H.264-기반 인트라 프레임의 디지털 워터마킹 문제

최현준* · 서영호** · 김동욱*

The Problems in Digital Watermarking into Intra-Frames of H.264/AVC

Hyun-Jun Choi* · Young-Ho Seo** · Dong-Wook Kim*

본 연구는 한국과학재단 특정기초연구(R01-2006-000-10199-0)지원으로 수행되었음.

요 약

본 논문은 H.264의 인트라 프레임을 대상으로 일반적인 디지털 워터마킹 방법을 목표로 하여 인트라 예측이 워터마킹에 미치는 영향과 일반적인 워터마킹 방법이 H.264의 인트라 프레임에서는 그 효용성이 매우 낮다는 것을 보이고자 한다. 대상 워터마킹 방법으로는 비가시성과 강인성 워터마킹으로써 H.264 압축과정 중에 수행하는 것으로 가정한다. 문제는 워터마크 데이터 추출을 위한 재압축 과정(re-engineering) 중 영상데이터의 계수값이 변화하는 것이며, 이것은 H.264의 인트라 예측 자체의 문제임을 실험적으로 보였다. 즉, 워터마크 데이터를 삽입하지 않은 압축된 데이터를 동일조건으로 재압축 하였을 때 그 결과 데이터가 첫 번째 압축결과와 다르며, 이것은 계수값 자체뿐 만 아니라 예측모드가 변화한다는 것을 보였다. 또한 인트라 프레임에 대한 몇 가지 워터마킹 실험을 통해 전형적인 공격이 이 문제를 더욱 두드러지게 만들어 워터마킹 방법의 효용성을 크게 훼손함을 보였다.

실험적 데이터를 고려하면 결론적으로 기존 형태의 워터마킹 방법은 H.264의 인트라 프레임에 대해 그 효용성을 보장할 수 없으며, 따라서 새로운 방법의 연구 개발이 절실하다고 판단된다.

ABSTRACT

This paper intend to show the affect of the intra-prediction on the typical digital watermarking method and the fact that the watermarking method has very low effectiveness when it is performed for the intra-frames of H.264. The target watermarking method was the one for imperceptibility and robustness and was assumed to be performed during the intra-compression process by the H.264 technique. Also this method was assumed to insert watermark data and to extract it for certification if needed. The problem is that the resulting data from the re-engineering of the watermark insertion process to extract the watermark data is different from the one before. We experimentally showed that it stems from the intra-prediction itself. That is, we showed that the resulting image data from only compression without watermarking changes if it is re-compressed by the same conditions as the first compression and it is because the intra-prediction modes as well as the coefficient values change. Also, we applied one blind and one semi-blind watermarking methods to show that the typical attacks after watermarking makes this problem much more serious and lowers the effectiveness of the watermarking method dramatically.

Therefore we concluded by considering the experimental data that a typical watermarking method which has been researched so far cannot guarantee the effectiveness of intra-frame watermarking and it is highly required to developed a new kind of methodologies.

키워드

H.264, Digital Watermarking, Intra prediction, Content Security, Copyright Protection

* 광운대학교 전자재료공학과

** 광운대학교 교양학부

접수일자 2008. 08. 20

Ⅰ. 서 론

2003년 H.264가 동영상압축을 위한 국제표준으로 채택된[1] 후 이 기술의 활용분야는 급속도로 확대되고 있다. 특히 지상파 및 위성파 DMB 등과 같은 무선통신분야로부터 이 기술이 적용되고 있으며[2], IPTV 등과 같은 유선통신분야 뿐만 아니라 stereoscope나 MVC(multi-view video coding)와 같은 차세대 영상분야로까지 확대되고 있다[3].

디지털 데이터는 그 특성상 변조와 위조가 쉬워 이전의 기술에서도 디지털 영상 콘텐츠의 소유권 보호를 위한 워터마킹 기술이 꾸준히 연구되고 발전되어 왔다[4]. 디지털 워터마킹 기술은 그 적용 대상 영상/비디오가 획득된 후 저장되는 경우에는 그 데이터를 압축하는 기술에 근거할 필요가 없으나, 획득된 데이터를 곧바로 전송하여야 하는 등의 경우에는 여러 가지 제약 때문에 압축 기술을 기반으로 수행되어야 한다. 따라서 새로운 압축 또는 데이터처리기술이 개발되면 그 기술을 기반으로 하는 워터마킹 기술이 개발되어야 한다.

H.264 뿐만 아니라 이전의 MPEG 기술로 압축된 데이터는 그 압축방법에 따라 인트라(intra) 프레임과 인터(inter) 프레임으로 나눌 수 있다. 이 중 인터 프레임은 움직임 예측/보상기술에 따라 압축하는 프레임이기 때문에 움직임 벡터를 사용하기에는 화질열화의 정도가 심하고 인터 차영상은 그 값이 매우 작아 이전 기술에서는 대부분의 워터마킹 기술이 인트라 프레임을 대상으로 하였다. 따라서 본 논문에서도 H.264-기반의 인트라 프레임에 대한 워터마킹 기술을 타겟으로 한다.

