

# 워터마크를 이용한 전송 채널 에러의 검출 및 은닉

정희원 박운기\*, 정희원 전병우\*\*

## Error Detection and Concealment of Transmission Error Using Watermark

Woonki Park\* Byeungwoo Jeon\*\* *Regular Members*

### 요약

전송 채널에서의 잡음은 시간적으로 연속적인 동영상의 정상적인 복원을 어렵게 하고, 수신측의 복원 영상의 질을 크게 저하시키므로 전송 채널 에러를 정확하게 검출하여 에러에 대한 영향을 최소화 할 수 있는 방법이 필요하다. 본 논문에서는 전송 채널에 의해 발생한 오류를 찾아내고 제거할 수 있도록 전송 데이터 내에 워터마크를 삽입하는 방법을 제안한다. 제안된 방법은 부호화기에서 동영상 압축 데이터의 일부분인 양자화된 DCT 계수의 LSB에 특정한 패턴을 삽입하며, 복호화기에서는 이러한 LSB의 특정한 패턴을 검사함으로써 전송된 데이터의 신뢰성을 확인할 수 있다. 또한 에러에 의해 손상된 부분을 효과적으로 복원하기 위해 움직임 추정 벡터의 유사도 정보를 삽입함으로써 에러가 발생된 블록의 복원 능력을 향상시킬 수 있도록 한다. 제안하는 방법은 에러에 쉽게 노출되는 무선 환경 하의 휴대용 단말기와 같은 동영상 전송·수신 시스템에 효과적으로 사용될 수 있다.

Key Words : Error detection, Error concealment, Watermark, Video compression

### ABSTRACT

There are channel errors when video data are transmitted between encoder and decoder. These channel errors would make decoded image incorrect, so it is very important to detect and recover channel errors. This paper proposes a method of error detection and recovery by hiding specific information into video bitstream using fragile watermark and checking it later. The proposed method requires no additional bits into compressed bitstream since it embeds a user-specific data pattern in the least significant bits of LEVELs in VLC codewords. The decoder can extract the information to check whether the received bitstream has an error or not. We also propose to use this method to embed essential data such as motion vectors that can be used for error recovery. The proposed method can detect corrupted MBs that usually escape the conventional syntax-based error detection scheme. This proposed method is quite simple and of low complexity. So the method can be applied to multimedia communication system in low bitrate wireless channel.

### I. 서론

최근 무선 채널에서의 동영상 데이터의 통신이

활발하게 이루어지고 있다. 일반 문자 데이터에 비해 상대적으로 데이터량이 매우 큰 동영상 데이터는 H.261, H.263, MPEG 등과 같은 방법을 통해 압축이 되는데, 이 경우 압축 비트스트림의 한 비트

\* 삼테크아이앤씨 기술연구소(friend10@samtechic.co.kr), \*\*성균관대학교 정보통신공학부(bjeon@yurim.skku.ac.kr)

논문번호 : 030263-0618, 접수일자 : 2003년 6월 18일

※ 본 연구는 대학IT연구센터 육성, 지원사업의 연구결과로 수행되었음

가 의미하는 상대적 정보량이 압축 전과 비교하여 커지게 된다. 즉, 압축된 데이터에 비트 손실이 발생했을 경우, 비트 손실에 대한 영향이 압축 전의 비트 손실에 비해 매우 커지게 된다. 전송 채널 에러에 의해 압축 데이터가 손실되었을 경우, 이를 정확하게 검출하고, 손실된 부분을 복원할 수 있는 방법은 크게 에러 검출, 에러 복원, 에러 은닉 등으로 구분될 수 있다<sup>[1]</sup>. MPEG-4 비디오는 채널 에러에 대한 대책으로 재동기기술(Resynchronization Marker), 데이터구분화(Data partitioning), 역방향 VLC(Reversible VLC)와 같은 기술을 적용하고 있다<sup>[2]</sup>. 재동기 기술은 정해진 데이터량에 맞추어 17~23비트의 크기를 갖는 Resync Maker를 헤더정보로 삽입함으로써, 채널 에러의 파급 효과를 일정 부분 내로 한정할 수 있다는 장점이 있다. 데이터 구분화 기술은 Motion Header 정보와 Texture 정보 사이에 Motion Marker를 삽입하여 에러에 의해 패킷 데이터가 손상될 경우 Motion Marker를 통해 에러가 발생하지 않은 부분을 복원해 낼 수 있는 장점을 지닌다. 역방향 VLC(Reversible VLC)는 압축데이터의 대부분을 이루는 Texture 정보의 가변장 부호어가 역방향으로도 해석될 수 있도록 고안된 가변장 부호화기를 이용하여 복호화 중에 에러가 발견될 경우 뒤부터 역방향으로 해석하도록 하여 에러 발생 위치를 좀더 정밀하게 지역화하고 복호 능력을 향상시키는 기술이다.

