

프레임간 상대적인 차에 의한 셔트 검출 기법을 이용한 비디오 워터마킹

정인식, 권오진
세종대학교 전자공학과
전화 : 02-3408-3828 / 핸드폰 : 016-396-5843

Video Watermarking Using Shot Detection

In-Shik Jeong, Oh-Jin Kwon
Dept. of Electronics Engineering, Sejong University
E-mail : always@image.sejong.ac.kr

Abstract

This paper proposes a unique data embedding algorithm for the video sequence. It describes two processings: shot boundary detection and robust data embedding. First, for the shot boundary detection, instead of using absolute frame differences, block by block based relative frame differences are employed. Frame adaptive thresholding values are also employed for the better detection. Second, for the robust data embedding, we generate message template and then convolve and correlate it with carrier signal. And then we embed data on the time domain video sequence. By using these two methods, watermarks into randomly selected frames of shots. Watermarks are detected well even if several certain shots are damaged because we embed watermark into each shot equally.

I. 서론

최근 비디오 콘텐츠의 사용이 급격히 증가하고 있다. Video CD는 활성화되었고 가장 최근에는 DVD가 각광을 받고 있다. 현재의 컴퓨터 환경의 발달로 비

디오 콘텐츠의 질이 과거에 비해서 상대적으로 매우 높아졌다. 그러나 불행하게도 이와 같은 발전과 더불어 불법복제 또한 심각한 사회문제를 초래하고 있다. 컴퓨터 환경의 발달로 불법복제 및 도용이 매우 간편해진 것이 지금의 현실이다. 종래에는 이러한 저작권 보호 및 허가된 사용자만이 볼 수 있도록 하기 위해서 비디오 콘텐츠의 일부분에 암호화된 text를 삽입하는 방법이 제시되었지만 이것 또한 완전하게 저작권을 보호하기에는 한계가 있다.

본 논문의 II장에서는 기존의 방법들을 설명하고, III,IV장에서는 제시하는 방법과 그 실험 결과를 설명한다.

II. 지금까지 제시된 방법들

비디오 data embedding은 정지영상과 달리 spatial, frequency domain 외에 time domain까지 확장할 수 있다.

초기에 사용된 방법으로는 '-1'과 '1'로 이루어진 워터마크 bit와 '-1'과 '1'로 이루어진 random number를 서로 곱하고 조절하여 영상의 전 영역에 삽입하는 기법이 있다[1]. 또한 객체기반 비디오 워터마킹도 제안되었다[2]. 이 방법은 영상을 8x8 블록으로 나누고 각 블록에 대하여 DCT를 하고 저자 서명을 DCT하여 8x8 블록에 곱해 주고 미리 구해둔 spatial masking값을 곱한 후, 이전 프레임의 현 위치의 블록과 차에 의

하여 워터마킹 삽입 유무를 결정한다. 앞의 방법들과 달리 MPEG과 같이 압축되어 있는 비디오에서는 연산 시간과 연산량을 줄이기 위해서 각 프레임의 I-frame에 대하여 data를 삽입한다[3]. 또한 *m-sequence*와 *bit-planes*를 이용하는 방법도 있다[4]. 최근에는 MPEG 비디오의 motion vector를 이용하여 data를 삽입하는 기법도 제안되었다[5].

III. 비디오 워터마킹

3.1 서트의 검출

본 논문에서는 기존 알고리즘에 비해 “false positive” 오류와 “false negative” 오류를 현저하게 줄일 수 있는 알고리즘을 사용하였다[6].

서트의 경계를 찾는 알고리즘은 다음과 같다.

- 1) 전체 영상을 MxM 서브 블록으로 분할한다.
- 2) *t*와 *t+1*번째 프레임에서 각 서브 블록의 화소들 간의 차이의 합을 다음 식을 사용하여 구한다.

$$B_t(k) = \sum_{i=0}^M \sum_{j=0}^M (I_t(i, j, k) - I_{t+1}(i, j, k)) \dots\dots\dots(1)$$

여기에서 $I_t(i, j, k)$ 는 *t* 번째 프레임에서 *k* 번째 블록의 (*i, j*)번째 화소이다.

