

# 움직임 벡터의 정보 숨김을 이용한 H.264/AVC의 향상된 오류 검출 방법

## Improved Error Detection Scheme Using Data Hiding in Motion Vector for H.264/AVC

고만근, 서재원  
충북대학교

Man-Geun Ko(aokin8132@cbnu.ac.kr), Jae-Won Suh(sjwon@cbnu.ac.kr)

### 요약

제한된 통신 채널을 통한 실시간 전송을 위해 비디오 데이터의 압축이 필요하다. 압축된 영상 비트열은 전송 중에 발생하는 오류에 아주 민감하다. 전송 과정에서 오류가 발생한 패킷을 받게 되면, 복호기에서는 현재 입력되는 프레임을 잘못 재구성하게 되고 영상의 시공간적 특성에 의해 연속되는 프레임으로 전파된다. 이런 복호 영상의 화질 저하를 줄일 수 있는 방법으로 오류 검출과 오류 은닉이 있다. 이 논문에서는 오류 검출률을 높이기 위해 정보 숨김을 이용한 새로운 오류 검출 방법을 제안한다. H.264/AVC의 화면간 예측에서 발생하는 각 매크로블록의 움직임 벡터의 차분값에 특정한 정보를 숨겨 부호화하였다. 복호기에서는 전송받은 특정 정보의 체크를 통해 오류의 위치를 쉽게 찾아낼 수 있도록 하였다. H.324M 모바일 실험 도구를 이용한 실험 환경을 통해 제안한 알고리즘이 PSNR과 주관적 화질 측면에서 좋은 성능을 보임을 확인하였다.

■ 중심어 : | H.264/AVC | 오류 검출 | 데이터 숨김 | 움직임 벡터 |

### Abstract

The compression of video data is intended for real-time transmission of band-limited channels. Compressed video bit-streams are very sensitive to transmission error. If we lose packets or receive them with errors during transmission, not only the current frame will be corrupted, but also the error will propagate to succeeding frames due to the spatio-temporal predictive coding structure of sequences. Error detection and concealment is a good approach to reduce the bad influence on the reconstructed visual quality. To increase concealment efficiency, we need to get some more accurate error detection algorithm. In this paper, We hide specific data into the motion vector difference of each macro-block, which is obtained from the procedure of inter prediction mode in H.264/AVC. Then, the location of errors can be detected easily by checking transmitted specific data in decoder. We verified that the proposed algorithm generates good performances in PSNR and subjective visual quality through the computer simulation by H.324M mobile simulation tool.

■ keyword : | H.264/AVC | Error Detection | Data Hiding | Motion Vector |

\* 이 논문은 2011년도 충북대학교 학술연구지원사업의 연구비 지원에 의하여 연구되었음.

접수번호 : #130403-003

접수일자 : 2013년 04월 03일

심사완료일 : 2013년 05월 14일

교신저자 : 서재원, e-mail : sjwon@cbnu.ac.kr

## I. 서론

최근 스마트폰과 태블릿 PC와 같은 이동형 기기의 발달과 인터넷이나 무선 네트워크의 통신 환경의 발전과 함께 이를 통한 화상 회의 또는 UCC와 같은 다양한 멀티미디어의 사용이 급증하고 있다. 이런 멀티미디어에서는 동영상 정보가 중심이 된다. 이렇게 동영상 데이터의 통신이 활발히 이루어지는데, 그 정보량은 일반 문자나 음성, 화상 데이터에 비해 상대적으로 매우 크다. 이 영상 정보를 대역이 제한된 채널을 통해 효율적으로 전송하기 위해 여러 압축 표준들이 만들어져 왔다. 비디오 압축 표준 기관 중 ITU-T에서는 전화망을 통한 실시간 화상 전화/회의를 위한 H.26x 표준안을, ISO/IEC에서는 영상의 저장/스트리밍 전송을 목적으로 하는 MPEG-x를 각각 고안하였다. 압축의 목적이 다른 두 개의 표준 기관들이 서로간의 공통 영역을 인정하고 여러 분야에 적용이 가능한 새로운 표준안을 만들기 위해 JVT(Joint Video Team)을 결성하여 제정한 것이 H.264/AVC이다[1][2].

H.264/AVC는 압축 효율을 높이기 위해, 그동안 제안되었지만 복잡도 문제로 채택되지 못했던 여러 기술들을 사용하여 높은 압축률을 가져올 수 있었다. 하지만 이렇게 압축된 비트열을 전달하는 통신 채널은 오류가 쉽게 발생할 수 있는 환경이기 때문에, 이런 고성능의 압축된 비트열은 단일 또는 군집 오류에 의해 쉽게 오류가 발생하게 된다. 이런 오류가 발생된 비트열의 복호 과정은 오류를 제어하기 위한 기법을 사용하지 않으면 사용자측에 심각한 화질 열화를 초래하게 된다.

