Edge를 이용한 1-bit 영상의 원본 증명용 워터마킹 알고리즘

박 용 정, 권 오 진
세종대학교 전자공학과
전화 : 02-3408-3828 / 핸드폰 : 011-9072-8866

Watermarking Algorithm for 1-bit Image Authentication using Edges

Yong-Jung Park, Oh-Jin Kwon
Dept. of Electronics Engineering, Sejong University
E-mail : mirusin@image.sejong.ac.kr

Abstract

This paper proposes a new watermarking algorithm to embed data in 1-bit images using edges for protecting illegal copies or modifications. This method is based on edge detection to decide the most invisible region. This paper also introduces a new shuffling method which embeds data in each blocks uniformly and finds positions under the limited attacks. In experiment, we compare the proposed shuffling method with M. Wu's method.

I. 서론

최근 멀티미디어의 발달과 인터넷의 확산으로 인하여 많은 양의 정보가 디지털 형식을 이용해 저장, 전송되어지고 있다. 이에 따라 여러 가지 멀티미디어 데이터에 대한 소유권 문제와 이를 효율적으로 보호할 수 있는 기술이 요구되고 있다. 최근 워터마킹 기술은 멀티미디어 데이터에 대한 소유권을 효과적으로 보호하고, 데이터의 불법 복제 및 배포를 제한할 수 있도록 하는 기술로 각광을 받고 있다.

원 영상에 데이터를 삽입하는 것은 약간의 불필요한 픽셀들을 바꿔줌으로서 행해진다. 칼라 영상의 경우, 가장 단순한 방법은 각 픽셀의 LSB(Least-Significant Bit)를 조작하는 것이다[1]. 이 경우는 영상의 질이 저하되는 것을 거의 느끼지 못하는 반면, 약간의 조작에도 쉽게 삽입된 데이터를 잃을 수 있다.

본 논문에서는 칼라 영상보다는 1-bit 영상에 데이터를 삽입하는 방법에 더 관심을 가진다. 예를 들면, 흰색과 검정색으로만 이루어진 애니메이션 영상, 문서 또는 바코드 등이 1-bit 영상에 속한다. 이러한 영상들은 칼라 영상에 비해 약간의 변화만 주어도 쉽게 눈에 띄기 때문에 데이터를 삽입하기가 더 어렵다.

우선, 1-bit 영상에 데이터를 삽입하는 기존의 방법에는 M. Wu의 알고리즘이 있다[2]. M. Wu의 알고리즘에서는 발생 가능한 3×3 또는 5×5 블록의 패턴에 대한 smoothness와 connectivity를 계산함으로써 각 픽셀의 변환될 우선 순위를 Look Up Table(LUT)로 미리 만들고, LUT을 이용해서 블록마다 가장 우선 순위가 높은 픽셀을 변환하는 방법을 제시했다. 이 방법은 LUT을 만드는 시간이 많이 드는 것이 단점이다. 또 다른 방법으로는 암호 키 K와 weighting matrix W를 사용하는 것이다[3]. 이 방법은 원 영상을 서브블록 F로 나눠주고, F를 같은 크기의 K와 W를 가지고 exclusive-OR 연산과 pairwise multiplication 연산을 행한다. 연산에 의해 나온 결과들을 모두 더하고, weight difference를 구함으로써 데이터를 삽입하는 방

법이다. 앞에 소개한 방법들과는 조금 다른 word-shift 방법도 있다[4]. 이 방법은 연속적으로 쓰여진 단어들 중에서 몇 개만을 골라 오른쪽으로 이동시켜준 후, 이것을 처음 위치에 있던 단어들과 더해준다. 이렇게 해주면, 데이터가 삽입된 단어들은 다른 단어들보다 진하게 보일 것이다. 이 방법은 쉽게 데이터를 삽입할 수 있고, 주로 문서에 사용된다.

지금까지 기존의 방법들을 살펴보았다. 본 논문에서는 위에서 살펴본 방법들처럼 1-bit 영상에 데이터를 삽입하는 새로운 알고리즘을 소개한다. Ⅱ장에서는 새로운 알고리즘을 소개하고, Ⅲ장에서는 본 논문에서 제안한 알고리즘을 이용한 데이터 삽입의 결과 영상과 M. Wu의 결과를 비교 제시하고, 결론은 Ⅳ장에 서술한다.