지금까지 H.264 인트라 프레임에 워터마킹을 수행하는 몇몇 기술들이 발표되었다. Noorkami[5]는 워터마킹을 수행할 위치를 무작위로 선정하였으며, Qiu[6]는 DCT 영역의 강인성 워터마킹과 움직임 벡터의 파괴성(fragile) 워터마킹을 복합한 워터마킹 기술을 발표하였고, Lu[7]은 블록 극성(polarity)과 인덱스 변조(index modulation)을 이용한 워터마킹 기술을 발표하였다. 그러나 이들 방법들의 성능은 그리 좋지 못하여 현실적인 사용이 어렵다. 또한 Noorkami[8]의 다른 연구에서는 인간의 시각모델을 이용하여 확률적으로 워터마크의 존재유무를 판별하는 방법을 제안하였다.

H.264가 국제표준으로 채택된 시기를 감안하면 이 기술을 기반으로 하는 디지털워터마킹 기술이 상당히 연구되고 개발되었어야 할 것이나, 현재까지 발표된 연구결과는 그리 많지 않다. 그 이유는 H.264가 인트라 프레임에 대해서도 공간예측/보상을 수행하며 이 알고리듬이 heuristic이므로 워터마크를 추출하기 위해 필요한 재압축 과정이 그 전의 결과를 그대로 생성해 내지 못하기 때문이다. 이 이유가 이전 연구들에서 좋은 결과를 보이지 못하는 이유이기도 하다. 본 논문에서는 이 알고리듬의 특성을 파악하고 이 알고리듬이 디지털 워터마킹에 미치는 영향을 파악하는 것에 그 목적을 두고 있다. 즉, 본 논문은 H.264-기반 인트라 프레임을 대상으로 하는 디지털 워터마킹 방법을 제안하고자 하는 것이 아니라 인트라 프레임에 워터마킹을 수행하기 어려운 문제점을 분석하는 것을 목적으로 한다. 따라서 먼저 대표적인 영상데이터에 대한 디지털워터마킹 방법을 규명하고, 이 방법을 타겟으로 할 때 인트라 예측이 워터마킹에 미치는 영향을 분석한다. 이 분석을 실증하기 위해서 몇 가지 실제적인 워터마킹 방법을 적용하여 그 결과로 일반적인 워터마킹 방법의 비 실효성을 보이고자 한다.

Ⅱ. 배경

2.1. H.264의 인트라 예측

앞서 언급한 것과 같이 H.264에서는 heuristic 알고리듬 중 하나인 인트라 예측을 수행하는데, 그림 1에 인코딩 과정과 디코딩 과정을 나타내었다[1]. 인코딩은 이미 코딩된 좌측 및 상측의 블록들의 값을 이용하여 총 13가지(16×16 블록단위의 4가지, 4×4 블록단위의 9가지)의 예측모드를 모두 연산하여 그 중 최적의 모드를 선택하여 예측하고, 그 예측된 결과와 원 영상과의 차이영상을 추출하여 DCT 및 양자화를 수행하고 최종적으로 엔트로피 코딩을 거쳐 비트 스트림으로 만들어진다. 양자화된 결과는 또한 역양자화, 역 DCT, 역 예측을 통해 복원하여 다음 블록들의 예측에 사용한다.

인트라 예측모드의 결정에 사용되는 함수는,

$$Cost = Distortion + \lambda_{\text{mode}} \bullet Rate \quad (1)$$
$$(\lambda_{\text{mode}} = 0.85 \times 2^{(QP-12)/3})$$

이며, 여기서 *Distortion*은 SAD(Sum of Absolute Differences) 또는 SSD(Sum of Sqaures of Differences)와

같은 에러함수를 나타내고, *Rate*는 해당 매크로블록(macro-block)의 비트율, 그리고 QP는 양자화 인덱스를 각각 나타낸다. 즉, 13개의 예측모드 중 Cost값이 가장 작은 모드를 예측모드로 결정한다.

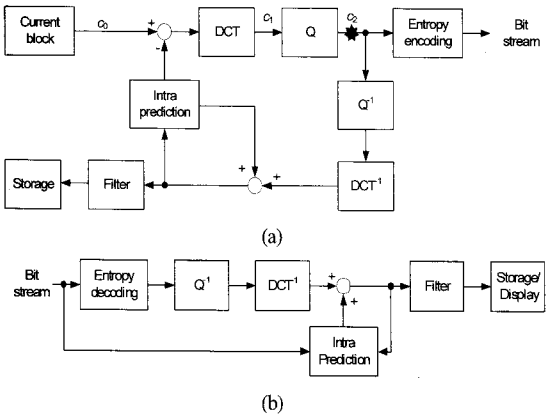


그림 1. H.264의 인트라 예측; (a) 인코딩, (b) 디코딩
Fig. 1. Intra prediction of H.264;
(a) encoding, (b) decoding

2.2. 영상/비디오의 디지털 워터마킹

워터마킹 기술은 매우 오래전부터 발전되어 온 기술인데, 여기서는 최근에 발표되고 사용되는 디지털 데이터, 특히 영상/비디오 데이터에 대한 워터마킹 기술들을 본 논문과 연관되는 것들을 중심으로 개략적으로 분류하여 설명하고자 한다[4]. 워터마킹은 대상 데이터 중 워터마킹을 수행할 위치들을 선정하여 수행하는 것이 일반적이다.

2.2.1 신호처리 과정과의 연계성

워터마킹을 수행할 디지털 데이터의 성격에 따라, 신호처리(예를 들어, 데이터 압축) 과정 중에 수행하여야 하는 경우가 있고, 신호처리와 전혀 무관하게 수행할 수 있는 경우가 있다. 전자의 경우는 TV의 중계방송과 같이 데이터를 획득한 직후 그 데이터를 전송하여야 하는 경우에 해당하고, 후자는 VoD 서비스와 같이 데이터를 획득하여 저장하고 필요할 때 그 데이터를 전송 또는 사용하는 경우이다. 전자의 경우가 더 많은 제약조건을 가지고 있다고 볼 수 있으며, 후자는 워터마킹을 위해서 어떠한 신호처리 방법을 사용하여도 무방하다.