사용자가 만족할 만한 화질을 보장하기 위해 다양한 오류 강인 기술들이 제안되었다[3][4]. 그 중 오류가 발생된 비트열에 의한 영향을 효과적으로 줄일 수 있는 가장 기본적인 접근 방법은 오류 검출과 오류 은닉이다. 이 논문에서는 잡음이 발생되기 쉬운 채널 환경에서 H.264/AVC로 코딩된 비디오 비트열이 패킷 단위 전송 시 발생하는 오류 검출에 대한 문제점을 제기한다. 오류 검출을 위해 일반적으로 쓰이고 있는 오류 검출 방법은 syntax/semantic 체크 알고리즘이다[5]. 이 방법은 압축 표준에서 정해진 구문과 그 복호 과정에서

발생할 수 없는 상황들을 체크하는 가장 심플한 알고리즘이다. 하지만 이 방법만으로는 오류에 대한 검출률에 한계가 있다. 또한 제대로 된 오류를 검출하지 못하게 될 때, 오류가 발생된 지점이 아닌 지연된 위치에서 검출되는 문제점이 있다. 이를 해결하기 위해 최근 들어 검출률의 정확성 향상을 목적으로 연성 워터 마킹을 바탕으로 하는 숨김 정보를 이용한 여러 오류 검출 방법들이 연구되어 발표되었다[6][7]. 또한, 복호기 자체에서 재구성된 이미지의 영상적 특성을 이용하여 잘못된 복호된 이미지를 검출하는 방법 또한 제안되었다[8]. [6]에서는 전송 과정에서 발생되어지는 DCT(Discrete Cosine Transform) 계수의 개수 합을 짝수로 조정하여 보내는 방법을 제안하였다. 이 방법은 오류가 발생하여도 부/복호기 사이의 약속이 맞을 경우 오류로 처리하지 않는 단점을 갖고 있다. [7]은 위의 문제를 해결하기 위해 DCT계수 처리에 몇 가지 파라미터를 사용하여 좀 더 복잡도를 높이지만 그만큼 정확성을 높이는 방법을 제안하였다. 이 방법들은 오류 검출률을 높일 수 있었지만, DCT 계수에 대한 직접적인 워터 마크 삽입으로 인한 PSNR 손실을 피할 수는 없었다. [8]은 부호화기의 복잡도를 증가시키지는 않았지만, 복호기측의 부담을 높여 실시간적인 영상 재구성을 하는데 다소의 시간이 걸리게 되었다.

이 논문에서 움직임 벡터 영역에 정보 숨김을 이용한 새로운 오류 검출 방법을 제안한다. 위에 제시한 문제점들을 해결하기 위해 숨김 정보의 삽입으로 인한 비트율의 증가와 화질 저하를 최소화하기 위해 율-왜곡 비용 값을 고려하여 움직임 벡터를 활용하는 방법을 제시한다. 2장에서는 H.264/AVC에서의 율-왜곡 비용 값을 기반으로 한 매크로블록 단위의 모드 결정 방법에 대해 간략히 소개한다. 3장에서는 오류 검출률을 높이기 위해 제안한 알고리즘의 삽입과 검출 방법을 설명한다. 4장에서 이에 대한 실험을 바탕으로 결과 분석을 하고 마지막으로 결론을 도출한다.

## II. H.264/AVC의 매크로블록 모드 결정

H.264/AVC는 이전의 코덱 표준과는 다르게 화면간

예측과 화면내 예측을 위한 다양한 크기의 화소 블록을 생성할 수 있도록 하였다. 이런 여러 예측 블록들을 사용한 모드들을 비교하여 가장 적은 율-왜곡 비용을 갖는 모드를 결정하고 부호화하여 복호기로 전송한다.

### 1. 화면간 모드 예측

기존의 비디오 압축 표준의 화면간 매크로블록 모드와는 달리 H.264/AVC의 매크로블록은 이미 전송된 여러 개의 참조 프레임으로부터 [그림 1]에서 보는 것과 같이 16×16 크기부터 4×4까지 다양한 블록 사이즈를 이용한 움직임 예측과 보상을 수행한다.

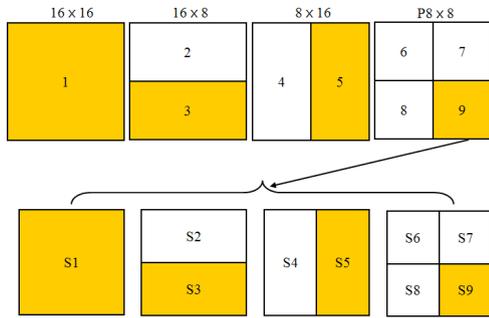


그림 1. 화면간 예측 모드의 종류

화면간 예측 과정에서 각 매크로블록은 한 개에서 최대 16개의 움직임 벡터(MV: Motion Vector) 쌍을 생성하게 된다. 한 쌍의 움직임 벡터는 화면간 예측 과정을 통해 이전 프레임에서 현재의 예측할 매크로블록과 유사한 블록의 위치를 나타내므로

$$mv = mv_x + mv_y \tag{1}$$

과 같이 수평 성분과 수직 성분으로 표현하게 된다. 이 MV값은 그대로 보내는 것은 부호화 과정에서 많은 비트를 차지하게 되므로 H.264/AVC에서는 주변 블록을 통해 찾은 예측된 움직임 벡터(PMV: Predicted MV)와 예측 과정에서 찾은 움직임 벡터간의 차인

$$mvd = pmv - mv \tag{2}$$

를 구하게 되고 이 값을 부호화하여 전송한다.

다양한 블록 타입을 이용한 예측 과정을 통해

$$J_{inter} = SSDs, r(mv) + \lambda_{motion} \cdot R_{motion}(mvd, REF) \tag{3}$$

와 같이 계산되어지는 율-왜곡 비용이 가장 적은 값을 갖는 것이 최적의 화면간 모드로 결정되어진다.

