

디지털 콘텐츠의 저작권 보호를 위한 로고 검출 알고리즘에 관한 연구

최준기*

A Study on a Logo Extraction Algorithm for Copyright Protection of Digital Contents

Joon Kee Choi *

요약

최근들어 인터넷상에서 디지털 콘텐츠의 보안 방법은 절실히 지고 있는 상황이다. 그 방법으로 가시적인 워터마크를 콘텐츠에 삽입한 경우가 많은데, 이는 원래의 콘텐츠를 훼손시킬 뿐만 아니라 워터마크를 손쉽게 제거가 가능하다.

본 논문에서는 로고 영상을 비가시적인 워터마크로 삽입하고, 이를 검출하는 알고리즘을 제안한다. 기존의 연구들은 이에 대해 뚜렷이 제시한 논문이 없고 이의 필요성이 절실한 상황이다. 실험 결과 제안한 알고리즘이 타당함을 증명하였으며, 향후 이용 가치가 충분히 있는 것으로 판명되었다.

Abstract

Recently, on the internet, we need the protection methods of digital contents. To protect web contents, we inserted a visible watermark but it could be removed easily and covered some place of original contents.

In this paper, we proposed a logo detection algorithm that it insert an invisible logo watermark and extract after attacks. Previous works, it didn't show clearly.

Experimental results show that it can be extracted all logo images after an attack. And it is proved that proposed scheme is an appropriate algorithm.

▶ Keywords : 디지털 워터마킹, 이산 웨이블릿 변환, 로고 추출 알고리즘

* 인덕대학 여성정보행정과 조교수

알고리즘을 적용하여 공격을 당했을 때 어느 정도 로고 영상이 검출될 수 있는지를 실험해보고자 한다.

2장에서는 제안한 로고 검출 알고리즘에 대해서 자세하게 논하고, 3장에서는 실험 방법 및 결과를 보이며 마지막으로 4장에서는 결론을 맺는다.

1. 서론

인터넷 이용이 일반화된 지금 많은 정보와 지식을 웹상에서 얻고 있다. 따라서, 웹상에서의 비즈니스가 활성화되고 있고, 정보를 공유하기 위해 수많은 콘텐츠를 확보하여 온라인상에 배포하고 있다. 특히 사진, 동영상 그리고 사운드 등의 디지털화 된 콘텐츠를 공유하기 위한 노력들을 많이 하고 있으나, 이의 소유권 문제가 대두되고 있다.

불법적인 배포 방지를 위해 그 동안은 콘텐츠를 훼손하는 방법을 취해왔다. 즉, 콘텐츠 상에 시각적인 로고 영상을 삽입하여 원 콘텐츠의 일부가 가려지는 모양이었다. 이것은 고품질의 QoS(Quality of Service)를 유지하기 어렵고, 시각적인 로고 영상을 삭제함으로써 불법적인 용도로의 사용이 용이하다.

그래서, 비가시적인 워터마킹(Watermarking) 기술이 연구·개발되고 있는데, 이것은 콘텐츠의 질을 변형시키지 않을 뿐만 아니라 불법 유통을 최대한 억제할 수 있는 장점이 있다[1, 2, 3].

비가시적인 콘텐츠 보호 방법에는 DCT(Discrete Cosine Transform)의 고전적인 방법에서부터 DWT(Discrete Wavelet Transform) 방법 그리고 DRM(Digital Right Management) 방법들이 최근 이용되고 있다[4, 5, 6].

이것들은 주로 워터마크를 삽입 또는 검출하는 알고리즘들이며, DRM의 경우는 암호화 기법을 사용하여 콘텐츠를 포장하는 기술이다.

이러한 알고리즘들을 이용하여 콘텐츠에 비가시적인 워터마크를 삽입하게 되는데, 워터마크 삽입 시 일정한 패턴의 노이즈를 삽입하게 된다. 일정한 패턴의 노이즈는 로고 영상일 수도 있고 랜덤하게 발생시킨 불규칙 노이즈일 수도 있다. 초기에는 랜덤 노이즈를 가지고 워터마크 시퀀스로 많이 이용하였으나, 이 후 심볼(Symbol), 로고(Logo) 등의 상징 마크를 많이 이용하려는 추세에 있다.