2.2.2 Blind형과 non-blind형

소유권 주장 등을 목적으로 워터마크 데이터를 추출할 때 워터마킹 대상 데이터 정보의 사용여부에 따라 blind 워터마킹(사용하지 않는 경우)과 non-blind 워터마킹(사용하는 경우)으로 나눈다. 대체로 알고리듬 자체는 blind 워터마킹이 복잡하며, 최근에는 blind 워터마킹을 선호하는 추세이다.

2.2.3 강인성(robust) 워터마킹과 파괴성(fragile) 워터마킹

워터마킹이 수행된 콘텐츠에 공격을 가할 경우 그 공격에도 삽입 또는 설정된 워터마크 데이터가 남아있도록 하는 방법(robust)과 공격에 의해 워터마크 데이터 또는 대상 콘텐츠의 데이터가 망가지도록 하는 방법(fragile)이 있다. 전자는 공격이 가해졌음에도 불구하고 그 콘텐츠의 소유권을 주장하기 위해 사용되고, 후자는 주로 공격의 유무를 판별하기 위해 사용된다.

2.2.4 데이터 삽입형과 원 데이터 추출형

소유권자의 특정 데이터를 대상 데이터의 특정위치에 삽입하는 방법과 대상 데이터의 특정위치의 데이터들을 추출하여 워터마크 데이터로 삼는 방법이 있다. 일반적으로는 데이터 삽입형을 주로 사용하고 있다.

2.2.5 절대치 삽입형과 상대치 삽입형

데이터 삽입형에서 워터마크 데이터를 삽입할 때 그 값을 그대로 삽입하는 방법(식 2)와 삽입위치의 데이터에 대한 상대치를 삽입하는 방법(식 3)이 있다.

$$I' = I + \alpha W \tag{2}$$

$$I' = I(1 + \alpha W) \tag{3}$$

여기서 I는 워터마크 데이터를 삽입할 위치의 콘텐츠 데이터, W는 해당위치에 삽입할 워터마크 데이터, I'은 워터마크가 삽입된 콘텐츠 데이터를 각각 나타내고, $\alpha(0<\alpha\leq 1)$는 워터마크 데이터에 대한 가중치이다. 워터마크 데이터가 2진값으로 이루어진 경우는 각 비트를 0과 1 또는 -1과 1의 두 값을 주로 사용한다. 0과 1의 2진 값을 사용할 경우에는 콘텐츠 데이터의 특정 비트를 워터마크 비트로 치환하는 방법을 사용하기도 한다.

본 논문에서 타겟으로 삼는 워터마킹 방법은, H.264로 영상/비디오 데이터를 압축하는 과정 중에 blind 또는 non-blind 워터마킹을 수행하며, 그 방법은 유사시 소유권을 주장할 수 있는 강인성 워터마킹 방법이다. 또한 워터마크 데이터를 대상 콘텐츠에 삽입하는 방법을 가정하며, 워터마크 데이터를 2진 데이터로 가정하여 콘텐츠의 특정 위치에 있는 데이터의 특정 비트를 워터마크 데이터로 치환하는 방법을 가정한다. 이상의 워터마킹 방법은 현재 연구하고 사용하는 대표적인 방법에 해당한다고 볼 수 있다.

이 가정에서 워터마킹을 수행하는 위치는 H.264의 압축과정과 밀접한 관계가 있다. 그림 1 (a)에서 세 위치를 나타내었다. 그 중 C_0는 그 대상 데이터가 공간영역의 값인데, 공간영역보다 주파수영역에서의 워터마킹 방법이 더 효과적이라고 대부분의 연구결과에서 발표하고 있다. 한편 C_1과 C_2는 DCT에 의한 주파수영역 데이터의 위치이며, 그 중 C_1은 양자화 이전의 데이터를 대상으로 하는데, 워터마크 데이터를 이 위치에서 삽입할 경우 양자화에 의해서 그 값이 소멸되거나 훼손될 가능성이 매우 높다. 물론 양자화에 의해서 소멸될 것을 감안하여 식 (2)와 (3)의 a를 조정할 수 있으나, 이 경우는 양자화 후의 C_2위치에서 수행하는 것과 거의 차이가 없다. 따라서 본 논문에서는 C_2 위치에서 워터마킹을 수행하는 것으로 한다.

Ⅲ. 워터마크 데이터 추출을 위한 H.264의 재압축 과정

디지털 워터마킹 기술의 목적은 필요시 원하는 워터마크 데이터를 추출하여 원하는 목적에 사용하는 것이다. 일반적으로 전송받은 영상 또는 비디오는 수요자에 의해 재생될 것이다. 그 후 이 데이터를 저장할 때 반드시 H.264의 포맷으로 저장된다고 볼 수는 없을뿐더러, 특히 불법적인 복제나 변/위조가 자행된 경우는 특히 그 데이터 포맷을 변경한다고 볼 수 있다. 따라서 수요자의 손에 있는 영상데이터로부터 원 소유자의 워터마크 데이터를 추출하기 위해서는 워터마크 데이터를 삽입할 때와 거의 같은 과정(re-engineering)을 거쳐 추출되는 것이 더욱 일반적이라고 할 수 있다. 따라서 본 장에서는 H.264 인트라 프레임을 대상으로 재압축 했을 때 나타나는 특성을 조사하고 이것이 워터마킹에 미치는 영향을 논의하고자 한다.

