드들은 동일 위치에 동일 비트가 중복되어 생성되므로 공모 공격을 당한 영상에서 추출된 코드는 공모 공격에 가담한 공모자들의 코드와 중복된 코드가 포함되어 있으므로 이 성질을 이용하여 확률적으로 가장 유사한 코드를 검색하는 것이 본 검색 알고리즘의 목표이다.

공모자를 검색하는 방법은 우선 공모영상에서 추출된 코드와 저장된 코드들 사이의 유사도를 측정하여 정렬한다. 코드 사이에 중복된 비트가 있으므로 유사도가 높은 코드가 공모자일 가능성이 높으므로 정렬된 자료에서 상위 5개의 코드를 공모자 후보 코드로 본다. 이들 중 공모자를 정확하게 검색하기 위해서 5개의 코드를 원본 영상에 각각 삽입하고 이 5개의 영상을 3개씩 조합하여 공모영상을 다시 만든다. 이렇게 만들어진 공모 영상이 현재 가지고 있는 공모 공격 당한 영상과 가장 유사한 영상들이라고 가정하고 만들어진 10개의 공모영상에서 각각 코드를 추출하여 실제로 공모공격이 가해진 영상에서

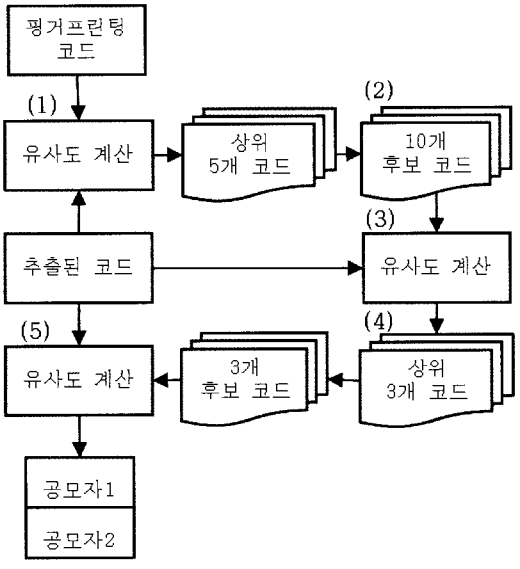


그림 7 공모자 검색

추출된 코드와 유사도를 다시 측정한다. 이 중 유사도가 가장 높게 나온 공모영상을 만든 3개의 코드의 조합을 가장 유력한 3명의 공모자 후보로 가정한다. 후보 코드 3개를 다시 위와 같은 방식으로 2개씩 조합하여 공모영상을 만든 다음 위와 같은 방식으로 다시 높은 유사도를 가지는 2개의 코드가 결정되면 이 코드를 할당 받은 사용자를 공모자로 판단한다. 핑거프린트 코드를 검색하는

기본적인 과정은 그림 7과 같고 공모자 2명을 찾는다.

4. 구현 및 성능 평가

4.1 실험 환경

본 시스템의 실험을 위하여 Visual C++ 6.0로 구현하였으며, 인텔 펜티엄4 1.5 Ghz, 512M의 컴퓨터에서 다양한 크기의 JPG, BMP 영상 22개를 대상으로 실험하였다.

4.2 워터마크의 삽입 강도에 따른 PSNR

워터마크의 삽입은 주파수 영역의 크기값(magnitude)에 대해서 9x9 마스크를 사용해 이웃 점들 간의 평균값(mean)과 표준분산(standard deviation)의 합을 삽입강도로 사용하였다. 삽입되는 강도를 조절하기 위해 표준분산에 10씩 증가하며 곱해주었다.

그림 8은 워터마크가 삽입된 영상의 PSNR에 대한 그래프를 나타낸다. 가로축의 삽입강도는 표준분산에 10을 곱해준 값이고, 세로축의 PSNR은 실험 영상들의 평균 PSNR을 나타낸다. 삽입 강도 10에서 100사이의 PSNR은 변형(distortion)이 식별 불가능한 40dB 이상을 유지하였다. 삽입 강도가 높아짐에 따라 PSNR은 감소하나 50dB에서 40dB사이에서 머물고 있음을 확인할 수 있다.