본 논문에서는 로고 영상을 가지고 워터마크 시퀀스를 생성하여 이를 비가시성을 만족하도록 삽입하고, 다시 이를 검출하는 알고리즘을 제안하고자 한다. 기존에 상징 마크를 삽입한 후 이를 다시 추출하는 방법을 명확히 제시한 연구들이 없어 본 연구를 통해 이를 제시하고자 하며, 제시한

II. 제안한 로고 검출 알고리즘

1. 전체적인 시스템 구조

본 논문에서 제안한 전반적인 시스템의 구조는 다음 <그림 1>과 같다.

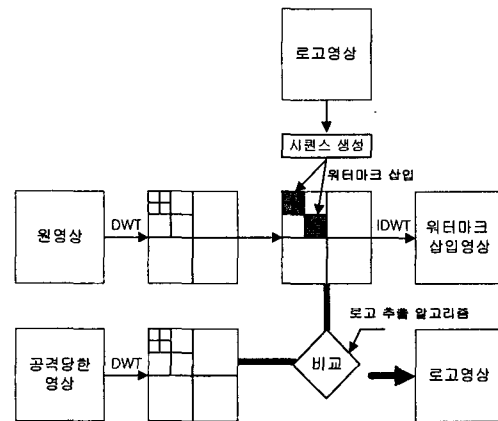


그림 1. 전반적인 시스템 구조
Fig. 1 Overall System Configuration

2. 워터마크 시퀀스 생성

워터마크는 256×256 그레이 로고 영상을 가지고 시퀀스를 생성하는데 다음 <그림 2>의 방법으로 시퀀스를 생성한다. <그림 2>에서 dat_Ori_logo는 원 로고 영상을 의미하며, dat_logo는 이를 실수로 변환한 값을 의미한다.

워터마크 시퀀스는 그레이 로고 영상을 읽어 들여 이를 실수 값으로 변환 후 255로 나누어 줌으로써 생성한다. 255 값으로 나누어주는 이유는 그레이 값이 0~255의 값을 갖기 때문에 나누어준 값이 최고 1을 넘지 않도록 하기 위함이다. 만일 시퀀스가 너무 크면 워터마크가 삽입된 영상이 훼손되기 때문이다.

```

unsigned char *imsi = (unsigned char *)
new char(m_x);
for(y=0; y < m_y; y++)
{
    fread(imsi, sizeof(unsigned char), m_x, in);
    for(x=0; x < m_x; x++)
    {
        *(* (dat_Ori_logo+y)+x) = imsi[x];
        *(* (dat_logo+y)+x) =
(float)(* (* (dat_Ori_logo+y)+x));

        fprintf(logo_value,"%f\n",* (* (dat_logo+y)+x)/255);
        // 시퀀스 생성
    }
}
    
```

그림 2. 워터마크 시퀀스 생성 과정
Fig. 2 A process of watermark sequence generation

3. DWT 알고리즘

웨이블릿 변환 알고리즘의 기본적인 개념은 다해상도 분석(Multiresolution Analysis)과 대역분할 부호화를 하나로 통합한 변환방식으로, 이들 부 대역들은 각 대역의 평균 신호와 각 방향을 세부적인 신호로 표현한다. 즉, 저주파 성분(LL), 수평성분(LH), 수직성분(HL), 대각 성분(HH)으로 이루어지며, 시간과 주파수영역에서 동시에 국부성(Localization)을 가지고 신호를 표현하므로 저주파 성분이 많은 일반 영상 신호의 분석에 활용할 수 있다[7].

웨이블릿 변환은 다음 식 (1)로 다중해상도 분석이 가능하다. 식 (1)에서 a는 스케일을 조정하는 성분이고, b는 시간 축을 따라 이동하는 성분을 나타낸다. $\frac{1}{\sqrt{a}}$ 는 정규화를 위한 상수이다. 원형 웨이브릿함수 ϕ 는 주기성을 가지고 있고 그 주기성은 시간 축을 따라 이동하면서 영(Zero)으로 감소하는 성질을 가지고 있어야 한다.

$$\phi^{a,b}(x) = \frac{1}{\sqrt{|a|}} \phi\left(\frac{x-b}{a}\right) \dots\dots\dots (1)$$

각 대역들에 필터링을 하여 성분들을 추출하는데 도버치의 이산 웨이블릿 필터[8]를 많이 이용하고 다음 식 (2)와 식 (3)에 의해 그 과정이 이루어진다.

$$\tilde{g}_n = (-1)^n h_{-n+1} \dots\dots\dots (2)$$

$$g_n = (-1)^n \tilde{h}_{-n+1} \dots\dots\dots (3)$$

여기서, h_n : 분해필터의 저주파 필터, \tilde{h}_n : 복구필터의 저주파 필터

4. 워터마크 삽입 방법

워터마크 삽입 방법은 먼저 생성된 워터마크 시퀀스에서 0이 아닌 값을 갖는 시퀀스들을 선택하여, 이를 적절한 웨이블릿 대역에 삽입하게 된다. 3-레벨일 때 워터마크 삽입 대역은 다음 <그림 3>과 같다.