3.1. 재압축 과정에 의한 인트라 예측모드의 변화

본 논문에서는 다음의 방법으로 H.264 인트라 프레임의 재압축 과정을 수행하였다.

<1> 비디오 스트림을 주어진 QP값으로 모든 프레임을 인트라 프레임으로 간주하고 압축코딩을 수행한다.
<2> 압축된 비디오를 재생한다.
<3> 필요하다면 재생된 비디오에 공격을 가한다.
<4> <2> 또는 <3>의 결과를 <1>과 동일한 조건과 방법으로 압축코딩 한다.
<5> <1>의 결과와 <4>의 결과를 비교한다.

이 방법에 따라 Foreman과 Mobile의 두 테스트 비디오를 각각 50 프레임씩 적용하였으며 각 비디오는 QP를 28과 30으로 각각 수행하였다. 이 때 객관성을 담보하기 위해서 H.264의 참조 소프트웨어인 JM 9.8을 사용하였다. 실험결과 중 먼저 예측모드가 변화한 결과를 표 1에 나타내었다. 표에서의 단위는 4×4 블록을 한 단위로 계산한 것이며, 표에 나타낸 각 case는 표 아래에 주석과 같다. 또한 그림 2에 한 프레임의 예를 보였는데, 그림에서 흰 색으로 표시된 부분이 예측모드가 변화한 4×4 블록들이다.

표 1. 재압축 과정에 의해 인트라 예측모드가 변화한 평균 4×4 블록 수
Table 1. The average number of 4×4 blocks changing their intra-prediction modes by re-engineering

Video	QP	1	2	3	4	Total
Foreman	28	0	4.8	0	29.2	34.0
	30	0	1.6	0	34.3	35.9
Mobile	28	0	0	0	61.0	61.0
	30	0	0	0	69.9	69.9

1: 특정 16×16 모드가 다른 16×16 모드로 변화한 경우
2: 16×16 모드가 16개의 4×4 모드로 변화한 경우
3: 16개의 4×4 모드가 16×16 모드로 변화한 경우
4: 특정 4×4 모드가 다른 4×4 모드로 변화한 경우
Total: Case 1 + Case 2 + Case 3 + Case 4

표에서 보듯이 Foreman의 경우는 평균 전체 블록(1,584 블록)의 약 2.2%, Mobile의 경우 약 4.1%의 블록들의 예측모드가 변화하였다. QP가 증가할수록, 즉 양자화 강도가 강할수록 예측모드의 변화률이 높으며, 고주파 성분이 적은 영상보다 고주파 성분이 많은 영상(Mobile)에서 예측모드가 변할 가능성이 높다는 것을 알 수 있다. 각 case별로 보면, 특정 16×16모드에서 다른 16×16모드로 변화하는 확률과 4×4모드에서 16×16모드로 변화하는 확률은 매우 낮으며, 4×4모드에서 다른 4×4 모드로 변화할 가능성이 가장 높다.

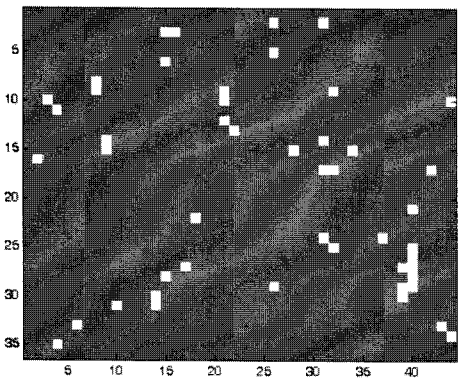


그림 2. 인트라 예측모드가 변화한 한 프레임의 예
Fig. 2. An example frame changing the intra-prediction modes

인트라 예측은 식 (1)의 cost값이 가장 작은 예측모드를 선택하여 그 모드에 해당하는 예측값으로 각 화소의 값을 예측하고 그 예측값과 화소의 원래값의 차이(차영상, residual image)를 구하여 그 차이값을 DCT 한 후 양자화한다. 결국 예측모드와 차영상의 양자화 결과만을 무손실 압축코딩하게 된다. 따라서 예측모드가 변화하면 해당 영상의 각 화소값이 변화하지 않더라도 차영상의 값은 물론 그 DCT 및 양자화한 값은 달라진다. 따라서 앞장에서 언급한 워터마킹을 수행할 경우 공격을 가하지 않은 재압축 만으로도 예측모드가 달라지며, 그 결과 워터마크를 삽입한 화소의 값이 변화하여 추출한 워터마크값이 삽입한 것과 다른 결과를 가져온다.

3.2. 재압축 과정에 의한 계수값의 변화

그러면 실제로 재압축 과정에 의해 얼마나 많은 계수값이 변화하는가?