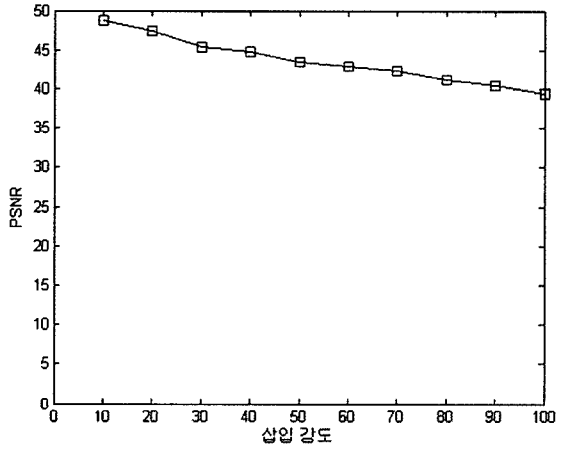


그림 8 워터마크 삽입 강도에 따른 PSNR

4.3 코드 삽입 후 측정된 PSNR

실험 영상에 대해서 9개 코드를 선정하여 코드를 삽입한 후에 영상의 변형정도를 수치적으로 알아내기 위해 PSNR을 조사하였다. 조사 방식은 한 키에 대해서 모든 영상의 PSNR을 조사하여 그 값들 중 최소값, 최대값 그리고 평균값을 기록하는 것이다. 그 결과는 표 1과 같으며 평균적으로 높은 값을 유지함을 보여주고 있다.

표 1 코드 삽입 후의 PSNR

Key	MIN	MAX	AVG
1	36.89431	52.41591	43.47306
51	37.40527	52.20637	43.45279
101	36.50715	52.01359	43.25637
151	36.19072	52.31905	43.27591
201	36.48120	52.01349	43.35736
251	37.40788	52.22057	43.38111
301	36.48507	52.93493	43.36694
351	36.15473	51.99401	43.34548
401	37.42504	51.65013	43.26589
451	37.18688	52.08204	43.36001

4.4 영상의 복원 결과

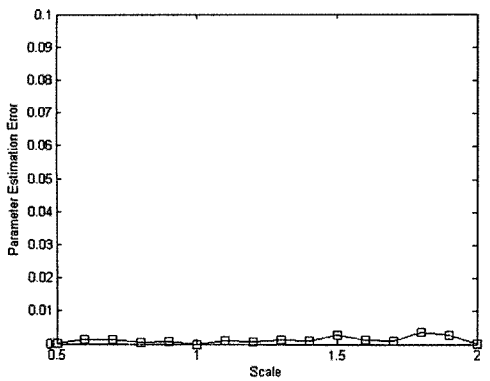
회전 및 크기변환을 실험하기 위해 StirMark 4.0을 사용하여 크기변환 비율 0.5에서 2.0의 비율과 회전각도 0°에서 180°로 공격한 후, RST 모듈을 이용해 복원을 하였다.

기하학적 공격에 대한 파라미터 추정 에러(Parameter Estimation Error)는 실험 영상의 평균으로 나타내었다. 크기변환에 대한 파라미터 추정 에러는 0.1을 넘지 않았으며, 회전에 대한 파라미터의 추정은 오차가 없는 정확한 결과를 얻었다. 그림 9는 크기변환, 회전에 대한 파라미터 추정 에러를 나타낸 그래프이다.

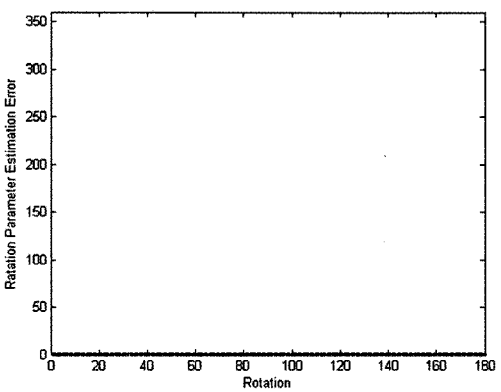
복원 영상의 MSE는 원영상과 복원영상에 대한 화소 차이를 이용해 구하였고, 축소 공격을 받는 영상은 복원하는 과정에서 생긴 보간(interpolation)에 의해 다소 큰 MSE가 얻어졌으며 확대 공격을 받은 영상일수록 적은 화질 열화로 영상을 복원해내었다.

크기 변환 비율 0.5에서 2.0 사이의 비율 중 원 영상으로 정확히 복원할 수 없는 경우가 있는데, 이는 영상의 크기 변환 시 사용되는 보간의 반올림 오차에 의해 원영상보다 최소 한 픽셀 이상의 차이가 나는 현상때문이다. 예를 들면, 512x512의 영상에 비율 0.7로 축소 공격을 가하게 되면 358x358의 영상이 생성된다. 이 영상을 완벽하게 복원하기 위해서는 1.0/0.7을 영상에 곱해주면 되지만, 복원된 결과 영상은 511.428의 크기를 갖으며, 반올림에 의해 511x511영상이 생성된다. 따라서, 본 MSE 측정에 대한 실험은 원 영상의 크기로 복원된 영상만을 가지고 실험하였다.