여기서  $s$ 는 현재 예측되고 있는 블록을  $r(MV)$ 은 예측된 움직임 벡터를 이용하여 보상된 블록을 뜻하고,  $SSD(s, r(MV))$ 는 두 개의 블록 사이에 발생한 픽셀들 간 제곱 차의 합(Sum of Squared Difference)이다.  $\lambda_{motion}$ 은 화면간 예측을 위한 라그랑지안 곱수이다.  $R_{motion}(MVD, REF)$ 은 움직임 벡터 차분값(MVD: Motion Vector Difference)과 사용된 참조 프레임의 인덱스와 프레임 넘버를 부호화하는데 쓰인 비트를 나타낸다.

### 2. 화면내 모드 예측

위에서 설명한 화면간 예측 모드와 더불어, H.264/AVC에서는 다양한 화면내 예측 모드 또한 제공한다. 매크로블록의 부호화를 위한 화면내 예측은 공간 영역에서 주변 재구성된 샘플을 참조하여 수행된다. 현재 매크로블록을 예측하기 위해 방향성 보간을 이용한 4×4와 16×16 화소 크기의 예측 블록을 구성한다. 그 후 현재 블록과 예측된 블록 간에 발생한 잔여 데이터를 이용하여 화면내 예측을 통해

$$J_{intra} = SSDs, r + \lambda_{mode} \cdot R_{mode}(s, r, M) \tag{4}$$

와 같이 계산되어지는 율-왜곡 비용이 가장 적은 값을 갖는 것을 최적의 화면내 모드로 결정하게 된다.

여기서  $s$ 는 현재 부호화되는 매크로블록을  $r$ 은 방향성 보간에 의해 예측된 매크로블록을 각각 의미하며, 왜곡 비용은  $SSD$ 에 의해 계산된다.  $\lambda_{mode}$ 는 화면내 예측 모드에서의 라그랑지안 곱수이다.  $R_{mode}(s, r, M)$ 은

현재 블록  $s$ 와 예측 블록  $r$  사이에 발생된 잔여 데이터와 예측된 화면내 모드 정보인  $M$ 을 부호화하는데 쓰이는 비트량을 뜻한다.

### 3. 최적 예측 모드 결정

발생 비트량의 최소화와 만족할 만한 화질의 보장 측면에서의 최적 모드를 결정하기 위하여, H.264/AVC 부호화기는 스킵 모드를 포함한 화면내 예측과 화면간 예측 모드 전부를 테스트 부호화하여 본다. 그리고 그 중 가장 적은 윌-왜곡 비용값을 갖는 모드를 P 프레임 중에서의 최적 모드로 결정하고 이를 최종 부호화하여 전송하게 된다.

## III. 제안하는 오류 검출 알고리즘

제안하는 오류 검출 방법은 채널 전송 중 발생하는 멀티미디어 내용의 손상을 검출하기 위해 전송되는 비트열 안에 정보 숨김을 삽입하였다. 삽입된 숨김 정보는 오류가 발생할 때 쉽게 손상되어 사용자가 알 수 있도록 하는 연성의 성질을 갖게 하였기 때문에, 전송 오류가 발생되면 삽입된 정보와 다르게 된다[9]. 그러므로 복호기측에서는 이를 이용하여 오류를 쉽게 검출할 수 있으며, 매크로블록 단위로 삽입된 숨김 정보에 의해 오류가 발생한 매크로블록 지점 또한 정확히 알 수 있게 된다. 이는 패킷 단위로 오류 처리를 하게 되는 방법보다 전송 중 정확히 전송된 정보를 최대한 유지할 수 있기 때문에 은닉된 영상들의 PSNR을 더욱 향상시킬 수 있게 해준다.

### 1. 제안하는 숨김 정보의 삽입 방법

H.264/AVC는 다양한 크기의 블록 크기를 매크로블록의 화면간 예측 모드에 사용한다. 이것을 제안하는 오류 검출 방법을 사용할 수 있도록 활용하였다.

부호기는 화면간 모드 예측을 하는 동안 숨김 정보를 삽입하게 된다. 이 논문에서는 다양한 크기의 화면간 예측 과정에서 발생하는 MVD의 총합을 강제로 짝수가 되도록 하였다. 이것을 강제 짝수화된 움직임 벡터

차분값 (FEMVD: Forced Even MVD)이라 부르기로 한다. 오류 검출을 위한 숨김 정보 삽입은 모든 화면간 예측 모드에 적용된다. 이를 위해 첫 번째로 각 화면간 예측 모드에서 발생하는 MVD의 총합을 TSMVD(Total Sum of MVD)라 부르기로 하고

$$tsmvd = \sum_{1}^{NPB-1} mvd + lmv \quad (5)$$

와 같이 계산한다.

여기서,  $NPB$ (Number of Prediction block)는 현재 화면간 예측 모드에서 발생하는 분할된 블록들의 개수를 나타낸다.  $LMVD$ (Last MVD)는 맨 마지막 블록에서 발생한 MVD를 가리킨다.  $LMVD$ 는 MVD의 정의에 따라

$$lmvd = lpmv - lmv \quad (6)$$

와 같이 표현이 가능하다.