이에 대해 앞 절에서 수행한 실험결과로부터 얻은 데이터를 표 2에 열거하였다. 표 2에 의하면 Foreman 영상의 경우 평균적으로 4×4 블록당 약 0.18개(1.2%), Mobile 영상은 약 0.63개(3.9%)의 계수가 재압축 과정만의 이유로 변화하는 것을 알 수 있다. 짐작한 대로 그 변화의 대부분은 예측모드가 변화한 블록들에서 발생하였는데, 특이한 것은 예측모드가 변화하지 않은 블록에서도 계수가 변화한다는 것이다. 예측모드가 변화하지 않은 블록에서 Foreman 영상의 경우 평균 약 0.16개(1%), Mobile 영상의 경우 약 0.44개(2.8%)개의 계수가 변화한다. 앞에서도 언급한 것과 같이 이러한 계수변화는 워터마킹을 재압축 과정 자체가 그 실효성을 잃게 만드는 결과를 초래하며, 워터마크 삽입 후 공격이 가해지는 경우에는 더욱 그 실효성을 떨어뜨린다. 공격이 가해진 경우에 대한 결과는 다음 장에서 실제의 예를 들어 다시 설명하기로 한다.

이와 같이 재압축 과정에 의해 예측모드가 변화하고, 그에 따라 계수값이 변화하는, 심지어 예측모드가 변화하지 않은 블록에서조차 계수값이 변화하는 것은 인트라 예측이 가지는 heuristic한 성격 때문인 것으로 사료

표 2. 재압축 과정에 의해 변화하는 4×4 블록당 평균계수의 개수
Table 2. Average number of coefficients per 4×4 clock changing their values by re-engineering

Video	QP	Average number of coeff. in an unchanged block	Average number of coeff. in a changed block	Total average number of changed coeff. in a block
Foreman	28	0.1427	1.3888	0.1687
	30	0.1734	1.5266	0.1968
Mobile	28	0.4072	5.1711	0.5915
	30	0.4743	4.5102	0.6506

표 3. Blind 워터마킹 예에 대한 실험 결과
Table 3. Experimental results for the blind watermarking example

Attack		Foreman				Mobile			
		QP=28		QP=30		QP=28		QP=30	
		Error Rate(%)	PSNR (dB)	Error Rate(%)	PSNR (dB)	Error Rate(%)	PSNR (dB)	Error Rate(%)	PSNR (dB)
No watermarking		-	37.08	-	35.55	-	34.54	-	32.56
No attack		1.34	36.69	1.52	35.09	1.57	34.28	1.84	32.31
Gauss. Noise Addition	2%	16.16	32.25	9.34	31.61	19.67	31.24	12.22	30.15
	4%	39.04	27.66	28.96	27.40	40.43	27.27	32.45	26.81
Blurring	0.5	34.80	32.13	34.24	31.73	37.95	24.26	36.92	24.13
	1.0	36.74	28.55	35.73	28.43	39.57	21.42	37.27	21.39
Sharpening		19.32	29.89	23.56	29.36	34.77	22.71	32.70	22.55
JPEG Compress	Q=8	6.39	35.86	4.07	34.64	7.99	31.44	6.72	30.45
	Q=3	29.42	33.24	23.41	32.64	31.99	27.10	25.76	26.78

된다. 동일한 조건으로 재압축 과정을 한다고는 하지만 실제의 입력값에는 차이가 있다. 첫 번째 압축과정에서는 원 영상이 대상 데이터로 입력되는 반면, 재압축 과정의 경우에는 이미 한 번 압축되었던 데이터가 입력된다. 물론 압축과정에서 DCT와 양자화과정을 거치면서 데이터가 정제되어 출력되기 때문에 그 차이가 상당히 줄어든다고 할 수는 있지만 그 차이가 완전히 없어지지는 않는 다는 것이다. 실제의 실험에서도 그림 1 (a)의 DCT전, DCT 후, 양자화 후의 각 데이터값들이 첫 번째 압축과정과 두 번째 압축과정에서 조금씩 차이가 나는 것을 볼 수 있다. 이 차이가 물론 양자화로 무시되는 경우가 대부분이지만, 앞의 표에서 본 것과 같이 양자화 과정에서도 그 차이가 살아남는 경우가 발생하기 때문이다.

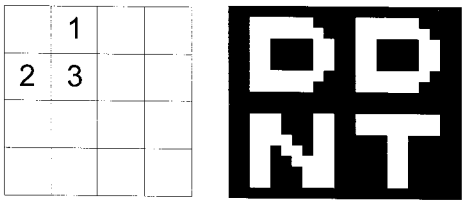
Ⅳ. 디지털 워터마킹의 예

앞 장에서 설명한 재압축 과정 자체에 의한 예측모드 및 데이터의 변화는 워터마킹 기술을 적용하는데 큰 장애가 된다. 더욱이 워터마킹의 삽입과정에서 데이터를 변화시킨다면 재압축 과정과 워터마크 삽입에 의한 데이터의 변화가 누적되어 더 많은 데이터의 변화를 가져오게 될 것이며, 이것은 워터마크 데이터를 추출하기 위한 재압축 과정에서 더 많은 예측모드 및 데이터의 변화를 초래할 것이다. 뿐만 아니라 워터마킹된 데이터에 대해 악의적/비악의적 공격이 가해지는 경우는 이 변화가 더욱 커져 추출된 워터마크 데이터에 대한 신뢰도가 매우 저하될 것이다.