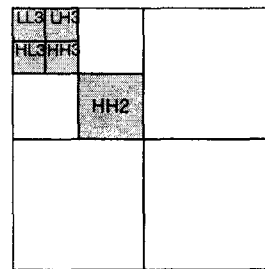


그림 3. 워터마크 삽입대역
Fig. 3 Watermark insertion subband

워터마크 삽입 대역을 <그림 3>과 같이 정한 것은 이미 다양한 실험을 통해 증명한다 있다[9]. 그리고, 가중치 등의 가변 값도 이미 실험을 통해 증명된 값들을 이용한다.

워터마크 삽입 과정은 다음 <그림 4>와 같다. <그림 4>에서 Ori_dat는 원 영상 값 즉, 본 논문에서 사용한 256x256 크기의 레나 영상 값을 의미하며, Water_arr[j]는 워터마크 시퀀스를 1차원 배열로 저장한 값이다. 그리고 배열 Index[k][k1]은 이후 로고 검출 시 워터마크 삽입 지점을 찾기 위해서이다. 즉, 이 값이 1인 배열은 그 지점에서 워터마크가 삽입되었음을 의미한다.

```

for ( k = 0 ; k < m_y ; k ++){ // 0이 아닌 값들만 추출
for (k1 = 0 ; k1 < m_x ; k1++){
    fscanff(in,"%f\n",&Temp_arr(k)(k1));
    if(Temp_arr(k)(k1) > 0){
        Water_arr(i) = Temp_arr(k)(k1);
        i++;
    }
}
}
for ( k = 0 ; k < m_y ; k ++){ // 워터마크 삽입
for (k1 = 0 ; k1 < m_x ; k1++){
    if((( k1 >= 0 ) && ( k1 < 64 )) && (( k ) = 0 )
    && ( k < 64 )){ //LL3,HL3,LH3,HH3
        j=j+1;
    }
}
}
    
```

```

        *(w_dat + k) + k1) = *(Ori_dat + k) +
        k1)*(1 + constraints_b * Water_arr(j));
        Index(k)(k1)=1;
    }
    else if((( k1 > 63 ) && ( k1 < 128 )) && (( k > 63
    ) && ( k < 128 ))) //HH2
        j=j+1;
        *(w_dat + k) + k1) = *(Ori_dat + k)
        k1)*(1 + constraints_a * Water_arr(j));
        Index(k)(k1)=1;
    }
    else{
        j=j+1;
        *(w_dat + k) + k1) = *(Ori_dat + k) +
        k1);
        Index(k)(k1)=0;
    }
}
}

```

그림 4. 워터마크 삽입 과정
Fig. 4 Process of watermark insertion

5. 로고 검출 알고리즘

워터마크 시퀀스로 사용했던 로고를 다시 검출하는 알고리즘은 다음 (그림 5)와 같이 구현된다.

```

for ( k = 0 ; k < m_y ; k ++){
    for (k1 = 0 ; k1 < m_x ; k1++){
        if(원영상값 < 0){
            양수값으로 변환;}
        if(공격당한 영상값 < 0){
            양수값으로 변환;}
    }
}
for ( k = 0 ; k < m_y ; k ++){
    for (k1 = 0 ; k1 < m_x ; k1++){
        if(워터마크 삽입 화소에서 원영상값이 공격당한 영상값보
        다 크면){
            Ratio(k)(k1)=(1-공격당한 영상값/원영상값);
            Seq_Ratio(k)(k1) = 워터마크 시퀀스*Ratio(k)(k1);
            Last_Seq(k)(k1)
                = 워터마크 시퀀스 - Seq_Ratio(k)(k1);
            if(Last_Seq(k)(k1) < 0){
                Last_Seq(k)(k1) = Last_Seq(k)(k1) * (-1);
            }
            For_the_Last(count) = Last_Seq(k)(k1);
            count++;
        }
        else if(워터마크 삽입 화소에서 공격당한 영상값이 원영상
        값보다 크면){
            Ratio(k)(k1)=(1-원영상값/공격당한 영상값);
            Seq_Ratio(k)(k1) = 워터마크 시퀀스*Ratio(k)(k1);
            Last_Seq(k)(k1)
                = 워터마크 시퀀스 - Seq_Ratio(k)(k1);
        }
    }
}