- 3) 2)의 결과인 $B_t(k)$ 로부터 각 프레임별로 다음 식을 사용하여 R_t 를 구한다.

$$R_t = \sum_k B_t(k) \dots\dots\dots(2)$$

- 4) *t*번째 프레임의 전 2개 프레임과 후 2개 프레임에 대하여 같은 연산을 수행하여 다음의 R_{MAX} 를 계산한다.

$$R_{MAX} = \max(R_{t-2}, R_{t-1}, R_{t+1}, R_{t+2}) \dots\dots\dots(3)$$

- 5) 임의 임계값 h_2 에 대하여 $R_t \geq h_2 * R_{MAX}$ 이면 서트의 경계로 결정한다.

- 6) 다른 임계값 h_1 에 대하여 $R_t < h_1 * R_{MAX}$ 이면 서트의 경계가 아닌 것으로 결정한다.

- 7) $h_1 * R_{MAX} \leq R_t < h_2 * R_{MAX}$ 인 경우에는 다음과 같이 인접 프레임 간 히스토그램의 차이를 계산한다.

$$D_t = \sum_j |H_t(j) - H_{t+1}(j)| \dots\dots\dots(4)$$

- 8) $D_{MAX} = \max(D_{t-1}, D_{t+1})$ 값을 계산하고 임의 임계값 h_3 에 대하여 $D_t \geq h_3 * D_{MAX}$ 이면 서트의 경계로 결정하고 그렇지 않으면 서트의 경계가 아닌 것으로 결정한다.

과정 2)의 $B_t(k)$ 계산식에서 화소 값의 차이를 구할 때 절대값을 취하지 않았음에 유의하여야 한다. 이

는 로컬 또는 글로벌 모션이 있는 프레임의 경우 화소 값의 차이가 양수 값과 음수 값을 동시에 블록 내에 갖게 되는 현상이 고려되었다. 과정 5)와 6)에서는 절대적인 임계값을 사용하지 않고 상대적 임계값인

R_{MAX} 를 사용하여 성능을 향상시켰다. 과정 7)은

R_t 만으로는 서트의 경계 여부를 판단하기가 어려운 경우, 추가로 히스토그램의 차이를 적용한다.

3.2 Data Embedding

본 논문에서 사용되는 알고리즘은 Honsinger의 논문을 기반으로 하였다[7].

Honsinger의 논문에서 워터마크 message template $T(x, y)$ 는 message 영상과 carrier 영상의 convolution에 의하여 생성된다. Message 영상 $M(x, y)$ 는 iconic 영상 또는 사용자에게 입력받은 text를 bit화 시켜 이진 영상으로 생성된다. Carrier 영상 $C(x, y)$ 을 생성하는 가장 큰 목적은 message 영상의 에너지를 어떠한 곳에도 집중되지 않고 전체영상에 골고루 분포하도록 하는데 있다. Carrier 영상은 인위적으로 만든 phase와 magnitude의 Inverse Fast Fourier Transform(IFFT)에 의하여 만들어진다. 이때 에너지 편중을 막기 위하여 uniform random number를 생성하여 phase를 만들어 준다. Carrier 영상의 magnitude는 flat spectrum 또는 Contrast Sensitivity Function(CSF)에 의하여 생성된다. Flat spectrum은 Stirmark 공격에 대응하기 위하여 필요하고[8], CSF를 사용할 경우 검출률이 높아지고 비가시성이 높아진다.

Data 삽입의 원리는 다음과 같다.

$$I(x, y) = a(M(x, y) * C(x, y)) + I(x, y) \dots\dots\dots(5)$$

식(5)에서 ‘*’은 cyclic convolution으로 정의하고 a 는 임의의 상수이고, $I(x, y)$ 와 $I(x, y)$ 는 각각 원 영상과 워터마크가 삽입된 영상이다. 식(5)는 다음과 같이 쓸 수 있다.

$$I(x, y) = a(T(x, y)) + I(x, y) \dots\dots\dots(6)$$

식(6)은 message 영상과 carrier 영상을 cyclic convolution에 의하여 message template를 만들고 이것을 원 영상에 삽입하는 방식이다. 삽입과정을 좀더 자세히 살펴보면 다음과 같다.