$LPMV$ (LastPMV)와  $LMV$ (LastMV)는 각각 현재 예측중인 블록 중 가장 마지막에 위치한 블록의 예측 움직임 벡터와 탐색 과정 중 결정된 움직임 벡터이다.

이렇게 TSMVD의 값이 계산된 후, 블록의 마지막 움직임 벡터에 대한 숨김 정보는

$$lmv = \begin{cases} lmv & , \text{if } tsmvd \text{ is even} \\ lmv_w & , \text{if } tsmvd \text{ is odd} \end{cases} \quad (7)$$

를 통해 삽입이 이루어질지에 대한 판단이 결정된다.

$LMV_w$ 는 정보 숨김이 행해진 움직임 벡터를 뜻한다. 즉, 현재 예측되는 화면간 모드에서 발생한 TSMVD의 값이 짝수라면 정보 숨김 과정은 발생하지 않는다. 하지만, 홀수값이 되면 제안된 알고리즘에 의해 마지막 블록의 움직임 벡터값이 변하게 된다.  $LMV$ 는 각각  $LMV_x$ 와  $LMV_y$ 로 이루어져 있는데, 이는  $LMV$ 의 수평 성분과 수직 성분을 뜻한다. 삽입 방법에 의한 화질 열화를 최소화하기 위해 예측 과정을 통해 얻은  $LMV$ 는

$$lmv_w = \min[\mathcal{J}(lmv_{x-1}, lmv_y), \mathcal{J}(lmv_{x+1}, lmv_y), \mathcal{J}(lmv_x, lmv_{y-1}), \mathcal{J}(lmv_x, lmv_{y+1})] \quad (8)$$

의 과정을 통해  $J_{inter}$  값을 가장 작게 만드는 위치로 이동하게 된다.

각 모드별로 삽입되는 과정을 설명하면 먼저,  $16 \times 16$  블록 모드일 경우 예측을 하기 위해 이용되는 블록은 하나이므로  $MVD$ 과  $LMVD$ 의 값은 같게 된다. 만약  $TSMVD$ 의 값이 짝수가 아니라면  $LMV_x$ 와  $LMV_y$ 는 식 (8)에 의해 수정되어진다. 다음  $16 \times 8$ 과  $8 \times 16$  블록 크기의 예측 모드일 경우에는 그림 1에서 보듯이  $16 \times 8$  모드에서는 3번 블록,  $8 \times 16$  모드에서는 5번 블록이 예측 과정에서의 마지막 블록에 해당되어 숨김 정보 삽입이 수행된다. 마지막으로,  $P8 \times 8$ 블록 크기 예측에서는 삽입 방법이 더욱 분할되어 복잡해진다. [그림 1]과 같이 현재 부호화되는 매크로블록은 4개의  $8 \times 8$ 블록으로 나누어지고 각각은  $8 \times 8$ ,  $8 \times 4$ ,  $4 \times 8$ 과  $4 \times 4$ 블록의 여러 크기로 예측이 이루어지게 된다. 그러므로 숨김 정보가 삽입되는 위치는 규칙에 의해 마지막  $8 \times 8$ 블록에 적용되고, 그 중 "S1", "S3", "S5" 또는 "S9"에서 행해진다.

이와 같은 과정을 통해 최종적으로 부호화되게 되는  $MVD$ 값은 모든 매크로블록이 짝수값이 되며

$$J_{inter} = SSDs_r(mv_w) + \lambda_{motion} \cdot R_{motion}(mv_w, REF) \quad (9)$$

와 같이 계산되는 새로운 화면간 울-왜곡 비용값을 갖게 된다.

제안하는 알고리즘에서 쓰인 움직임 벡터에 대한 강제 짝수화의 대안으로 강제 홀수화도 생각해볼 수 있다. 하지만, 이 방법은 너무나도 큰 단점을 지니고 있다. 만약  $TSMVD$ 의 값을 강제로 홀수가 되도록 정보 숨김 과정이 이루어지면 H.264/AVC의 비트량 감소에 큰 역할을 하는 스킵 모드 발생이 불가능해진다.  $16 \times 16$ 블록 화면간 예측에서 예측된  $MV$ 와 탐색으로 찾아낸  $MV$ 값이 같다면  $MVD$ 는 0인 짝수값을 갖는데, 이를 홀수화시키면 스킵 모드의 조건을 만족하지 못하기 때문이다.

이는 곧 부호화 과정에서 발생하는 총 비트량을 크게 한다. 움직임이 적은 영상에서는 스킵 모드의 발생량이 약 90%를 차지하므로 좋은 대안이 될 수 없다.

## 2. 제안하는 알고리즘의 요약

제안하는 알고리즘의 삽입과정을 [그림 2]의 흐름도를 통해 나타내었다. 알고리즘의 과정을 정리하면 다음과 같다. 먼저 현재 부호화되는 매크로블록에서 각 화면내 모드에서 발생하는  $TSMVD$ 값을 계산한다. 만약, 현재 예측 모드에서 발생된  $MVD$ 의 총합이 짝수가 되면 삽입 규칙에 의해 아무런 수정없이 울-왜곡 비용을 계산하게 된다. 홀수가 되면 제안하는 삽입 규칙에 의해 현재 예측 블록의 마지막 블록의  $MV$ 값이 수정되어진다. 그리고 수정된  $MV$ 에 의해 발생된 울-왜곡 비용을 계산하는 과정을 수행하게 된다. 화면간 예측 모드의 마지막 단계인  $P8 \times 8$  화소 블록 중  $4 \times 4$  화소 예측 과정까지 마치게 되면 식 (9)에 의해 정의되는 가장 적은 화면간 울-왜곡 비용값을 갖는 모드와 식 (4)에 의한 정보 숨김이 없는 일반적인 화면내 예측 과정까지의 계산을 모두 끝낸 후 각 모드의 비용값을 비교하여 최적 모드 결정에 적용하게 된다.