```

```

if(Last_Seq(k)(k1) < 0){
    Last_Seq(k)(k1) = Last_Seq(k)(k1) * (-1);
}
For_the_Last(count) = Last_Seq(k)(k1);
count++;
}
else if(워터마크 삽입 화소에서 원영상과 공격당한 영상의
값이 같으면){
    Ratio(k)(k1)=0;
    Seq_Ratio(k)(k1) = 워터마크 시퀀스*Ratio(k)(k1);
    Last_Seq(k)(k1)
        = 워터마크 시퀀스 - Seq_Ratio(k)(k1);
    For_the_Last(count) = Last_Seq(k)(k1);
    count++;
}
}
}
for ( k = 0 ; k < m_y ; k ++){
    for (k1 = 0 ; k1 < m_x ; k1++){
        if(로고 영상에서 워터마크 시퀀스로 쓰인 값이면){
            Last_Logo_Value(k)(k1)
                = For_the_Last(kk) * *(dat_logo + k) + k1);
            Final(k)(k1)
                = *(dat_logo + k) + k1) -
                Last_Logo_Value(k)(k1);
            kk++;
        }
        else{
            Final(k)(k1) = *(dat_logo + k) + k1);
        }
    }
}
for ( k = 0 ; k < m_y ; k ++){
    for (k1 = 0 ; k1 < m_x ; k1++){
        *(dat3 + k) + k1) = Final(k)(k1);
    }
}
}

```

그림 5. 로고 검출 알고리즘
Fig. 5 Logo extraction algorithm

위의 알고리즘에서 공격 정도에 따른 변화율을 계산해야 하기 때문에 모든 값들은 양수이어야 하며, 그 변화율을 계산하여 로고 영상값(dat_logo)에 비례값을 취하면 복원된 영상(dat3)이 생성된다.

로고 검출 시 공격의 정도에 따른 로고 영상의 훼손 정도를 측정하기 위해 가시성을 판별하는 최대신호대잡음비(PSNR)를 측정하여 비교하였다. 검출된 로고 영상과 이의 PSNR 값은 다음 <표 3>과 같다.

III. 실험 방법 및 결과

실험에 쓰인 영상들은 표 1과 같이 가중치 등의 가변 값들은 <표 2>에서 보인다.

표 1. 실험 영상들
Table. 1 Experimental images

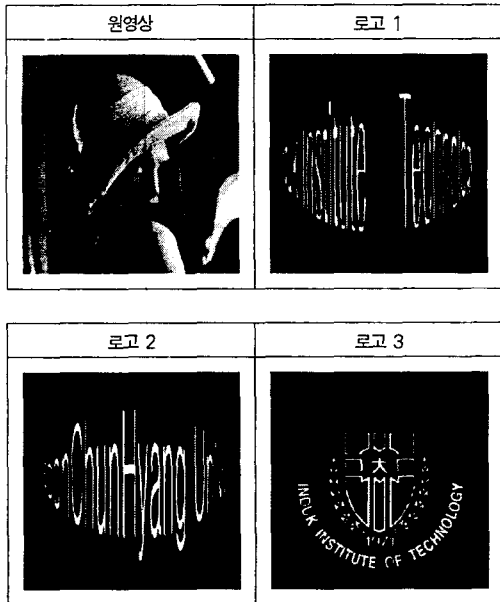
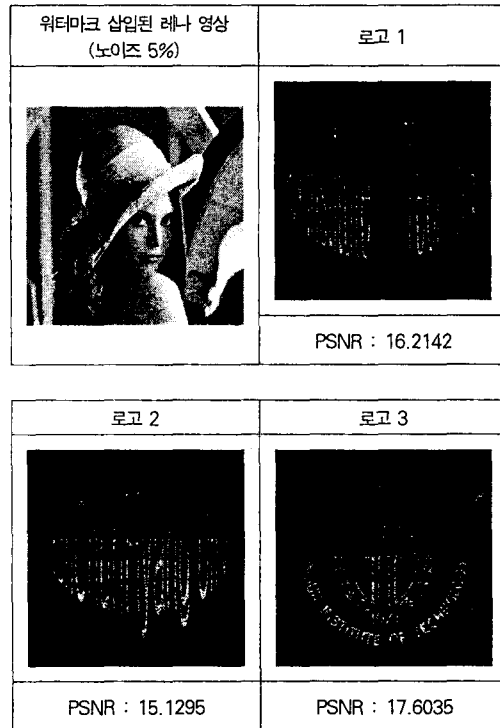