채널 에러에 의한 영향을 최소화하기 위한 방법 또한 많은 연구가 진행되고 있는데, 데이터 인터리빙(data interleaving), forward error correction(FEC)과 채널 전송 레벨에서 에러를 검출하고 지역화(localization)하는 방법 또한 연구되고 있다<sup>[3][4]</sup>. 최근에는 압축된 비디오데이터에 워터마크를 삽입하여, 에러에 의해 손실된 부분을 검출하는 방법이 활발하게 연구되고 있다. 비디오 데이터에 워터마크를 삽입하는 방법은 반 화소단위 움직임 추정 알고리즘에서 반 화소단위에 특정 패턴을 삽입하는 방법과 양자화된 DCT 계수의 LSB에 특정 패턴을 삽입하는 방법이 있다<sup>[5]</sup>. 제안하는 방법은 양자화된 DCT 계수의 LSB에 특정 패턴을 삽입하여 에러를 검출하도록 한다.

## II. 전송 채널 에러의 영향

일반적으로 채널 에러에 의해 VLC 데이터가 잘

못 복호화 될 경우 복호화 에러로 판명되는 경우가 있는데, 이는 하나의 8×8 블록에 64개 보다 많은 DCT 계수가 존재하는 경우, 해석할 수 없는 VLC 코드가 나오는 경우 등이다. 이러한 에러 검출 방법을 구분 기반 에러 검출(Syntax-based Error Detection)이라 한다. 하지만 이러한 에러 확인 방법으로는 전체 에러를 모두 검출할 수 없다는 문제점이 있다. 즉, 복호화시 구분상 문제를 일으키지 않는 이상 에러로 검출되지 않고 지나가기 때문이다.

|  |                                   |
|--|-----------------------------------|
| <b>Case 1: Wrong sign data</b>         |                                   |
| Encoder:                               | 0101 01s(s=0)                     |
|  | (LAST=0, RUN=0, LEVEL=3 sign s=0) |
| Decoder:                               | 0101 01s(s=1)                     |
|  | (LAST=0, RUN=0, LEVEL=3 sign s=1) |
| <b>Case 2: Wrong RUN or LEVEL data</b> |                                   |
| Encoder:                               | 0101 01s                          |
|  | (LAST=0, RUN=0, LEVEL=3)          |
| Decoder:                               | 0101 00s                          |
|  | (LAST=0, RUN=1, LEVEL=2)          |
| <b>Case 3: Wrong LAST data</b>         |                                   |
| Encoder:                               | 0010 110s                         |
|  | (LAST=0, RUN=10, LEVEL=1)         |
| Decoder:                               | 0010 010s                         |
|  | (LAST=1, RUN=6, LEVEL=1)          |

그림 1. 채널 에러에 의해 왜곡되는 VLC 데이터

그림 1은 채널 에러에 의해 VLC 데이터의 해석이 가능하면서도 복원되는 내용이 잘못된 예로 sign bit가 에러에 의해 변경되는 경우, RUN과 LEVEL 데이터가 변경되어 DCT 계수를 복원할 때 원하지 않는 결과가 나오는 경우, 그리고 블록의 끝을 알려주는 LAST 데이터가 오인되어 블록간의 구별이 잘못되는 경우를 보이고 있다. 단순히 Case 1과 같이 sign bit가 잘못되는 경우는 전체 화질에 큰 문제를 일으키지 않을 수도 있으나, Case 2 또는 Case 3의 경우는 그림 2(b)와 같이 복원 영상의 화질에 큰 문제를 일으킬 수 있다.