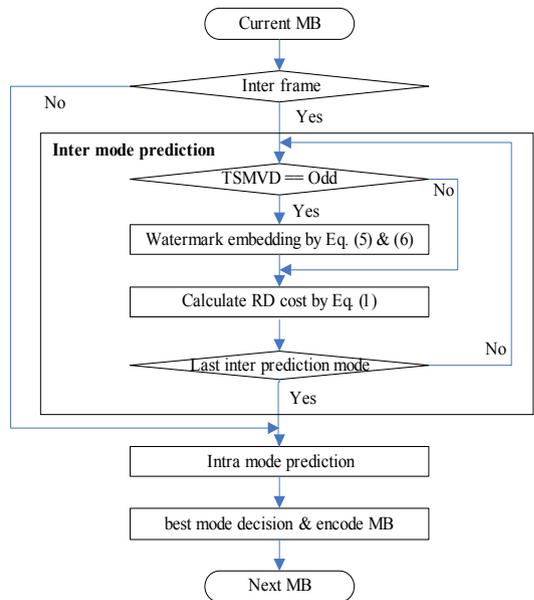


그림 2. 제안하는 알고리즘의 흐름도

### 3. 복호기에서의 오류 검출과 오류 은닉

복호기에서는 원본 영상을 필요로 하지 않고, 단지 비트열로 전송되어 복호된 TSMVD의 값이 짝수인지 홀수인지만을 체크함으로써 보다 쉽고 정확히 오류를 검출할 수 있다. 만약 TSMVD의 값이 홀수이면 현재 복호되는 매크로블록에서 전송 오류가 발생된 것으로 간주하게 되므로, 오류 검출을 매크로블록 단위에서 정확히 처리할 수 있다.

오류 검출 과정에서 스킵 모드는 특별 처리가 필요하다. 이는 몇 개의 비트만으로 여러 개의 매크로블록 정보를 담고 있는 구문이기 때문이다. 만약 스킵 모드의 구문이 끝난 바로 다음 매크로블록이 현재 오류가 검출된 지점이라면 스킵 모드로 복호된 매크로블록 또한 오류가 발생된 매크로블록으로 결정하고 은닉이 수행되도록 하였다. 오류 은닉을 위한 방법으로 간단한 TR (Temporal Replacement) 알고리즘을 사용하였다 [10][11].

## IV. 실험 결과

제한한 오류 검출 성능을 확인하기 위해, 본 논문에서는 JVT 참조 소프트웨어인 JM 12.2[12]을 이용하여 실험하였다. 실험을 위해 QCIF 사이즈의 9개 비디오 영상을 사용하였다. 실험 환경은 [표 1]과 같이 설정하였다. 오류 검출 방법의 성능을 평가하기 위해 WCDMA 네트워크상의 화상 통화 서비스 환경을 가정하였다. 숨김 정보가 삽입된 H.264/AVC비트열은 H.324M 시뮬레이터 [13]에 의해 오류가 발생하도록 하였다.

표 1. 실험 환경

| 실험 변수       | 설정          |
|-------------|-------------|
| 프로파일        | Baseline    |
| 울-왜곡 기법     | 활성화         |
| 영상 부호화 형태   | IPPP (300장) |
| 프레임율        | 30fps       |
| 탐색 영역       | 32          |
| 양자화 파라미터    | 28          |
| 비트 오류율(BER) | $10^{-4}$   |
| 참조 프레임 수    | 1           |

각 알고리즘에 따라 재구성된 이미지의 평균 PSNR과 각 방법이 삽입된 비트열의 총 패킷 중 오류가 발생된 패킷률을 비교하여 [표 2]와 [표 3]에 각각 정리하였다.

[표 2]는 각 알고리즘에 의하여 발생된 영상들의 PSNR을 아무것도 삽입하지 않는 영상들과 비교한 것이다. 제안한 알고리즘에 의한 PSNR이 다른 방법들과 비교하여 부호화를 하였을 경우 그 화질 저하가 -0.086 db로서 가장 적게 발생한 것을 확인할 수 있다.

[표 3]에서의 각 영상별 패킷수를 나타낸 것인데, 발생 패킷수에 차이가 있는 것은 숨김 정보 삽입에 따른 총 발생 비트량이 발생 패킷수에 영향을 주었기 때문이다. 또한 각 전송 비트열마다 갖고 있는 정보가 다르기 때문에 각 알고리즘은 서로간의 비교를 하는 것은 무의미하다. 그러므로 구문 검출 방법과의 검출 성능과 비교하게 된다. [표 4]에서 보듯이 제안하는 오류 검출 알고리즘(FEMVD)은 다른 데이터 숨김 삽입 알고리즘인 FEW(Forced Even Watermarking)[6]이나 RBW (Relation Based Watermarking)[7]보다 나은 성능을 보임을 확인할 수 있었다. 오류 검출률은 CSED에 비해 12.62%가 증가하였는데, 모든 알고리즘은 기본적으로 구문 검출법이 같이 수행하도록 적용하였다. 이는 구문 검출법이 적용되지 않으면 각각의 데이터 숨김 알고리즘조차 일반적 복호 과정에서 오류로 간주하게 되는 경우가 발생하기 때문이다.