본 장에서는 실제의 워터마킹과 그 위에 부과된 공격이 삽입된 워터마크 추출에 얼마나 큰 영향을 미치는지를 보이기 위해서 두 종류의 워터마킹 기술을 실제로 적용한 예를 보이고자 한다. 이 두 방법 모두 2장 말미에서 설명한 이 논문에서 타겟으로 하는 방법의 범주에 속한다. 워터마킹을 적용하는 위치는 두 방법 모두 양자화 이후(그림 1 (a)의 C_2)이며, C_2의 위치는 비록 양자화가 이루어진 후이기는 하지만 DCT 영역이므로 주파수 대역별로 계수가 분포되어 있다. H.264 이전에는 대부분 8×8 블록의 DCT를 수행하였으나 H.264의 경우는 4×4 블록으로 DCT를 수행하므로 계수들도 16개 단위로 주파수 대역을 형성한다. 본 연구와 같이 삽입한 워터마크의 비가시성(imperceptibility)과 강인성을 필요로 하는 경우 많은 논문에서 DC계수를 제외한 가장 저주파 계수에 워터마크를 삽입하는 것이 가장 좋은 성능을 보였었다. 따라서 본 논문에서도 그림 3 (a)에 나타낸 세 위치를 워터마킹 위치로 간주한다. 특히 이 위치의 세 계수 중 대각선방향의 계수(위치 3)를 선택하였는데, 실험에 의하면 이 세 계수 중 미미한 차이지만 대각선방향의 계수가 가

장 높은 성능을 보였기 때문이다.

디지털 워터마킹의 강인성과 비가시성은 위에서 언급한 워터마킹 위치 뿐 만 아니라 워터마크 데이터의 양에도 큰 영향을 미친다. 본 논문에서는 기본적으로 8×8 블록마다 한 개의 워터마크 데이터를 삽입하는 것으로 한다. 이 경우 8×8 블록에는 4개의 4×4 블록이 존재하는데, 본 논문에서는 좌상위 4×4 블록에 워터마크 데이터를 삽입하는 것으로 하였다. 본 논문에서는 또한 QCIF(176×144) 크기의 비디오 영상을 대상으로 하였으며, 워터마크 데이터는 그림 3 (b)에 나타낸 22×18(396개) 크기의 2진 영상으로 하였다. 워터마크를 삽입하는 방법은 양자화 후 해당 위치의 계수의 LSB(least significant bit)를 해당 워터마크 비트와 치환하는 방법을 사용하였다.



(a) (b)

그림 3. (a) 4×4 블록 내의 워터마킹 위치, (b) 2진 워터마크 영상
Fig. 3. (a) Watermarking position in a 4×4 block, (b) binary watermark image

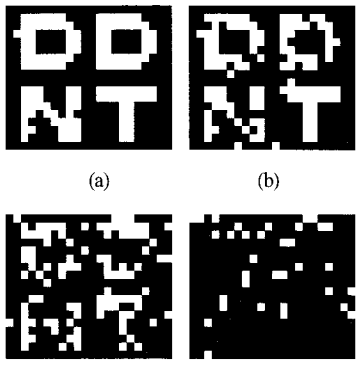
4.1. 고정위치의 Blind 워터마킹

워터마킹의 첫 번째 예는 워터마킹 위치를 고정시키고 고정된 각 위치마다 한 개의 워터마크 비트를 삽입하는 것이다. 여기서는 위에서 언급한 기본적인 위치를 그대로 수용하여 8×8 블록마다 한 개의 워터마크 데이터를 삽입하며, 그 위치는 좌상위 4×4 블록의 위치 3(그림 3 (a))이다. 즉, 인트라 예측모드에 무관하게 균일하게 워터마크 데이터를 전 영상에 분포시키며, 한 개의 영상에 정확히 하나의 워터마크 영상이 삽입된다.

본 논문에서는 워터마크 데이터의 삽입뿐만 아니라 악의적/비악의적 공격에 대한 결과도 관심 내에 있으므로 대표적인 공격들을 고려한다. 먼저 대표적인 비악의적 공격은 데이터를 다른 기술로 압축하는 것을 선택하였으며, 그 방법은 JPEG 압축이었고, 두 가지 압축률(Q=8, 3)로 압축하였다. 악의적인 공격은 전형적인 방법들, 즉 Gaussian 잡음첨가(2%와 4%), blurring 공격(0.5, 1.0), sharpening 공격들을 사용하였다.

앞의 실험에서와 같이 Foreman과 Mobile을 대상 영상으로 하고 각 영상의 QP는 28과 30을 각각 적용하여 각 경우 50 프레임의 영상을 인트라 압축하면서 워터마킹을 삽입하고 공격한 후 워터마크를 추출한 결과를 표 3에 나타내었다. 표 3에서는 각 경우에 대해 결과영상의 화질(PSNR)과 추출한 워터마크의 에러율을 각각 나타내었는데, 화질은 원영상 대비의 값이며, 에러율은 원 워터마크 데이터에 대해 발생한 에러비트의 비율이다. 표에서 'No watermarking'은 워터마킹을 수행하지 않은 압축만을 수행한 영상을 나타내고, 'No attack'은 워터마킹을 수행하였으나 공격을 가하지 않은 영상에 대한 결과이며, 그 외의 열은 워터마킹을 수행하고 해당 공격을 가한 영상에 대한 결과들이다.

압축만 수행한 결과에서 보듯이 동일한 조건에서 고주파 성분이 적은 Foreman 영상의 화질이 더 좋은 것을 알 수 있다. 앞에서 언급한 것과 같이 재압축 과정 자체에 의한 워터마크 데이터의 손실은 'No attack'열에 나타나 있는데, 약 1~2%의 에러율을 보이는 것을 알 수 있다. 공격을 가한 경우의 에러율은 급격히 증가하는 것을 볼 수 있는데, 특히 Gaussian 잡음첨가와 몽롱화(blurring) 공격에서 가장 두드러지는 것을 볼 수 있다. 에러율 대비 워터마크 데이터의 인식정도를 나타내기 위해 네 가지의 추출된 워터마크를 그림 4에 나타내었다.