II. 제안된 방법

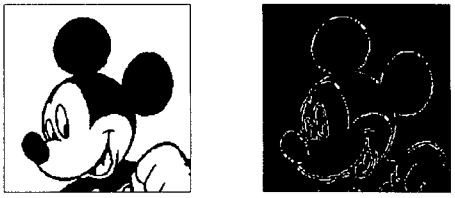
1-bit 영상에 데이터를 삽입하기 위한 기본적인 방법은 일정한 규칙에 입각하여 흰색은 검정색으로, 검정색은 흰색으로 경우에 따라 픽셀을 변환함으로써 이루어진다. 이 방법은 칼라나 흑백 영상이 아닌 1-bit 영상에 대한 것이므로, 최대한 눈에 띄지 않는 영역을 선택하는 것이 가장 중요한 핵심이다.

이 장에서는 데이터를 삽입할 영역을 결정하는 방법과 삽입하는 과정을 서술한다. 또한, 새로운 shuffling 방법을 이용하여 블록별로 데이터를 고르게 삽입하고, 이 방법이 가지는 장점에 대해서 설명한다.

2.1 Edge 추출을 이용한 삽입 영역 결정

1-bit 영상에서 임의의 픽셀들을 변환해주면 쉽게 눈에 띈다. 예를 들면, 수평으로 같은 픽셀 값을 가지는 3×3 블록을 가정했을 때, 그 중에서 가운데 픽셀을 바꿔주면 그 변화가 두드러지게 나타난다.

본 논문에서는 위에서 언급한 것 이외의 변화들을 최소한 줄이면서, 눈에 띄지 않는 영역을 결정하기 위해 edge 추출을 이용한다. Edge는 영상에서의 윤곽선이나 다른 명암을 가진 두 영역사이의 경계를 말한다. 그림 1(b)는 그림 1(a)인 원 영상의 edge를 추출한 것이다. 그림 1(b)를 보면, edge가 가느다란 선으로 원 영상의 윤곽선을 나타내고 있는 것을 볼 수 있다. 여기서, edge에 해당하는 픽셀들이 데이터를 삽입하고자 하는 영역이 된다. 왜냐하면, 윤곽선에 해당하는 픽셀들의 변화는 다른 임의의 픽셀들을 변환해주는 것보다 훨씬 눈에 띄지 않기 때문이다. 또한, edge에 해당하는 픽셀들 중 일정한 수의 픽셀들을 변환하는 것이므로 데이터를 삽입한 영상은 시각적으로 원 영상과 거의 구분이 되지 않는다. 이 방법은 복잡한 계산을 하지 않고도 변환될 픽셀들을 쉽게 결정할 수 있고, 빠르다는 장점이 있다. 본 논문에서는 canny의 edge 추출을 사용했다[5].



(a) 원 영상 (b) Edge 추출 영상

그림 1 데이터를 삽입하기 위한 영역 결정

2.2 데이터를 삽입하는 방법

데이터를 삽입하기 위해서는 원본 증명 과정에서 원 영상을 사용하지 않는다는 점을 고려해야 한다. 그러기 위해서는 일정한 규칙에 의해서 데이터를 삽입해주어야 한다.

본 논문에서는 shuffling 과정을 거친 $N\times N$ 영상을 $M\times M$ 서브 블록으로 나눠주고, $M\times M$ 서브 블록마다 오직 한 픽셀을 변환하는 방식을 취한다. 블록의 크기가 작아질수록 더 많은 양의 데이터를 삽입할 수 있는 반면, 영상이 많이 손상된다. 반대로, 블록의 크기가 커질수록 삽입할 수 있는 데이터의 양은 줄어드는 반면, 영상은 거의 손상되지 않는다. 그러므로, 블록의 크기는 삽입할 데이터의 양과 영상의 질을 고려하여 결정하는 것이 중요하다. shuffling 방법은 2.3 절에서 자세히 다룬다.

각 서브 블록마다 데이터를 삽입하기 위해 우선 각 서브 블록에 해당하는 픽셀들 중 검정색 픽셀들의 수를 구한다. 그 다음, 검정색 픽셀들의 수가 짝수가 되도록 데이터를 변환해주면 된다. 다시 말하면, 각 서브 블록에서 검정색 픽셀들의 수가 짝수이면 데이터를 삽입하지 않고 넘어간다. 반대로, 홀수이면 검정색을 흰색으로 또는 흰색을 검정색으로 변환시켜 데이터를 삽입한다. 이 방법은 원본 증명 과정에서 원 영상을 사용하지 않고, 각 서브 블록에서 검정색 픽셀들의 수를 계산함으로서 원본인지 아닌지를 가려낼 수가 있다.

데이터를 삽입하는 기본적인 과정은 다음과 같다.

1) $N\times N$ 원 영상의 edge를 추출한다.
2) 원 영상과 edge 추출 영상을 shuffling 한다.
3) Shuffling된 원 영상과 edge 추출 영상을 $M\times M$ 서브 블록으로 나눠준다.