크기변환 공격에 대해 복원시킨 영상의 MSE 중 축소 공격을 가한 영상의 MSE는 대체로 높은 결과를 얻었다. 이는 축소할 때 보간(interpolation)에 의해 영향을 받은 후, 다시 영상을 복구할 때 쓰이는 보간의 결과이다. 확대 공격을 가한 경우에는 확대 비율이 높아질

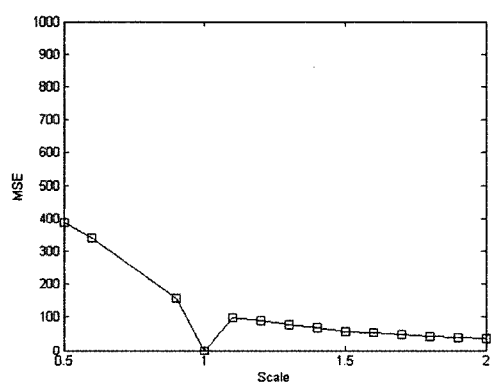


(a) 크기 변환

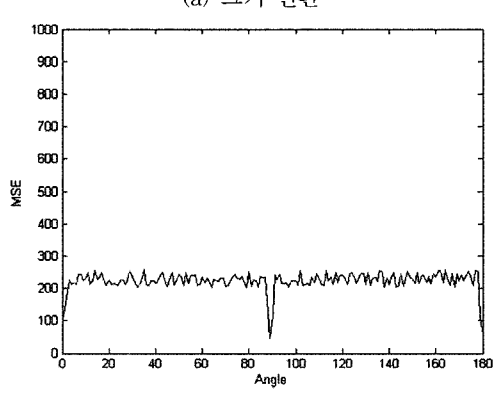


(b) 회전

그림 9 Parameter Estimation Error



(a) 크기 변환



(b) 회전

그림 10 복원 영상의 MSE

수록 MSE가 줄어드는 결과를 볼 수 있다. 회전 공격에 대해서는 0°와 90°, 180°를 제외하고는 MSE가 규칙적으로 150에서 300 사이에 머물렀다. 그림 10은 기하학적 공격에 대해 복원된 영상의 MSE를 나타낸다.

4.5 필터링 공격 후 코드 추출율

실험 영상에 대해서 9개 코드를 선정하여 코드가 삽입된 영상이 필터링 공격을 받았을 경우 코드의 추출률을 조사하였다. 이를 위해 인텔 이미지 프로세싱 라이브러리의 필터링 함수를 이용하였으며 Blur, Sharpen, Min, Max, Median, ColorMedian 필터를 사용하였다. 한 키에 대해서 필터링 공격을 당한 21개의 영상의 추출률을 조사하여 그 값들 중 최소값, 최대값 그리고 평균값을 기록하였다. 그 결과는 표 2와 같으며 평균적으로 높은 값을 유지함을 보여주고 있다.

표 2 필터링 공격 후 추출 결과

Filter	MIN	MAX	AVG
Blur 3x3	0.40467	0.92607	0.68995
Sharpen 3x3	0.63812	0.99611	0.84949
Blur&Sharpen	0.49027	1.00000	0.78546
Sharpen&Blur	0.49805	1.00000	0.78458
Min	0.23346	0.97276	0.55907
Max	0.22957	0.87160	0.54634
Median	0.40467	0.99611	0.73293
ColorMedian	0.412451	1.00000	0.72533

4.6 JPEG 압축 후 코드 추출율

실험 영상에 대해서 9개 코드를 선정하여 코드가 삽입된 영상이 JPG 압축 공격을 받았을 경우 코드의 추출률을 조사하였다. 한 키에 대해서 압축 공격을 당한 21개의 영상의 추출률을 조사하여 그 값들 중 최소값, 최대값 그리고 평균값을 기록하였다. 그 결과는 표 3과 같으며 평균적으로 높은 값을 유지함을 보여주고 있다.

4.7 공모 공격 실험 결과

공모 공격에 강인함을 실질적으로 보이기 위해 우선 2,3,5,7,10,18,22,32명에 대한 조합을 선택하여 실험 영상에 공격을 가한 후 핑거프린팅 코드를 검색하였다. 그 결과는 표 4와 같으며 공모자 수가 10명 미만일 경우 모두 찾으며 10이상일 경우 실패하는 영상이 발생한다.