표 2. 숨김 정보가 삽입된 영상별 PSNR

| 영상         | 각 알고리즘별 평균 PSNR |                    |                    |                    |
|------------|-----------------|--------------------|--------------------|--------------------|
|            | Original        | FEMVD              | FEW                | RBW                |
| Salesman   | 35.56           | 35.42              | 35.12              | 35.47              |
| Coastguard | 34.07           | 34.03              | 32.89              | 33.98              |
| Grandma    | 36.56           | 36.46              | 36.19              | 36.45              |
| Claire     | 39.94           | 39.83              | 39.41              | 39.73              |
| Mot & Dau  | 36.39           | 36.36              | 35.72              | 36.28              |
| Silent     | 35.72           | 35.59              | 35.30              | 35.58              |
| Akiyo      | 38.29           | 38.24              | 37.76              | 38.17              |
| Foreman    | 35.87           | 35.77              | 35.37              | 35.71              |
| Carphone   | 36.58           | 36.51              | 35.71              | 36.43              |
| Average    | 36.553          | 36.467<br>(-0.086) | 35.941<br>(-0.612) | 36.422<br>(-0.131) |

표 3. 총 패킷당 오류 발생률 비교

| 영상         | 패킷 발생 수 (예러/ 패킷) |        |        |
|------------|------------------|--------|--------|
|            | FEMVD            | FEW    | RBW    |
| Salesman   | 8/303            | 7/303  | 10/303 |
| Coastguard | 29/303           | 20/303 | 24/303 |
| Grandma    | 3/305            | 6/303  | 7/304  |
| Claire     | 4/303            | 3/303  | 6/303  |
| Mot & Dau  | 11/307           | 9/303  | 8/305  |
| Silent     | 11/304           | 12/303 | 13/303 |
| Akiyo      | 4/304            | 6/303  | 5/304  |
| Foreman    | 15/304           | 18/303 | 19/304 |
| Carphone   | 20/311           | 20/303 | 19/308 |

표 4. 오류 검출률 비교

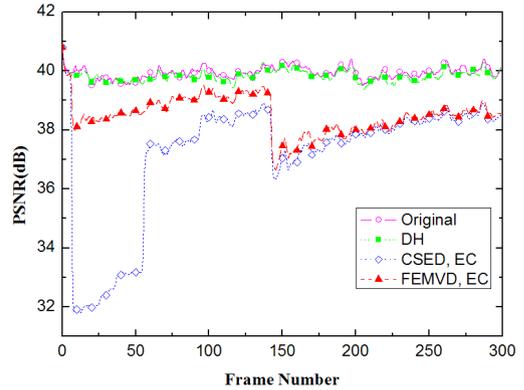
| 영상         | 검출률 (%)  |                   |          |                  |          |                  |
|------------|----------|-------------------|----------|------------------|----------|------------------|
|            | Proposed |                   | Ref. [6] |                  | Ref. [7] |                  |
|            | CSED     | FE-MVD            | CSED     | FEW              | CSED     | RBW              |
| Salesman   | 75       | 87.5              | 85.7     | 100              | 70       | 70               |
| Coastguard | 75.8     | 86.2              | 70       | 80               | 75       | 83.3             |
| Grandma    | 66.6     | 100               | 100      | 100              | 85.7     | 85.7             |
| Claire     | 75       | 75                | 66.6     | 66.6             | 83.3     | 83.3             |
| Mot & Dau  | 72.7     | 81.8              | 66.6     | 66.6             | 62.5     | 75               |
| Silent     | 72.7     | 90.9              | 66.6     | 66.6             | 69       | 69               |
| Akiyo      | 75       | 100               | 66.6     | 66.6             | 100      | 100              |
| Foreman    | 73.3     | 73.3              | 88.8     | 86.8             | 68       | 73.6             |
| Carphone   | 90       | 95                | 80       | 80               | 73.6     | 73.6             |
| Average    | 75.12    | 87.74<br>(+12.62) | 76.76    | 79.46<br>(+2.70) | 76.34    | 79.27<br>(+2.93) |