- 1) 사용자의 key와 text를 입력받아 $C(x, y)$ 와 $M(x, y)$ 을 생성한다. 이때 $C(x, y)$ 와 $M(x, y)$ 의 크기는 임의적으로 조정할 수 있다. 본 논문에서는 128x128로 설정하였다.

- 2) $C(x, y)$ 와 $M(x, y)$ 의 cyclic convolution에 의하여 $T(x, y)$ 을 생성한다.

- 3) $T(x, y)$ 의 값이 $I(x, y)$ 의 값보다 크기 때문에 $T(x, y)$ 의 값을 a 값에 의하여 비례적으로 축소시킨다.
- 4) $I(x, y)$ 을 128x128 블록으로 나눈다. 이것은 $C(x, y), M(x, y), T(x, y)$ 모두 128x128의 크기로 생성되었기 때문이다.

5) 축소된 $T(x, y)$ 을 각각의 블록에 더해준다. 이러한 과정에 의하여 data embedding이 이루어진다. 이때 주의할 점은 convolution 과정이 FFT domain에서 이루어진다.

검출은 삽입의 역 과정으로, 그 원리는 다음과 같다.

$$M(x, y) = I(x, y) \otimes C(x, y) \dots\dots\dots(7)$$

식(7)에서 ‘ \otimes ’은 cyclic correlation으로 정의되고, $M(x, y)$ 은 워터마크가 삽입된 영상에서 추출된 message 영상이다. $C(x, y)$ 의 성질을 보면

$$C(x, y) \otimes C(x, y) = \delta(x, y) \dots\dots\dots(8)$$

여기서 $\delta(x, y)$ 는 Dirac delta function이다. 또한 cyclic convolution 및 cyclic correlation은 교환법칙이 성립하고, 이것을 식(5)(8)과 같이 식(7)에 적용하면

$$M(x, y) = aM(x, y) * \delta(x, y) + I(x, y) \otimes C(x, y) \dots\dots(9)$$

되고, 이상적인 경우 $aM(x, y) * \delta(x, y)$ 는 $aM(x, y)$ 가 되고, $I(x, y) \otimes C(x, y)$ 는 0이 된다.

검출과정을 자세히 살펴보면

- 1) $I(x, y)$ 을 128x128 블록으로 나눈다.
 - 2) 각 블록에서 같은 위치에 있는 화소 값을 모두 더해준다.
 - 3) 사용자로부터 key를 받아 $C(x, y)$ 을 생성한다.
 - 4) 2)의 결과와 3)의 결과를 cyclic correlation한다.
- 이러한 과정에 의하여 사용자가 입력한 text 또는 iconic 영상을 검출할 수 있다. 검출과정도 삽입과정과 같이 cyclic correlation을 FFT domain에서 수행한다.

3.3 비디오 워터마킹

3.1 과 3.2 장에서 서트의 검출 및 data embedding 방법에 대하여 살펴보았다. 이 장에서는 비디오물에 적용하는 방법에 대하여 알아보겠다. 3.1 장에서 찾은 서트의 경계는(그림 1. 1) 안정되지 않은 프레임이므로 일정개수만큼 프레임간의 차이를 구하여 가장 차이가 적은 프레임을 찾는다. 이것은 서트의 경계에서 급격하게 변화 후 안정되는 프레임(그림 1. 2)을 찾는 것이다. 안정된 프레임을 찾고 마지막으로 임의적인 성질과 암호화를 위하여 Message Digest algorithm 5(MD5)를 적용한다. MD5는 입력으로 text 또는 숫자가 들어가면 출력은 32byte가 된다. 각 byte의 상·하

위 4bit는 그림 1.에서 첫 번째와 마지막 3을 가리킨다.