그리고 baboon 하나의 영상에 대해서 모든 코드들을 대상으로 공모 조합을 만들어 최대한 테스트를 하여 전체적인 성공률을 추정하였다(표 5 참조). 공모자가 2명인 경우, 100% 공모자를 추출하였으나 10 이상인 경우, 잘못 찾는 경우가 발생하였음을 알 수 있었다.

표 3 압축 공격 후 추출 결과

JPEG	MIN	MAX	AVG
90	0.692607	1.000000	0.964096
80	0.443580	0.964981	0.881146
70	0.264591	0.879377	0.732048
60	0.245136	0.848249	0.628935
50	0.190661	0.774319	0.545631
40	0.186770	0.614786	0.466042
30	0.054475	0.603113	0.390343
20	0.093385	0.509728	0.298019
10	0.007782	0.365759	0.169261

표 4 공모 공격 sample 테스트

공모자수	결과	공모자수	결과
2	22/22	10	19/22
3	22/22	18	19/22
5	22/22	22	19/22
7	22/22	32	6/22

표 5 공모 공격 테스트

공모자수	총조합	테스트 개수	실패빈도	성공률(%)
2	입력요	60000	0	100
10	입력요	60000	326	99.5
20	입력요	11860	2055	82.6

5. 결론 및 향후 연구

본 연구에서는 하나의 콘텐츠를 대상으로 하는 기하학적 공격 및 압축, 필터링 공격과 다수의 콘텐츠를 서로 평균하여 새로운 콘텐츠를 생성하는 평균화 공모 공격을 대상으로 이산 웨이블렛 변환을 사용하여 영상을 주파수 영역으로 변환한 후, 그 주파수 영역에 일정 위치와 함께 랜덤하게 선정된 위치에 핑거프린팅 코드를 삽입하는 denoising 방법을 응용한 삽입 및 추출 알고리즘과 다수의 사용자가 공모하여 새로운 콘텐츠를 제작한 경우에도 공모자 일부를 추적할 수 있는 기법을 개발하였다. 실험 결과 기하학적 공격에 대한 파라미터를 정확히 추정하였으며, 이를 통해 최소의 오차를 가진 원영상을 복원할 수 있었다. 또한, 저주파수 대역에 핑거프린팅 코드를 삽입하였음에도 불구하고 화질열화의 발생을 최소화하였으며 PSNR 값이 평균적으로 40dB 이상을 유지하였고, JPEG압축과 필터링 공격에 강인함을 보였다. 그러나, 기하학적 공격에 대한 파라미터를 정확하게 추정하였음에도 불구하고, 원 영상으로 복원하는 과정에서 나타나는 보간에 의해 최소제곱오차의 값이 다소 높게 나타났으며, 특정 비율에 대해서는 원영상과 복원된 영상의 크기가 한 화소 정도 차이가 나는 문제점이 있었다. 그리고, 핑거프린팅 코드 삽입 과정과 ECC 과정을 서로 독립적으로 적용하여 사용한 해밍 코드의 에러수정 능력이 다른 ECC 알고리즘에 비해 떨어지는 문제점을 발견하였다.

앞으로 추출률을 보다 높이기 위해서는 기하학적 변형의 복원 과정에서 나타나는 화질 저하에 대한 더욱 완벽한 복원방법과 핑거프린팅 추출과정에서 ECC 알고리즘을 적용시키기 위한 더 깊은 연구가 필요하다.

참고문헌

[1] Watson A.B., Yang G.Y., Solomon J.A, and Villasenor J., "Visual Thresholds for Wavelet Quantization Error," SPIE Proceeding, Vol. 2657,paper #4,1999

[2] S. Pereira and T. Pun, "Robust template matching for affine resistant image watermarks," IEEE Trans. Image Processing, 9(6) 1123-1129 (2000)

[3] Hernandez, J.R., Amado, M. and Perez-Gonzalez, F., "DCT-Domain Watermarking Techniques for Still Images: Detector Performance Analysis and a New Structure," Image Processing, IEEE Transactions on, Vol. 9, pp. 55-68, Issue: 1, Jan. 2000.

[4] Podilchuk C.I and Wenjun Zeng, "Image adaptive watermarking using Visual Models," Selected Areas in Communications, IEEE Journal on, 16, pp. 525-539, 4, 1998.

[5] Podilchuk C. and Zeng W., "Perceptual watermarking of still images," Multimedia Signal Processing, 1997., IEEE First Workshop on, pp. 363-368, 23-25 June 1997.