표 5. 은닉된 영상별 PSNR 비교

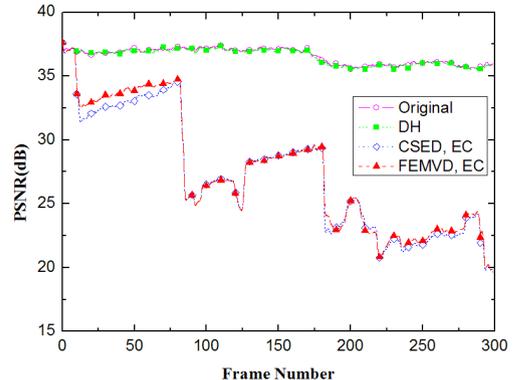
| 영상         | 은닉된 영상의 평균 PSNR |                 |          |                 |          |                 |
|------------|-----------------|-----------------|----------|-----------------|----------|-----------------|
|            | Proposed        |                 | Ref. [6] |                 | Ref. [7] |                 |
|            | CSED            | FE-MVD          | CSED     | FEW             | CSED     | RBW             |
| Salesman   | 30.45           | 31.03           | 32.34    | 32.63           | 30.80    | 30.92           |
| Coastguard | 22.84           | 23.55           | 23.29    | 23.44           | 20.27    | 21.49           |
| Grandma    | 35.92           | 36.17           | 35.78    | 35.78           | 34.00    | 34.00           |
| Claire     | 37.13           | 38.51           | 38.77    | 38.77           | 34.15    | 34.71           |
| Mot & Dau  | 31.94           | 32.23           | 27.83    | 27.83           | 28.84    | 29.62           |
| Silent     | 32.49           | 32.75           | 31.15    | 31.15           | 29.96    | 30.02           |
| Akiyo      | 33.97           | 37.54           | 32.98    | 33.01           | 34.68    | 34.78           |
| Foreman    | 26.56           | 26.92           | 21.46    | 21.49           | 23.46    | 24.08           |
| Carphone   | 27.33           | 27.56           | 24.58    | 24.94           | 24.92    | 25.16           |
| Average    | 30.96           | 31.80<br>(0.85) | 29.80    | 29.89<br>(0.10) | 29.01    | 29.42<br>(0.41) |

[표 5]는 [표 4]의 결과를 통하여 오류가 검출된 영역에 오류 은닉 방법이 적용되어 재구성되어진 영상들의 PSNR을 비교한 것이다. 제안된 방법에 의한 향상된 오류 검출률은 보다 정확한 오류 지점을 찾아내게 되고,

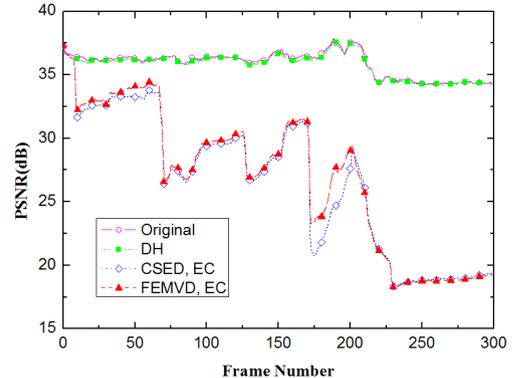
이것은 오류가 검출되는 것이 지연되지 않는 은닉 영상의 재구성을 할 수 있도록 도와준다. 그리하여 검출률이 높아진 결과는 그만큼 정확히 복호된 영상의 정보를 유지하게 하여 은닉 영상의 PSNR 또한 높게 만드는 것을 알 수 있다.



(a) Claire



(b) Carphone



(c) foreman

그림 3. 영상별 PSNR 비교



그림 4. 주관적 화질 비교

[그림 3]과 [그림 4]는 제안된 알고리즘이 적용된 Claire, Carphone, 그리고 foreman 영상들에 대한 실험 결과를 보여준다.

[그림 3]에서는 각 프레임별 발생된 PSNR의 그래프를 나타내며, [그림 4]는 주관적 화질을 비교한 것이다. [그림 4]의 각 상단은 구문 검출법인 CSED에 의한 에러 검출, 에러 은닉, 에러 전파를, 하단은 제안한 알고리즘에 의한 결과를 보여준다. 동그라미 표시는 검출되지

못하고 에러 은닉 후 재구성 이미지에 남아 화질을 떨어뜨리는 결과를 가져오는 영역을 표시한 것이다.

[그림 3](a)의 Claire 그래프에서 7번째 프레임에서 발생된 패킷 오류는 급격한 PSNR 하락을 발생시켰다. 이미지로 확인하면 [그림 4](a)의 동그라미 부분에 의해 나타난 것이다. 은닉을 하지 않을 경우 각 영상의 좌측 이미지에서 보는 것과 같이 초록색영역으로 된 부분이 영상 데이터를 복호하지 못하고 있음을 알 수 있다.

[그림 4](a)에서 좌측은 오류가 검출된 7번째 프레임을, 중간은 은닉이 수행된 7번째 프레임, 우측은 은닉된 영상의 오류가 전파된 21번째 프레임을 나타낸다. 하지만, 제안한 알고리즘에 의해 이 동그라미로 표시된 오류가 검출되어 은닉 영상 후 전파 과정에서 화질 열화가 크게 생기지 않음을 확인할 수 있다. 오류가 정확히 검출된 것을 좌측 이미지에서 보면 제안한 알고리즘 방법이 오류 지점 검출의 지연이 없으므로 더 많은 초록색 영역이 표시됨을 알 수 있다.

두 번째 Carphone 영상 결과에서 왼쪽, 중간 이미지는 각각 오류가 발생되어 검출되고 은닉된 182번째 프레임을 나타내며, 오른쪽 이미지는 오류가 전파된 184번째 프레임을 보이고 있다. 구문 검출법에서 검출하지 못한 잘못된 모드 및 계수 복호된 매크로블록을 제안한 방법에서는 검출하여 제대로 은닉하고 있음을 확인할 수 있다.