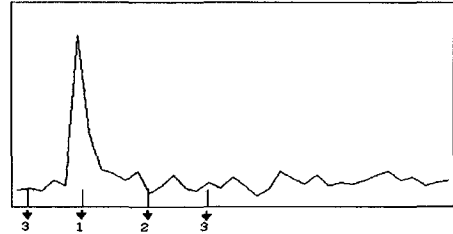


그림 3. 워터마크의 삽입위치 결정

그림 1.에서 첫 번째 3은 서트의 경계에서 MD5의 하위 4bit에 해당하는 숫자만큼 이전 프레임방향의 위치의 프레임이다. 마지막 3은 상위 4bit을 의미하며 안정된 프레임에 더해준다. 이와 같이 한 서트에서 앞·뒤 양쪽 두 곳의 위치에 워터마크가 삽입된다. 특정 프레임의 훼손에 내성을 갖기 위해서 각 위치에서 주위의 여러 프레임에 워터마크를 삽입하는 방법도 가능하다.

IV. 실험 결과

본 장에서는 다양한 종류의 비디오물에 대하여 서트의 경계를 이용하여 워터마크를 삽입하고, 삽입된 data를 정확히 검출할 수 있는지 알아보겠다.

실험에 사용된 비디오물은 s1-6으로 표시하였다. S1은 총 36개의 서트를 가진 심야 토크쇼이다. S2는 상업용 광고물로서 10개의 서트로 구성되어 있으며, 고객의 인공적인 흔들림 화면이 포함되어 있다. S3은 자동차 광고용 비디오로서 14개의 서트로 구성되어 있고 내용이 진행이 매우 빠른 서트의 변화가 있다. S4는 50개의 서트로 구성된 다큐멘터리이다. 비행기 이륙시에 발생하는 연기에 의해서 “false positive” 오류가 발생할 확률이 높은 비디오물이다. S5는 51개의 서트로 구성된 “트루라이즈” 영화 예고편이다. 폭발 및 사격 장면에서 순간적으로 화면이 밝아져 다수의 “false positive” 오류가 발생하는 요인이 된다. S6은 “리셀웨폰4” 영화의 예고편으로 144개의 서트로 구성되어 있다. s6은 전체적인 카메라의 흔들림과 등장인물의 이름이 소개될 때 글씨가 전면으로 순간적으로 확대되었다가 축소되는 효과를 사용하기 때문에 서트를 검출하기가 매우 까다로운 비디오물이다.

블록의 크기인 $M \times M$ 값은 4×4 를 적용하였고, 임계값은 각각 $h_1=1.5$, $h_2=4.0$, $h_3=2.0$ 로 결정되었다.

표 1.에서 “total No. of frame”은 전체 프레임 개수를 말하고, “recall”은 실제로 존재하는 서트의 수와 이중 제대로 검출된 서트 수의 비율을 표시하고,

“precision” 은 검출한 총 셔트 수와 이중 제대로 검출

| | s1 | s2 | s3 | s4 | s5 | s6 |
|--------------------|------|-----|------|------|------|------|
| Total No. of frame | 7762 | 449 | 452 | 1304 | 1451 | 4037 |
| Recall | 1.0 | 1.0 | 1.0 | 0.97 | 0.9 | 0.99 |
| Precision | 0.97 | 1.0 | 0.92 | 0.96 | 0.95 | 0.94 |

표 1. s1-6에 대한 셔트 검출 결과

한 셔트 수의 비율을 표시한다.

삽입과정에서 key 값으로 “Elec”를 사용하였고, text 로 “Sejong Univ”를 사용하였다.