(a) 패킷 손실, 패킷 길이=200bit, 패킷 손실률=1/100, 영상전송률=256kbps



(b) 비트 에러, 비트 에러율 1/1000, 영상전송률=192kbps

그림 2. 패킷 손실과 비트 에러에 의한 화질 저하

그림 3은 인터넷 환경 하에서 비디오 압축 데이터가 패킷화되어 전송되는 모습을 그린 것이다. 일정한 크기의 패킷으로 나누어진 압축 데이터는 전송 채널을 통해 복호화기로 전송되는데, 이때 채널 환경에 따라 패킷이 손실될 수 있다. 본 논문을 위한 예비 실험으로 패킷의 길이를 200비트로 하고, 패킷의 손실확률을 1/100으로 하였을 때의 화질을 측정해 본 결과 그림 2(a)에 나타난 바와 같이 비트 에러에 비해 상대적으로 화질 저하가 적었다. 그 이유는 패킷 손실이 발생할 경우 디코더 측에서 손실된 부분을 정확하게 판단하여 화질 저하를 줄이기 위한 에러 은닉 기술을 이용할 수 있기 때문이다. 하지만 비트 에러의 경우 그림 1에서 나타난 바와 같이 기존의 구문 분석 방법이 이를 모두 검출할 수 없기 때문에 발생한 에러를 그대로 지나쳐 에러 은닉 기술이 부분적으로 적용될 수밖에 없다. 그림 2(b)는 비트 에러가 발생하였을 때의 화질 저하를 나타낸 그림이다. 따라서 효과적인 에러 검출을 위한 방법을 제안하는 본 논문은 비트 에러 환경만을

고려하였다.

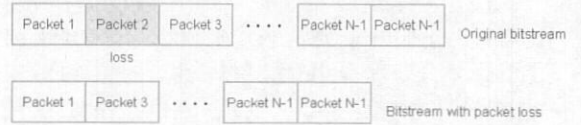


그림 3. 패킷 손실

### III. 에러 검출 워터마크 패턴 삽입 방법

구문 해석 기반의 에러 검출 방법으로는 모든 채널 에러를 검출하기가 불가능하기 때문에, 이와 더불어 채널 에러 검출 능력을 높이기 위한 방법이 필요하다. 제안하는 방법은 인트라 매크로 블록 (Intra macroblock)과 인터 매크로 블록 (Inter macroblock)에 따라 달라지는데, 기본적으로 부호화되는 블록의 양자화된 DCT 계수의 최하위비트 (LSB)에 미리 약속된 패턴을 삽입하여, 이의 변화를 관찰하는 방법이다.

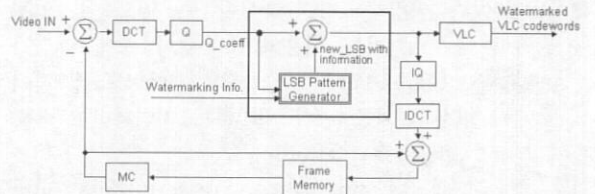


그림 4. 제안하는 워터마크 삽입 방법

그림 4는 제안하는 워터마크 삽입 방법의 블록도이다. 이는 움직임 추정과 DCT 변환, 양자화와 가변장 부호화 기술이 적용되는 일반적인 비디오 부호화기의 블록도인데, 제안하는 워터마크 삽입 방법은 양자화 후에 도출되는 양자화 계수의 LSB에 특정 정보를 삽입하여 이 정보를 복호화기로 보내주는 방법이다. 굵은 상자 내부에 있는 “LSB Pattern Generator”는 원하는 워터마크 정보와 현재의 LSB 값을 비교하여 워터마크 정보를 생성하는 부분이다. LSB 변경에 의한 원영상의 화질 저하를 최소화하기 위해 LSB 변경에 따른 화질의 변화 정도를 실험하였는데, 이를 통하여 짝수(0)보다 홀수(1) LSB로 변경할 경우 화질 저하가 적다는 것을 확인할 수 있었다. 이는 동영상 압축에서 움직임 추정을 통해 이전 시간의 영상의 차이 값을 부호화해 주게 되는데, 이때 발생하는 양자화된 DCT 계수는 ‘0’

또는 '1'이 될 확률이 높다. 만약 "LSB Pattern Generator"가 '0'이 아닌 양자화된 DCT 계수의 LSB를 '0'으로 한다면, 이 계수의 값은 반드시 '2' 또는 그 이상의 짝수이어야만 한다. 왜냐하면, 양자화된 DCT 계수가 '0'인 경우 가변장 부호화기에서 RUN으로 부호화하기 때문에 LEVEL의 값에 원하는 정보를 삽입할 수 없게 된다. 이렇게 LSB 패턴이 '0'이 되어 양자화된 DCT 계수가 '2' 또는 그 이상의 짝수가 되면 LSB 패턴 변경에 따른 원영상과의 차이가 커지게 되므로 가급적 LSB 패턴을 '1'이 되도록 워터마크 삽입 방법을 설계하였다.