4) 원 영상의 각 $M\times M$ 서브 블록마다 검정색 픽셀의 수를 계산한다.
5) 검정색 픽셀의 수가 짝수이면 다음 서브 블록으로 넘어가고, 홀수이면 edge에 해당하는 검정색 픽셀들 중 한 픽셀을 랜덤하게 결정한 후 변환한다.
6) 데이터 삽입이 끝난 영상을 역 shuffling 하면 삽입 영상을 얻을 수 있다.

원본을 증명하는 기본적인 과정은 다음과 같다.

1) $N\times N$ 실험 영상을 shuffling 한다.
2) 각각의 $M\times M$ 서브 블록마다 검정색 픽셀의 수를 확인한다.
3) 각 서브 블록마다 검정색 픽셀의 수가 모두 짝수이면 원본이고, 하나의 서브 블록에서라도 검정색 픽셀의 수가 홀수이면 원본이 아니다.

위의 과정을 그림 2 에서 나타내었다.

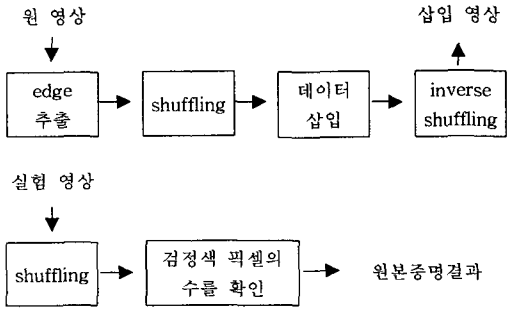


그림 2 데이터 삽입과 원본 증명 과정

2.3 새로운 shuffling 방법

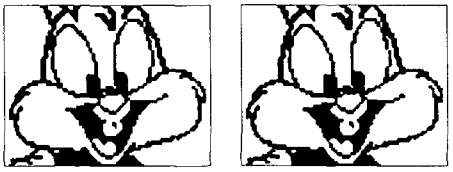
앞 절에서 설명했듯이, 데이터를 삽입하기 전에 shuffling 과정을 거친 후, $N\times N$ 영상을 $M\times M$ 크기의 서브 블록으로 나눠준다. shuffling은 랜덤 넘버를 가지고 각 픽셀의 위치를 섞어주는 방법이다. 여기서는 변환된 픽셀들이 한 블록에만 모이지 않고, 각 서브 블록에 골고루 퍼지게 해주는 역할을 한다.

본 논문에서는 기존의 shuffling 방법보다 보완된 새로운 shuffling 방법을 소개한다. 우선, 원 영상과 결과 영상의 크기를 $N\times N$ 이라고 하면, 원 영상은 $N/M\times N/M$, 결과 영상은 $M\times M$ 서브 블록으로 나눈다. 원 영상의 서브 블록과 결과 영상의 서브 블록을 $A_{x,y}$ $[x,y=0,1,\cdots N/M-1]$, $B_{i,j}$ $[i,j=0,1,\cdots M-1]$ 라고 하면, $A_{0,0}(0,0)$은 첫 번째 서브 블록의 첫 픽셀이다. 여기서, $A_{0,0}(0,0)$는 $B_{0,0}(0,0)$로, $A_{0,0}(0,1)$는 $B_{0,1}(0,0)$로 옮겨주면 된다. 다시 말하면, $A_{0,0}(0,k)$는 $B_{0,k}(0,0)$ $[k=0,1,2,\cdots N/M-1]$ 로 옮겨주면 된다. 나머지 픽셀들도 이와 같은 방법을 사용해서 픽셀들의 위치를 섞어준다. 그러나 이것은 너무 단순하고 동일한 패턴을 가지므로 각 서브 블록마다 $N/M\times N/M$의 랜덤 넘버를 생성하고, $A_{0,0}(rx,ry)$를 $B_{0,0}(0,0)$로 옮겨준다. rx,ry는 랜덤 넘버에 의해 결정된다. 이 방법은 M. Wu의 알고리즘에 비해 작은 블록 내에서 공격을 받았을 때 바뀐 픽셀의 위치를 찾을 수 있다는 장점을 지닌다. 단, 작은 블록은 $N\times N$ 영상을 $N/M\times N/M$ 으로 나눠준 각 서브 블록을 말한다. 공격받은 영상을 위의 방법을 이용해 shuffling 하면, 각 블록들 중에서 공격받은 픽셀을 포함하는 블록을 찾을 수 있다. 공격받은 블록의 모든 픽셀들을 특정한 값으로 표시하고 다시 역 shuffling 하면 일정 영역에서 이들이 모이게 되는 현상을 보이게 되는데 이를 이용하여 공격받은 부분을 찾을 수 있다.