표 2. 가변값들
Table. 2 Variables

가중치(α)	필터	웨이블릿 변환	워터마크 개수
0.02 (2-레벨 대역)	도버지 필터	3-레벨	8000~9000
0.01 (3-레벨 대역)			

실험 방법은 워터마크를 삽입한 영상에 랜덤한 노이즈를 삽입한 후에 이를 공격당한 영상으로 가정하고 실험을 진행하였다.

표 3. 실험 결과
Table. 3 Experimental results



<표 3>에서 PSNR을 측정한 것은 수치적으로는 의미가 없으나, 검출된 영상이 어느 정도 훼손되었는지를 보기 위해서 측정한 값이다. 5% 정도의 노이즈를 추가한 영상에서도 실험 결과에서 보는 것처럼 로고가 확실하게 검출되었음을 알 수 있다. 이것은 어떤 공격 형태에서도 로고가 살아남을 수 있음을 단적으로 증명해 준다.

IV. 결 론

지금까지 디지털 콘텐츠의 저작권 보호를 위한 로고 검출 알고리즘에 관한 연구를 수행하였다. 그레이 로고 영상을 위터마크 시퀀스로 변환하고 원영상에 삽입 후 노이즈를 첨가한 공격 영상에서 이를 다시 검출하는 방법을 제안하였다. 지금까지의 연구에서 그레이 로고 영상은 위터마크로 사용하고, 이를 검출하는 뚜렷한 연구가 없어 본 논문에서 이를 제안하였다. 실험 결과를 보면 제안한 알고리즘이 로고를 잘 검출할 수 있음을 보였으며, 이를 판용할 가치도 있는 것으로 판단된다.

향후에는 다음과 같은 연구를 진행해야 한다. 먼저 공격의 형태 즉, 불법적인 용도로의 사용을 위해 다양하게 영상을 변형시키는 경우에도 로고가 잘 검출될 수 있도록 제안한 알고리즘을 수정·보완해야 하며, 로고 검출 후 영상 보정 작업을 통해 완전한 로고 영상으로 만들 수 있도록 해야 한다.

참고문헌

[1] I. J. Cox, J. Kilian, T. Leighton and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," NEC Research Institute, Technical Report 95-10, 1995.

[2] Hartung, F and M. Kutter, "Multimedia Watermarking Techniques," Proc. of the IEEE, Vol. 87, No. 7, July, 1999.

[3] Lin, E. T and E. J. Delp, "A Review of Fragile Image Watermarks," Proc. of the Multimedia and Security Workshop (ACM Multimedia '99) Multimedia Contents, pp. 25~29, October 1999.

[4] O. Rioul and M. Vetterli, "Wavelets and Signal Processing," IEEE Signal Processing Magazine, pp. 14~38, 1991.

[5] M. Vetterli and J. Kovacevic, Wavelets and Subband Coding, Prentice Hall, Englewood Cliffs, 1995.

[6] P. P. Validyanathan, "Quadrature Mirror Filter Banks, M-band Extensions and Perfect reconstruction Techniques," IEEE ASSP Magazine, 4(3), pp. 4~20, 1987.

[7] X. G. Xia, C. G. Boncelet and G. R. Arce, "A multiresolution watermark for digital images," IEEE International Conference on Image Processing, pp. 548~551, Santa Barbara, CA, Oct. 1997.

[8] O. Rioul and M. Vetterli, "Wavelets and Signal Processing," IEEE Signal Processing Magazine, pp. 14~38, 1991.

[9] 강영창, 최준기, 김선형, "웨이브릿 영역에서의 디지털 위터마킹에 관한 연구," 한국정보과학회 춘계학술발표논문집, Vol. 28, No. 1, pp. 466~468, 2001.

저 자 소 개



최 준 기
 1993. 2 순천향대학교 전산학과 (공학사)
 1995. 2 순천향대학교 전산학과 (공학석사)
 1999. 2 순천향대학교 전산학과 (공학박사)
 1999. 3 ~ 현재 인덕대학 여성 정보행정과 조교수