세 번째 영상인 foreman에서는 100번째 오류 검출/은닉 프레임과 오류가 전파된 110번째 프레임을 보여준다. foreman에서는 구문 검출법에서는 오류를 검출하지 못한 DCT 계수의 잘못된 복호가 일어났고, 이는 프레임들의 시간적 경과가 일어난 후에도 계속적으로 남아 있게 되어 PSNR을 떨어뜨리는 결과를 낳게 되었다. 하지만, 제안한 방법에 의해서 foreman 영상 역시 좀 더 제대로 재구성된 은닉된 이미지가 만들어진 것을 확인할 수 있었다.

이런 실험 결과를 통해 오류가 발생된 지점의 정확한 검출은 오류 은닉 과정을 통해 사용자가 만족할 만한 화질을 제공할 수 있는 방법임을 알 수 있다.

## V. 결론

본 논문에서는 H.264/AVC의 화상회의/통화를 위한 프로파일인 베이스라인 프로파일에서의 움직임 벡터 영역의 정보를 이용한 새로운 오류 검출 알고리즘을 제안하였다. H.264/AVC의 화면간 예측에서 사용된 다양한 블록 크기의 움직임 예측의 영향을 분석하고, 이를 통해 화면간 부호화가 이루어지는 매크로블록에서 발생하는 마지막 블록의 움직임 벡터값에 숨김 정보를 삽입하였다. 이를 통해 보다 향상된 오류 검출률을 가져올 수 있었으며 부호화 과정에서의 PSNR 저하도 최소로 막을 수 있었다. 또한 높은 검출률을 바탕으로 나아진 화질의 은닉 이미지를 얻었다. 제안된 알고리즘은 실험 결과를 통해 기존의 구문 검출 알고리즘보다 정확한 전송 오류 위치를 검출할 수 있음을 확인하였다. 비록 부호화 과정에서 숨김 정보를 삽입함으로써 평균 0.09dB의 손실이 생겼지만, 제안된 알고리즘의 적용에 의해 정확한 오류 검출이 이루어져 복호기측에서는 약 0.85dB의 은닉 PSNR 상승 효과를 얻게 되었다.

## 참고 문헌

[1] Information Technology-Coding of Audio-Visual Object-part 10: Advanced Video Coding, Final Draft International standard, ISO/IEC 14496-10, 2005(3).

[2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. on Circuit and Systems for Video Technology*, Vol.13, No.7, pp.506-576, 2003(7).

[3] Y. Wang and Q. Zhu, "Error control and concealment for video communication: a review," *Proc. IEEE*, Vol.86, No.5, pp.974-997, 1998.

[4] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding

techniques," *IEEE Signal Processing Magazine*, Vol.17, No.4, pp.64-82, 2000.

- [5] L. Superiori, O. Lemethova, and M. Rupp, "Performance of a H.264/AVC error detection algorithm based on syntax analysis," in *Proc. Int. Conf. Advance Mobile computing Multimedia*, pp.1-10, 2006.
- [6] M. Chen, Y. He, and R. L. Legendijk, "A fragile watermark error detection scheme for wireless video communications," *IEEE Trans. on Multimedia*, Vol.7, No.2, pp.201-211, 2005.
- [7] O. Nemethova, G. C. Forte, and M. Rupp, "Robust error detection for H.264/AVC using relation based fragile watermarking," *Proc. Int. Conf. on Systems, Signals, and Image Processing*, 2006.
- [8] Y. Liu, Y. K. Liu, S. J. Qiao, and P. Wei, "Error detection For H.264/AVC coded video based on artifact characteristics," *IEEE Intelligent Computing and Intelligent Systems*, Vol.3, pp.293-298, 2010.
- [9] 김남득, "디지털 워터마킹 기술소개 및 동향보고", KOSEN/OSTIN, pp.1-10, 2001.
- [10] Y. K. Wang, M. M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The error concealment feature in the H.26L test model," in *Proc. of IEEE Int. Conf. on Image Processing*, pp.729-732, 2002.
- [11] 손남례, "H.264/AVC을 위한 움직임 벡터 복원 방법", *한국콘텐츠학회논문지*, 제8권, 제5호, pp.29-37, 2008.
- [12] Joint Video Team, JVT Reference software Encoder, version 12.2.  
<http://iphome.hhi.de/suehring/tml/download>.
- [13] ITU-T, SG 16, Video Coding Experts Group(Question 15), Mobile Mux Simulation Tools, Q15\_I\_53.doc.

저 자 소 개

고 만 근(Man-Geun Ko)

정회원



- 2007년 2월 : 충북대학교 전기전자컴퓨터공학부(공학사)
- 2007년 3월 ~ 현재 : 충북대학교 전자공학과(공학석사)
- 2009년 3월 ~ 현재 : 충북대학교 전자공학과 박사과정

<관심분야> : 오류 검출, 오류 은닉, 멀티미디어, H.264/AVC, HEVC

서 재 원(Jae-Won Suh)

정회원



- 1995년 2월 : 충북대학교 전자공학과(공학사)
- 1997년 2월 : 광주과학기술원 정보통신공학과(공학석사)
- 2003년 2월 : 광주과학기술원 정보통신공학과(공학박사)

- 2003년 4월 ~ 2004년 8월 : LG전자기술원 MM연 선임 연구원
- 2004년 9월 ~ 현재 : 충북대학교 전기전자컴퓨터공학부 부교수

<관심분야> : 디지털 영상신호처리, 동영상 표준화, H.264/AVC, SVC, HEVC