(a) (b)

(c) (d)

그림 4. 에러율에 따른 추출된 워터마크; (a) 1.1%, (b) 6.8%, (c) 19.7%, (d) 34.9%
Fig. 4. Extracted watermark at error rate; (a) 1.1%, (b) 6.8%, (c) 19.7%, (d) 34.9%

표 4. Semi-blind 워터마킹 예에 대한 실험 결과
Table 4. Experimental results for the semi-blind watermarking example

Attack		Foreman				Mobile			
		QP=28		QP=30		QP=28		QP=30	
		Error Rate(%)	PSNR (dB)	Error Rate(%)	PSNR (dB)	Error Rate(%)	PSNR (dB)	Error Rate(%)	PSNR (dB)
No watermarking		-	37.08	-	35.55	-	34.54	-	32.56
No attack		0.0	36.82	3.15	35.25	0.0	34.37	0.07	32.36
Gauss. Noise Addition	2%	18.20	32.30	9.32	31.73	14.53	31.28	8.37	30.16
	4%	38.68	27.64	37.20	27.46	39.50	27.25	32.17	26.88
Blurring	0.5	32.97	32.15	34.15	31.77	38.85	24.27	38.59	24.14
	1.0	37.24	28.56	37.23	28.45	41.25	21.43	40.72	21.39
Sharpening		24.42	29.99	24.41	28.49	35.77	22.73	32.64	22.55
JPEG Compress	Q=8	2.93	35.96	3.28	34.78	4.98	31.49	2.36	30.49
	Q=3	29.61	33.27	26.10	32.72	31.04	27.11	22.55	26.78

그림에서 보듯이 에러율이 10% 이상인 추출된 워터마크는 원 워터마크를 인식할 수 없다고 판단하여야 할 것이다. 따라서 이 워터마킹 방법은 그 실효성이 매우 낮다.

4.2. 비용함수에 의한 Semi-blind 워터마킹

두 번째 워터마킹 예는 식 (1)의 비용함수 값을 이용한다. 인트라 예측을 수행할 때 13개의예측모드 각각에 대해 식 (1)의 비용값을 계산하여 그 중 가장 적은 비용값을 갖는 예측모드를 선택한다. 재압축 과정에서 예측모드가 변화한다는 것은 재압축 과정에서 입력되는 데이터가 처음 압축과정에서와는 다른 데이터가 입력되기 때문이며, 그 차이는 DCT 및 양자화로 상당부분 그 차이가 사라진다는 설명은 앞에서 한 바 있다. 그 차이가 크지 않다면 예측모드를 결정할 때 최선의 예측모드와 차선의 예측모드의 비용값 차이가 클수록 재압축 과정에서 예측모드가 변화할 확률은 감소한다. 이점을 착안하여 이번 예제는 인트라 예측시 최선과 차선의 비용값 차이가 큰 것을 선택하도록 하였는데, 그 비용값 차이의 문턱값을 100으로 잡았다. 즉, 최선과 차선의 예측모드의 비용값 차이가 100이상인 블록만 워터마킹 위치로 선정하였다. 이 경우는 선택된 블록이 4×4 블록일수도 있고 16×16블록일 수도 있다. 따라서 앞의 예제에서와 같이 8×8 블록당 한 비트의 워터마크를 삽입할 수가 없으므로 여기서는 비용값 차이에 의해 선택된 모든 블록에 대해 4×4 블록 당 한 비트의 워터마크를 삽입하였다. 각 블록의 삽입위치는 그림 3 (a)의 위치 3이었다. 한 프레임에는 한 개의 워터마크 영상만을 삽입하여 앞의 예제와 동일한 양의 워터마크를 삽입하였다.

이 방법에 의해 워터마크를 추출할 때 어떤 이유에서 계수값이 변화하고 그에 따라 비용값 차이가 변화하면 비용값 차이가 100이상이었던 블록이 100 이하로 떨어질 수 있을 뿐 아니라 100이하이던 블록이 100이상이 될 수 있다. 실제의 실험에서 이런 경우가 상당히 발생하였는데, 워터마크를 추출할 때 삽입할 때와 동일한 계산에 의해 워터마크 삽입 위치를 찾는다면 삽입된 위치를 놓칠 수도 있고 삽입되지 않은 위치를 선정할 수도 있다. 삽입위치를 잘못 선정한다면 올바른 워터마크를 추출할 확률은 매우 낮아지므로, 본 논문에서는 삽입할 때 선정된 워터마크 위치를 추출할 때 사용하는 것(semi-blind)으로 하여 워터마크 삽입위치를 삽입할 때와 동일한 위치를 찾는 것으로 하였다.

이 방법으로 워터마킹을 수행하고 앞에서 정의한 공격을 가하여 실험한 결과는 표 4에 나타내었다. 표 3의 결과에 비해 공격을 가하지 않았을 때 추출된 워터마크의 에러율은 매우 낮았으나, 공격을 가했을때의 에러율은 큰 변화가 없거나 오히려 더 높아진 것을 확인할 수 있다. 이것은 재압축 과정과 공격에 의해서 단

순히 계수값만을 변화시키는 것이 아니라 예측모드 자체를 변화시키며 그 결과 차영상의 값이 첫 번째 압축시의 값과는 많은 차이를 보이는 결과라는 것을 알 수 있다.