1. 인트라 매크로블록에서의 워터마크 삽입 방법

인트라 매크로블록은 시간적으로 이전에 위치한 프레임의 데이터를 참조하는 움직임 추정 방법을 사용하지 않기 때문에 인트라 매크로블록에 비해 데이터의 양이 상대적으로 크다. 또한 인트라 블록의 VLC 코드가 잘못 해석되어 잘못된 LAST값이 해석되었을 경우, 뒤에 따라오는 코드들이 인트라 매크로블록임에도 불구하고 인트라 매크로블록으로 판단되는 경우가 발생하게 된다. 그림 5는 이러한 문제를 극명하게 보여주는 예이다. 즉, 경우에 따라서 구문 해석에 문제가 없으면서 복원영상에는 커다란 화질 저하를 가질 수 있는 문제점이 발생한다.



그림 5. 인트라블록이 인트라 블록으로 잘못 해석된 예

$$LSB\ pattern(Coeff\_pos) = \begin{cases} Coeff\_position\ mod\ 2 & \\ \text{when AC} \ \& \ (Coeff\_pos \geq pos\_start) & \\ 0 & \text{when DC} \end{cases} \quad (1)$$

그림 5와 같이 인트라 매크로 블록이 인트라 매크로블록으로 잘못 판단되는 등의 치명적인 문제를 막기 위해 식(1)과 같이 인트라 블록에 위치한 6개

의 DCT블록(4개의 휘도 성분 블록, 2개의 색차 성분 블록)의 맨 앞에 오는 DC 계수를 모두 짝수로 만들어 주어, LAST 비트가 잘못 판단되어 나타나는 오류를 검출할 수 있도록 하였다. 즉 복호화된 인트라 블록의 DC계수가 홀수인 경우에는 에러가 발생했음을 알 수 있다. 또한 AC계수의 경우는 zigzag정렬 후에 나열되는 순서에 따라 에러 검출 패턴을 삽입하게 되는데 미리 정하여진 위치(pos\_start)부터 나타나는 0이 아닌 AC값들의 LSB를 그 해당 계수의 블록 내 주사(scan) 위치(1-63)를 나타내는 인덱스의 LSB값과 동일하게 맞추어 주도록 하였다. 즉, 해당 계수가 만일 주사 순서로 홀수번째라면, 그 계수의 LSB가 1이 되도록 변경한다. 식(1)에서 Coeff\_position은 계수들이 존재하는 위치를 의미한다. 이와 같은 방법으로 에러 검출 패턴을 만든 이유는 그림 1의 Case 2에서 볼 수 있듯이 에러에 의해 RUN과 LEVEL에 오류가 생겼을 때 이를 정확하게 검출하기 위함이다. 한편 모든 AC계수들의 LSB를 Coeff\_position에 따라 정할 경우 에러 검출 능력은 향상되는 반면에, 압축되는 영상의 화질 저하를 수반하기 때문에 pos\_start값을 정하여 이 위치보다 낮은 주파수 성분에는 삽입이 되지 않도록 한다. 본 논문의 실험에서는 pos\_start 값을 11로 설정하여 실험하였다.

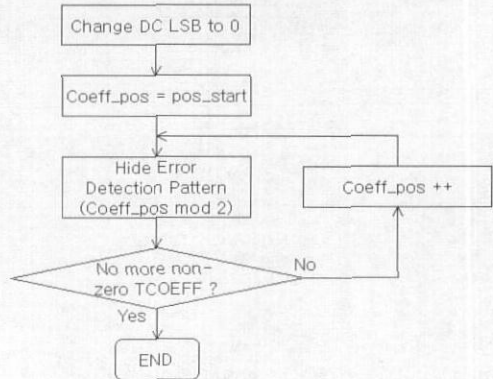


그림 6. 인트라 블록의 워터마크 삽입 순서도

그림 6은 인트라 블록에서 채널 에러 검출을 위한 패턴을 삽입하는 과정을 나타낸 순서도이다. 일단 DC의 LSB를 '0'으로 변경시키고 pos\_start 이후에 나타나는 AC계수에 에러 검출 패턴을 삽입하게 된다.

