

일반화된 정사각형 마스크를 이용한 벡터 워터마킹 기법

강 환일, 김 갑일, 정 요원
명지대학교 전기정보제어공학부

A Vector Watermarking Using the Generalized Square Mask

Kang Hwan II, Kim Kab II, Jung Yo Won,
Myongji Univ., Div. of Electrical, Information & Control Eng..

Abstract - 본 연구는 벡터데이터에 워터마크를 삽입하고 추출하는 알고리즘을 제시한다. 삽입방법은 각 좌표를 마스크 안에서의 좌표로 변환한 후 워터마크의 크기에 따라 삽입영역을 다르게 설정한다. 이 방법은 기존의 Sakamoto 알고리즘을 일반화한다. 즉 기존의 마스크는 $2^k - 1$ 이지만 이 논문에서는 일반화된 정사각형을 사용할 수 있다. 또한 잡음에 강인함을 보인다.

1. 서 론

디지털 워터마크삽입은 저작권보호를 하기 위한 소극적 대처방법의 한 형태이다. 평거프린트는 배포된 디지털 콘텐츠에 사용자의 정보를 삽입하여 복제물의 유통을 검사하는 방법이다. 디지털 워터마크 방법은 원본 디지털 콘텐츠에 사용자가 감지하지 못하는 신호를 삽입하고 이를 다시 추출 할 수 있는 알고리즘을 뜻하는 데 삽입하는 신호는 디지털 콘텐츠의 저작권을 입증할 수 있어야 한다. 디지털 워터마크방법에는 디지털매체에 따라 정지영상 워터마크, 동영상 워터마크, 오디오 워터마크 텍스트 워터마크등 다양한 형태로 나타나고 있다. 또한 벡터이미지에 워터마크를 삽입하려는 시도가 있다. [1] 벡터로 기술된 콘텐츠는 데이터의 사이즈를 감소시키고 확대나 축소 시 그 이미지의 손상을 방지하기 위해 고안된 것이다. 벡터로 기술된 이미지는 각각의 좌표값과 경로의 형태를 지정해 줌으로서 이미지를 기술한다. 따라서 벡터이미지의 워터마크 삽입방법은 크게 위상영역, 벡터구조의 기술, 벡터구조, 또는 좌표치 그 자체에 삽입하는 방법으로 분류할 수 있다. 본 논문에서는 기존의 좌표치 그 자체에 워터마크를 삽입하는 Sakamoto 알고리즘을 개선하여 직사각형의 마스크를 이용하는 방법을 고안하였다.

2. 본 론

2.1 Sakamoto 알고리즘

벡터로 기술된 이미지는 각각의 좌표값과 경로의 형태를 지정해 줌으로서 이미지를 기술한다. 따라서 벡터이미지의 워터마크 삽입방법은 크게 위상영역, 벡터구조의 기술, 벡터구조, 또는 좌표치 그 자체에 삽입하는 방법으로 분류할 수 있다. Sakamoto 알고리즘은 이러한 워터마크 삽입방법 중 좌표치 자체에 변화를 줌으로서 워터마크를 삽입한다. 또한 Sakamoto 알고리즘은 크게 등면적 분할방식과 정점분할 방식으로 나눌 수 있으며 본 논문에서는 등면적 분할방식에 대하여 간단히 설명한다.

2.1.1 워터마크 삽입방법

Sakamoto 알고리즘에서 워터마크 삽입방법은 다음과 같다.
Step 1 : 32비트 키 정보 K로부터 길이가 N인 의사난

수열 $(r(n)) = (0, 1)$ 을 생성한다.

Step 2 : 삽입하려는 정보를 비트 길이가 N이 되게 늘인 삽입비트열 $\{S_{e(n)}\}$ 과 $\{r(n)\}$ 과의 배타적 논리화를 취하여 삽입 비트 열 $\{S_{e(n)}\}$ 을 생성한다.

$$S_{e(n)} = S_{e(n)} \oplus r(n) \quad (1)$$

Step 3 : 벡터 데이터(이미지)의 전체공간을 블록의 크기가 $H \times V$ 이 되도록 전체영역을 $A^{(n)}$ 으로 분할한다. ($n=1, 2, 3, \dots, N$)

Step 4 : 분할된 영역 $A^{(n)}$ 에 포함된 i번째의 정점 $V_i^{(n)}$ 에 관하여 좌표치 $V_i^{(n)}(x, y)$ 와 m 비트길이의 좌표마스크 M_m 으로부터 새로운 기본좌표 $E_i(x, y)$ 를 얻는다.

$$E_i(x) = V_i^{(n)}(x) \wedge M_m \quad (2)$$

$$E_i(y) = V_i^{(n)}(y) \wedge M_m \quad (3)$$

얻어진 기본 좌표를 기준으로 하여 $(M_m)^2$ 의 범위내에서 면적이 같게 되게끔 0.1의 영역을 결정해 삽입비트열 $\{S_{e(n)}\}$ 에 의하여 정점 좌표 $V_i^{(n)}(x, y)$ 를 변경한다.