III. 실험 결과

본 장에서는 II장에서 제안한 방법에 대한 결과를 보여주고, M. Wu가 제시한 shuffling 방법과 본 논문에서 제시한 shuffling 방법의 결과를 비교한다.

이 실험에서 사용된 영상은 1-bit 애니메이션 영상과 문서 영상이다. 그림 3, 4 가 1-bit 애니메이션 영상과 문서 영상에 데이터를 삽입한 각각의 결과이다. 그림 3 은 실제 크기가 300×500 인 애니메이션 영상에 대하여 실험한 결과의 일부분을 확대시킨 것이고, 그림 4 는 300×100 인 문서 영상을 실험한 결과이다. 실험한 결과 영상에서도 알 수 있듯이 원 영상과 삽입 영상은 거의 차이가 나지 않는다. 이처럼, edge 추출을 이용하면 II장에서 설명했듯이 눈에 잘 띄지 않는 영역에 데이터를 삽입하는 것이 가능하다.



(a) 원 영상 (b) 삽입 영상

그림 3 캐릭터 영상에 데이터 삽입

It is also desirable has been embedded

(a) 원 영상

It is also desirable has been embedded

(b) 삽입 영상

그림 4 문서 영상에 데이터 삽입

또한, 본 논문에서 제시한 새로운 shuffling 방법은 작은 블록 내에서 공격을 받았을 때 그 위치를 정확히 찾을 수 있는 장점을 지닌다고 Ⅱ장에서 소개했다. 새로운 shuffling 방법의 결과를 M. Wu의 방법과 비교하여 그림 5 에 나타내었다. 그림 5(a)는 M. Wu 가 제시한 shuffling 방법의 결과이고, 그림 5(b)는 본 논문에서 제시한 shuffling 방법의 결과이다. 그림에서 차이를 더 잘 나타내기 위해 박스로 조작된 부분을 표시하였다. 영상에 약간의 조작을 가한 뒤 실험을 해본 결과, 새로운 shuffling 방법과 M. Wu의 shuffling 방법 모두 조작된 픽셀들이 일정 영역에서 모이게 되는 현상을 보인다. 그러나, M. Wu가 제시한 shuffling 방법은 그림 5(a)와 같이 노이즈가 첨가된 반면, 새롭게 제시한 shuffling 방법은 그림 5(b)와 같이 정확하게 조작된 부분을 찾아낼 수가 있다. 왜냐하면, M. Wu의 shuffling 방법은 단순히 랜덤 넘버를 사용해서 픽셀들을 섞어주었기 때문에 각 서브 블록들이 독립적이지 않은데 반해, 새로운 shuffling 방법은 블록별로 픽셀들을 일정하게 섞어주었기 때문에 각 서브 블록들이 독립적이다. 그러므로, 본 논문에서 제시한 shuffling 방법은 M. Wu의 shuffling 방법과 비교해서 작은 블록의 공격에서 위치 추적이 가능하다는 장점을 지닌다.

(a) M. Wu의 shuffling 방법 (b) 새로운 shuffling 방법

그림 5 각 shuffling 방법을 이용해 조작 위치를 나타낸 결과

Ⅳ. 결론

본 논문에서는 1-bit로 이루어진 문서나 애니메이션 영상을 전제로 하는 원본 증명용 워터마킹 방법을 소개했다. 데이터 삽입 영역을 결정하기 위해서 edge 추출을 이용한 간단하면서도 쉬운 방법을 사용하였고, M. Wu의 shuffling 방법과 비교해서 공격받은 영역의 위치추적이 가능한 새로운 shuffling방법을 제시했다.

향후에는 제한된 서브 블록에서의 공격이 아닌 전체적인 공격에서도 조작된 위치추적이 가능한 알고리즘을 개발하는 연구가 요구된다.

참고문헌

[1] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark" In *IEEE Int, Conf. Image Processing*, volume 2, pp 86-90, 1994.

[2] M. Wu, E. Tang, B. Liu. "Data hiding in digital binary image", *IEEE Inter. Conf. on Multimedia & Expo (ICME'00), New York City*, 2000

[3] Yu-Yuan Chen, Hsiang-Kuang, and Yu-Chee Tseng, "A Secure Data Hiding Scheme for Two-Color Images", in *IEEE Symp. on Computers and Communications*, 2000.

[4] J. Brassil, S. Low, N. Maxemchuk, L. O'Gorman, "Hiding Information in Document Images," *Proceedings of the 29th Annual Conference on Information Sciences and Systems*, Johns Hopkins University, pp 482-489, March 1995.

[5] J. Canny. "A computational Approach to Edge Detection" *IEEE Transaction of Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, pp.34-43, 1986.