V. 논의 및 결론

3장에서 보인 재압축 과정에 의한 예측모드 변화와 4장에서 보인 두 가지 예제에서 보듯이 heuristic한 방법인 인트라 예측에 의한 압축이 일반적인 워터마킹 방법의 효용성을 크게 감소시킨다. 이것은 앞에서도 언급한 것과 같이 워터마크 추출을 위한 재압축 과정에서의 인트라 예측이 원래의 압축과정에서의 예측과 다른 예측모드를 최선의 예측모드로 판정하는 것에 크게 기인하고 있다. 앞의 두 예제는 특별한 조건을 부여한 특별한 경우로 볼 수도 있으나 현실적으로 지금까지의 워터마킹 방법들이 이 범주를 크게 벗어나지 않기 때문에 이 방법들이 일반화된 방법에서 크게 벗어났다고는 볼 수 없다. 따라서 이와 유사한 방법, 즉, 삽입된 워터마크를 추출할 때 인트라 예측을 다시 수행해야 하는 방법은 H.264의 고유한 문제 때문에 그 효용성을 보장할 수 없다고 볼 수 있다.

이와 같은 문제는 다분히 H.264의 인트라 예측에서 비롯된다고 볼 수 있다. 그렇다면 인트라 예측과정을 피하는 워터마킹 방법을 생각할 수 있는데, 이 경우는 그림 1 (a)의 C_0나 다음 블록들의 예측을 위해 양자화 후 복원된 블록을 고려할 수 잇는데, 이 모두 공간영역의 데이터들이다. 공간영역 데이터는 이미 많은 연구에서 그 실효성이 떨어진다고 발표된 바 있는데, 그 이유는 공간영역 데이터를 변화시키면 그 결과가 DCT 및 양자화과정에서 어떻게 변화될지를 예측하기 힘들기 때문이다.

또 다른 가능성은 인트라 프레임이 아닌 인터 프레임에의 워터마킹이다. 이미 이런 방법에 대한 발표가 있기는 하였으나[9], 이 역시 인트라 프레임에 대한 방법에 비해 크게 개선된 성능을 보이지 못하고 있다. 우리 연구진은 이미 인터예측에 대해서도 유사한 실험을 수행하였는데, 인트라 예측과 유사한 문제가 발생됨을 확인하였다. 따라서 인터프레임에서도 일반적인 방법이 그 효용성을 보장받을 수 없다고 판단된다.

결과적으로 H.264에 대해서는 지금까지 일반적으로 생각하였던 워터마킹 방법 이 외의 방법, 즉 인트라 예측모드의 변화를 막을 수 있는 방법이 연구되어야 할 것이다. 여기에는 비용값을 계산하는 방법의 변화를 포함하여 인트라 예측모드가 재압축 과정에도 변화하지 않도록 하는 방법을 비롯해서 일반적이 방법이 아닌 H.264만의 특별한 방법 등이 고려되어야 할 것이며, 이 논문을 토대로 이와 같은 방법이 연구되어 신뢰할 수 있는 H.264의 워터마킹 방법이 개발되기를 바라는 바이다.

참고문헌

[1] ITU-T Recommendation H.264(2003), "Advanced Video Coding for generic audiovisual services", |ISO/IEC 14496-10:2003, "Information technology, Coding of audio-visual objects-Part10:Advanced video coding".

[2] 구제길 외, DMB 시스템 기술, 진한M&B, 2006.

[3] I. J. Cox, M. L. Miller, and J. A. Bloom, Digital Watermarking, Morgan Kaufman Publisher, San Francisco, CA, 2002.

[4] T. J. Naughton, Y. Frauel, B. Javidi and E. Tajahuerce, "Compression of digital holograms for three-dimensional object recognition," SPIE Proc. vol 4471, pp. 280-289, 2001.

[5] M. Noorkami and R. M. Mersereau, "Compressed-domain Video Watermarking for H.264", IEEE International Conference on Image Processing(ICIP 2005), vol.2, pp. 2353-2356, 2005.

[6] G. Qiu, P. Marziliano, A. T. S. Ho, D. He, and Q. Sun, "A Hybrid Watermarking scheme for H.264/AVC Video", Proceedings of the 17th International Conference on Pattern Recognition(ICPR 2004), vol. 4, pp. 2353-2356, 2006.

[7] T. T. Lu, W. L. Hsu, and P. C. Chang, "Blind Video Watermarking for H.264", Canadian Conference on Electrical and Computer Engineering(CCECE 2006), pp. 2353-2356, 2006.

[8] M. Noorkami and R. M. Merserau, "A Framework for Robust Watermarking of H.264-Encoded Video with

Controllable Detection Performance", IEEE Trans. on Information Forensics and Security, vol. 2, no. 1, pp. 14-23, March 2007.

[9] S. Sakazawa, Y. Takishima, and Y. Nakajima, "H.264 native video watermarking method", ISCAS 2006 Proceedings, pp. 1439-1442, 2006.

저자 소개

최 현 준(Hyun-Jun Choi)
한국해양정보통신학회논문지
제 12권 제 12호 참고

서영호(Young-Ho Seo)
한국해양정보통신학회논문지
제 12권 제 12호 참고

김동욱(Dong-Wook Kim)
한국해양정보통신학회논문지
제 12권 제 12호 참고