Step 5 : 같은 영역내에 포함되는 전부의 정점에 대하여 Step 4의 처리를 반복한다. 즉, 같은 영역내의 점집군에 대하여 같은 비트 정보 $S_{e(n)}(x, y)$ 가 삽입되게 한다.

Step 6 : 전부의 영역 $A^{(n)}$ 에 대하여 Step 4.5의 처리를 행한다.

2.1.2 워터마크 추출방법

워터마크 추출방법은 삽입방법과 반대의 절차를 통하여 추출할 수 있으며, 그 내용은 다음과 같다.

Step 1 : 벡터 데이터(이미지)를 블록크기 $H \times V$ 의 $A^{(n)}$ 으로 나눈다.

Step 2 : 삽입 시와 같은 방법으로 영역 $A^{(n)}$ 에 포함된 정점의 좌표 $V_i^{(n)}(x, y)$ 와 기준좌표 $E_i(x, y)$ 를 구해 정점의 좌표가 위치하고 있는 장소가 기준좌표를 기점으로 해서 좌표마스크 M_m 에서 정해진 워터마크 정보 0.1의 영역의 어디에 속하는 것인가를 조사하고 1비트의 워터마크 정보를 읽어낸다.

Step 3 : 영역 $A^{(n)}$ 내의 모든 정점의 좌표에서 읽어낸 각각의 1비트의 워터마크 정보에 따라서 다수결을 취해, 이것을 영역 $A^{(n)}$ 에서 읽어낸 1비트의 워터마크 정보로 한다.

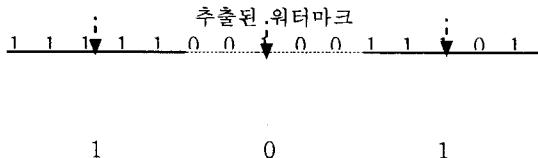
Step 4 : 모든 영역에 따라서 Step 3의 처리를 행하고, 각 영역 내에서 읽혀진 1비트의 워터마크정보를 결합해 읽어낸 비트열 $\{S_{e(n)}\}$ 을 생성한다.

Step 5 : 32비트의 키 정보 K에서 생성된 비트 길이 N

의 의사난수열 $\{\eta(n)\}$ 으로 읽어낸 비트열 $\{S_{\eta(n)}\}$ 과의 배타적 논리화를 취하는 것으로 n비트 길이에서 확장된 상태의 읽어낸 비트열 $\{S_{\eta(n)}\}$ 을 구한다.

Step 6 : 얻어진 $\{S_{\eta(n)}\}$ 을 그림 1과 같이 다수결을 취해 추출된 워터마크 정보를 얻는다.

그림 1. 추출된 워터마크 결정 방법.



2.2 제안된 알고리즘

Sakamoto 알고리즘에서 알 수 있듯이 사용되는 좌표마스크의 형태는 벡터 데이터(이미지)에 관계없이 항상 그 형태가 정방형($2^R - 1$)으로 고정되는 것을 알 수 있다. 이것은 데이터의 분포에 따라 워터마크를 삽입 시 원본 데이터의 왜곡과 임의의 좌표치 변화에 대하여 강인하지 못한 결과를 초래할 수 있다. 따라서 본 논문에서는 좌표마스크의 형태를 자유롭게(정사각형 형태) 결정할 수 있는 방법을 제안한다.

2.2.1 제안된 알고리즘의 워터마크 삽입방법

제안된 알고리즘에서 워터마크 삽입방법은 다음과 같다.

Step 1 : 32비트 키 정보 K로부터 길이가 N인 의사난수열 $\{\eta(n) = \{0, 1\}\}$ 을 생성한다.