[6] Wolfgang R.B.,Podilchuk C.I, and Delp, E.J, "Perceptual watermarks for digital images and video," Proceedings of the IEEE, Volume: 87, Issue: 7, pp. 1108-1126, July 1999.

[7] P. Meerwald, "Digital Image Watermarking in the Wavelet Trasform Domain," Master's thesis, Department of Scientific Computing, University of Salzburg, Austria, January 11 2001.

[8] Y.-S Seo and S.H Joo, "Robust Image

Watermarking using Quantization on the Lowest Wavelet Subband", Journal of The Korean Institute of Communication Sciences, Vol.28, No.9C, pp.898-907, Sep. 2003.
[9] Kaewamnerd N. and Rao K.R., "Wavelet based image adaptive watermarking scheme," Electronics Letters, 36, 312-313 , Issue: 4, Feb. 2000.
[10] Hongtao Ge, Fulin Su, and Yong Zhu., "Color Image Text Watermarking using Wavelet Transform and Error-correcting code," Signal Processing, 6th International Conference, Vol 2, pp.1584-1587. 26-30 Aug. 2002.
[11] Ching-Tang Hsieh and Yeh-Kuang Wu, "Digital Image Multiresolution Watermark Based on Human Visual System Using Error Correcting Code," Tamkang Journal of Science and Engineering, Vol. 4,No. 3, pp. 201-208, 2001.
[12] Terzija N., Repges M., Luck K., and Geisselhardt W., "Impact of different Reed-Solomon codes on digital watermarks based on DWT,"
[13] Trappe, W., Min Wu, Wang, Z.J., and Liu, K.J.R., "Anti-collusion Fingerprinting for Multimedia," Signal Processing, IEEE Transactions on, 51, 1069-1087, Issue: 4, 2003.
[14] Yang Zhao, Campisi, P.,and Kundur, D., "Dual Domain Watermarking for Authentication and Compression of Cultural Heritage Image," Image Processing, IEEE Transactions on, Vol. 13, pp. 430- 448, Issue: 3, March 2004.
[15] Morelos-Zaragoza, The Art of Error Correcting coding, 1th ed. England:Wiley, 2002.
[16] Inagmar j. Cox, Matthew L. Miller, and Jeffrey A. Bloom, Digital Watermarking, 1th ed. Morgan Kaufmann, 2002.
[17] Kundur, D and Hatzinakos, D., "A robust digital image watermarking method using wavelet-based fusion," Image Processing, 1997. Proceedings., International Conference on, Vol. 1, pp. 544-547, 26-29 Oct. 1997.
[18] Hsish C.T and Wu Y.K., "Digital Image Multiresolution Watermark Based on Human Visual System Using Error Correcting code," Tamkang Journal of Science and Engineering, Vol.4, No.3, pp. 201-208, 2001.
[19] Severine Baudry., Jean-Francois Delaigle., Bulent Sankur, Benoit Macq., and Henri Maitre., "Analyses of error correction strategies for typical communication channels in watermarking," Signal Processing in the ACM, Vol. 81, Issue. 6, pp. 1239-1250, 2001.
[20] Reza Safabakhsh, Shiva Zaboli and Arash Tabibiazar, "Digital Watermarking on Still Images Using Wavelet Transform," Proceeding of the International Conference on Information Technology: Coding and Computing, 2004.

이 홍 로

2002 충남대학교 컴퓨터과학과(이학사)
2004 충남대학교 컴퓨터과학과(이학석사)
2004~현재 충남대학교 컴퓨터공학과 박사과정
E-mail : hrlee@cnu.ac.kr

신 정 섭

2003 충남대학교 컴퓨터과학과(이학사)
2005 충남대학교 컴퓨터과학과(이학석사)
2005~현재 충남대학교 컴퓨터공학과 박사과정
E-mail : iplsub@cnu.ac.kr

김 형 신

1990 한국과학기술원 전자계산학과(학사)
1992 University of Surrey, satellite communication engineering, M.S
2003 한국과학기술원 전자계산학과(박사)
2005~현재 충남대학교 전기정보통신공학부 조교수
E-mail : kimhs@cs.cnu.ac.kr

황 치 정

1975 서강대학교 수학과(이학사)
1985 코네티컷 주립대학 전산학과(석사)
1987 코네티컷 주립대학 전산학과(박사)
1987 코네티컷 주립대학 객원교수
1988 한국원자력연구소 선임연구원
1999 충남대학교 전자계산소장
1997 충남대학교 정보통신연구소장
1988~현재 충남대학교 전기정보통신공학부 교수
E-mail : cjhwang@ipl.cnu.ac.kr