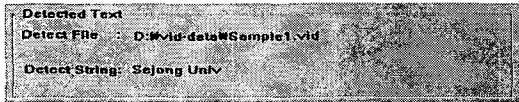


그림 2. 정확하게 검출된 예

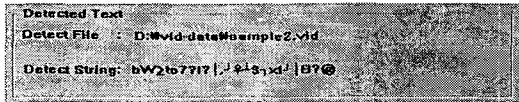


그림 3. 검출에러가 발생한 예

그림 2.를 보면 “Sejong Univ”가 정확히 검출되었다. S1, s4, s5, s6 비디오물이 이에 해당된다. 그림 3.은 dummy data가 검출된 예이다. S2, s3가 이에 해당된다. 본 장에서 앞부분에서 밝힌 각 비디오물의 셔트의 개수를 보면 s1-6 각각 36, 10, 14, 50, 51, 144개로 이루어져 있다. 본 논문에서 제안한 기법은 셔트가 적은 비디오물보다는 대략 30셔트 이상의 비디오물에 정확하게 검출이 된다. 이것은 정지영상의 경우 128x128블록으로 나뉘었을 때, 유사한 부분이 다수 존재하지만 비디오물은 기본적으로 영상의 크기가 작기 때문에 128x128블록으로 나누면 대부분의 블록이 상이한 부분을 많이 포함한다. 그러므로 많은 셔트에 워터마크를 삽입해야 검출이 가능하다. S2, s3은 셔트가 30개 보다 적기 때문에 정확하게 검출이 되지 않았다.

V. 결론

본 논문에서는 비디오 워터마크의 새로운 방법을 제시하였다. 먼저 셔트를 검출한다. 이것은 1) 화소간의 차이를 이용하는 방법을 블록단위로 적용하고, 2) 화소간의 차를 구할 때 절대적인 차이 대신에 상대적인 차이를 사용하고, 3) 프레임 별로 가변적인 임계값을 사용하였으며, 4) 화소간의 차이만으로는 셔트의 경계 검출이 어려운 경우 히스토그램의 차이를 선별적으로 적용시킴으로써 “false positive”를 상당부분 제거한다. 본 논문에서는 1)2)3)4)번에 의하여 결정된 셔트의

경계를 이용하여 워터마크를 삽입한다. 이때 워터마크를 안정된 프레임에 삽입하기 위하여 인접한 프레임간 차이가 가장 적은 프레임을 찾는다. 또한 MD5를 적용하여 삽입될 위치를 임의적으로 선택하고 암호화시킨다. 이러한 방법은 비디오물의 전 프레임에 워터마크를 삽입하는 것 보다 시간을 줄일 수 있고, 연속된 프레임에 워터마크가 삽입되는 것이 아니므로 비가시성을 높일 수 있다. 또한 비디오물의 전 프레임에 워터마크를 삽입하는 기법의 장점인 일부분의 훼손이나 제거에 강한 성질도 본 논문에서는 각각의 셔트에 임의적으로 워터마크 삽입하는 방식에 의하여 가질 수 있다.

참고문헌

- [1] F. Hartung and F. Girod, “Fast public-key watermarking of compressed video,” *IEEE Proc. ICIP 97*, pp. 528-531.
- [2] M. D. Swanson et al, “Object-based transparent video watermarking,” *IEEE workshop on Multimedia Signal Processing*, Princeton, pp. 369-374, June. 1997.
- [3] F. Hartung and B. Girod, “Watermarking of uncompressed and compressed video”, *Signal processing*, Vol. 66, pp. 283-301, 1998
- [4] B. G. Bobasseri, “Direct sequence watermarking of digital video using m-frames”, *IEEE Proc. ICIP 98*, Chicago, pp. 4-7, Oct. 1998.
- [5] J. Zhang, H. Maitre, J. Li, L. Zhang, “Embedding watermark in MPEG video sequence,” *IEEE workshop on Multimedia Signal Processing*, pp. 535-540, 2001
- [6] 정인식, 권오진, “프레임간의 상대적인 차이를 이용한 비디오의 셔트 검출 기법”, *대한전자공학회 하계종합학술대회 논문집 - IV*, 제 24권 1호, pp.101-104, 용평리조트(강원도), 2001.6.
- [7] C. Honsinger, “Data embedding using phase dispersion,” *IEE Seminar on Secure Images and Image Authentication*, pp. 5/1-5/7, 2000
- [8] <http://www.cl.cam.ac.uk/users/fapp2/watermarking/stirmark/index.html>