Step 2 : 삽입하려는 정보를 비트 길이가 N이 되게 늘인 삽입비트열 $\{S_{\eta(n)}\}$ 과 $\{\eta(n)\}$ 과의 배타적 논리화를 취하여 삽입 비트 열 $\{S_{\eta(n)}\}$ 을 생성한다.

$$S_{\eta(n)} = S_{\eta(n)} \oplus \eta(n) \quad (4)$$

Step 3 : 벡터 데이터(이미지)의 전체공간을 블록의 크기가 $H \times V$ 이 되도록 전체영역을 $A^{(n)}$ 으로 분할하고 ($n=1, 2, 3, \dots, N$). 각 블록에 몇 개의 정점이 포함되는지 조사한다($V_{A(\eta(i))}$).

Step 4 : 분할된 영역 $A^{(n)}$ 을 새로운 키 정보 K1으로부터 길이가 N인 의사난수열을 생성하고 블록단위로 Permutation($\sigma(i)$)한 다음 Threshold T를 결정한다.

Step 5 : $V_{A(\sigma(i))} > T$ 인 모든 블록에 대하여 워터마크를 삽입한다.

2.2.2 제안된 알고리즘의 마스크와 새로운 기본좌표 결정방법

분할된 영역 $A^{(n)}$ 에 포함된 i번째의 정점 $V_i^{(n)}$ 에 대하여 좌표치 $V_i^{(n)}(x, y)$ 와 길이가 일반화한 마스크로부터 새로운 기본좌표 $E_i(x, y)$ 를 얻는다. 새로운 좌표값을 얻는 방식은 모우드합수를 이용한다. 얻어진 기본 좌표를 기준으로 하여 마스크의 범위내에서 면적이 같게 되게끔 0.1의 영역을 결정해 삽입비트열 $\{S_{\eta(n)}\}$ 에 의하여 정점 좌표 $V_i^{(n)}(x, y)$ 를 변경한다. $V_i^{(n)}(x, y)$ 의 변경방법은 마스크를 대각선으로 분할하여 왼쪽밑은 영의 영역으로 하고 오른쪽 위는 1의 영역으로 하고 만약 좌표의 변환이 요구되면 대각선을 거울로 삼아 대칭점을 구하여 이동한다.

2.2.3 제안된 알고리즘의 워터마크 추출방법

워터마크 추출방법은 삽입과정의 역순으로 진행되며, 추출과정은 다음과 같다.

Step 1 : 벡터 데이터(이미지)를 블록크기 $H \times V$ 의 $A^{(n)}$ 으로 나누고, 32비트의 키 정보 K1으로부터 길이가 N인 의사난수열을 발생시켜 Inverse Permutation을 행한다.

Step 2 : 각 블록에 대하여 $V_{A(\sigma(i))} > T$ 을 조사하고, 삽입 시와 같은 방법으로 영역 $A^{(n)}$ 에 포함된 정점의 좌표 $V_i^{(n)}(x, y)$ 와 기준좌표 $E_i(x, y)$ 그리고 x' , y' 를 구해 정점의 좌표가 위치하고 있는 장소가 기준좌표를 기점으로 해서 좌표마스크에서 정해진 워터마크 정보 0, 1의 영역의 어디에 속하는 것인가를 조사하고 1비트의 워터마크 정보를 읽어낸다.

Step 3 : 영역 $A^{(n)}$ 내의 모든 정점의 좌표에서 읽어낸 각각의 1비트의 워터마크 정보에 따라서 다수결을 취해, 이것을 영역 $A^{(n)}$ 에서 읽어낸 1비트의 워터마크 정보로 한다.

Step 4 : 모든 영역에 따라서 Step 3의 처리를 행하고, 각 영역 내에서 읽어진 1비트의 워터마크정보를 결합해 읽어낸 비트열 $\{S_{\eta(n)}\}$ 을 생성한다.

Step 5 : 32비트의 키 정보 K에서 생성된 비트 길이 N의 의사난수열 $\{\eta(n)\}$ 으로 읽어낸 비트열 $\{S_{\eta(n)}\}$ 과의 배타적 논리화를 취하는 것으로 n비트 길이에서 확장된 상태의 읽어낸 비트열 $\{S_{\eta(n)}\}$ 을 구한다.

Step 6 : 얻어진 $\{S_{\eta(n)}\}$ 을 그림 1과 같이 다수결을 취해 추출된 워터마크 정보를 얻는다.

2.3 실험방법 및 결과

실험에 사용된 벡터 데이터(이미지)는 PostScript를 이용하였으며, 블록 수 $A^{(n)}$ 은 50개이다. 그림 2는 원본 이미지이며, 그림 3은 워터마크를 삽입한 이미지를 나타낸다. 삽입한 워터마크는 바이너리 [1 0 0 0 1]이며, 따라서 확장율은 10이다. 그림 4는 노이즈 공격을 가했을 때의 이미지를 나타낸다. 사용된 노이즈와 PSNR은 다음과 같다.

$$d = \text{Noise Amplitude}, \text{rand} = \{-1, 0, 1\}$$

$$V_x^d = V_x + d \times \text{rand} \quad (5)$$

$$V_y^d = V_y + d \times \text{rand} \quad (6)$$

$$\text{PSNR} = 20 \log_{10} \left(\frac{\text{MAX}(V_{x,y})}{\text{RMSE}} \right) \quad (7)$$

$$\text{RMSE} = \sqrt{\sum ((V_{x,y} - V_{x,y}^d)^2 / (V_{x,y})^2)} \quad (8)$$

노이즈를 가하지 않았을 경우 삽입한 워터마크 [1 0 0 0 1]는 100% 추출되었으며, 노이즈를 가했을 경우 ($d = 1$) 각 각의 마스크 사이즈와 PSNR, bit 반전율은 표 1과 같다.

Mask Size	Percent of Bit Reversion	PSNR
3 × 3	43.2	48.9413
4 × 4	28.8	51.5337
5 × 5	19.4	50.6527

표 1. 각 마스크 사이즈에 대한 노이즈 Attack과 비트 반전율, PSNR.

그림 2. 원본 이미지.

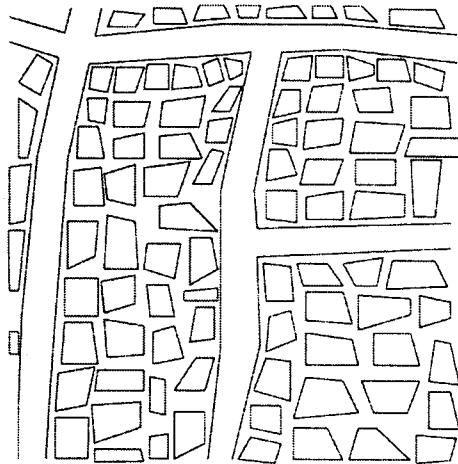


그림 3. 워터마크 삽입된 이미지

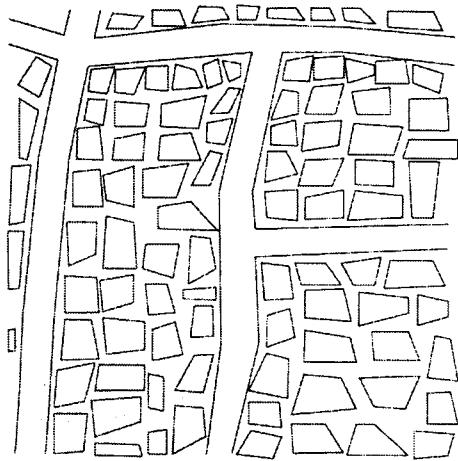
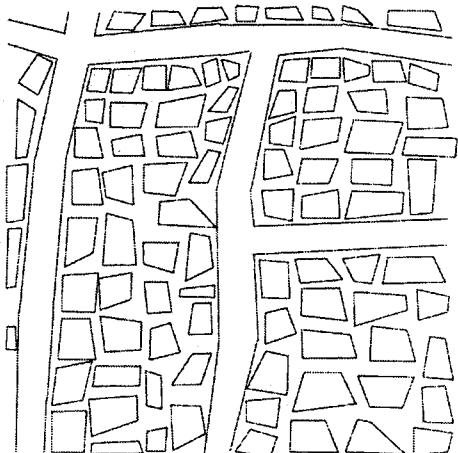


그림 4 노이즈 Attack시 이미지



3. 결 론

본 논문에서는 벡터 데이터(이미지)에 대한 워터마크 삽입과 추출, 그리고 원본 데이터 없이 워터마크를 검출하는 Sakamoto 알고리즘을 발전시킨 자유로운 마스크 사이즈의 사용을 제안했다. 향후, 여러 가지 다양한 벡터 데이터(이미지)에 대한 최적의 마스크 사이즈를 구할 것이며, 노이즈 Attack에 Robust한 알고리즘을 개발할 것이다.

4. 감사의 글

이 논문은 (주)마크애니의 연구비로 수행되었음.

(참 고 문 헌)

- [1] M. Sakamoto, Y. Matsuura & Y. Takashima, "A Scheme of digital watermarking for geographical map data," *Symposium on cryptography & Information security*, Okinama, Japan, Jan. 26-28, 2000.