2. 인트라 매크로블록에서의 워터마크 삽입 방법

인트라 매크로블록과는 달리 인트라 매크로블록은 시간에 따른 영상의 유사성을 이용하여 이전 프레임과의 움직임 추정과정을 수행한 후 움직임 벡터 정보를 보내주고, 이어서 그 차이값을 DCT, 양자화, 가변장 부호화 과정을 통하여 압축하게 된다. 따라서 인트라 매크로블록에 비해 상대적으로 데이터량이 작아지게 되고, 대부분의 양자화된 AC계수들이 '0' 또는 '1'의 값만을 갖게 된다. 양자화된 AC계수가 '0'인 경우는 RUN의 변수로 저장되어므로, '0'인 경우는 그대로 두고 AC계수들이 '0'이 아닌 경우를 고려하여 워터마크 정보를 삽입해야 한다. 그림 7은 인트라 매크로블록에 에러 검출 정보를 삽입하는 과정을 나타내는 순서도이다. 인트라 매크로블록에서 처리하는 방법과 구별되는 점은 인트라 매크로블록에서 에러 검출 패턴을 삽입하는 단위가 블록단위로 이루어지는데 반해 인트라 매크로블록에서는 매크로블록 단위로 이루어진다. 그 이유는 인트라 매크로블록의 경우 움직임 추정 과정의 결과이기 때문에 매크로블록을 구성하는 6개의 블록들의 상당수가 존재하지 않고, 존재하더라도 데이터량이 매우 적게 되어 '0'이 아닌 AC 계수들이 그리 많이 존재하지 않기 때문이다.

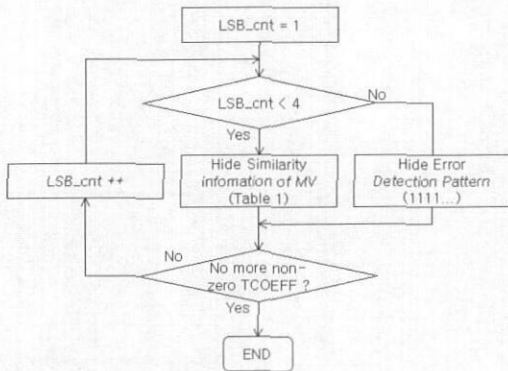


그림 7. 인트라 매크로블록의 워터마크 삽입 순서도

우선 인트라 매크로블록의 처음 4개의 '0'이 아닌 AC계수에는 에러 검출 패턴이 아닌 이웃 블록과의 움직임 벡터 유사도 정보를 삽입하게 된다. 4비트의 움직임 벡터 유사도 정보를 삽입하는 이유는 현재 부호화중인 블록과 이웃한 매크로블록이 채널 잡음에 의해 손실되었을 경우, 현재 매크로블록이 갖고 있는 이웃 매크로블록에 대한 움직임 유사도 정보

를 기반으로 하여 손실된 매크로블록을 복원할 수 있도록 하기 위해서이다.

$$\text{if } |MV_{curr} - MV_{neighbor}| \leq THD \text{ then embed '1' } \quad (2)$$

$$\text{else embed '0'}$$

표 1. 인트라 매크로블록의 4비트의 움직임 유사도 정보

| neighbor | 움직임 유사도 정보 위치 |
|----------|---------------|
| Bottom_x | First bit     |
| Bottom_y | Second bit    |
| Right_x  | Third bit     |
| Right_y  | Fourth bit    |

식(2)는 움직임 벡터의 유사도를 결정하는 식이다. 현재 매크로블록의 움직임 벡터와 이웃한 매크로블록의 움직임 벡터의 차이값이 임계값 THD보다 작거나 같을 경우 해당되는 위치의 AC계수의 LSB를 '1'로 결정하고, 그렇지 않을 경우 '0'으로 결정하게 된다. 표 1은 4비트의 움직임 유사도 정보를 나타내는 표이다. 첫 번째 비트에는 아래에 있는 매크로블록과 현재 매크로블록의 수평 움직임 벡터값을 식(2)를 이용하여 결정하고, 두 번째 비트는 아래에 있는 매크로블록의 수직 움직임 벡터값과 비교를 하여 결정한다. 또한 세 번째와 네 번째 비트는 오른쪽에 있는 매크로블록의 수평 움직임 벡터값과 수직 움직임 벡터값을 이용하여 결정한다. 현재 매크로블록과 이웃한 4개의 매크로블록 중 아래쪽과 오른쪽에 위치한 매크로블록을 이용한 이유는 현재 삽입하고 있는 유사도 정보가 부호화시 아래쪽과 오른쪽의 매크로블록에서 사용이 가능하도록 하기 위함이다. 즉 부호화기 입장에서 현재 매크로블록이 오른쪽과 아래쪽에 위치한 매크로블록에 비해 시간적으로 앞선 위치에 있는 데이터이기 때문이다. 따라서 부호화 시에 움직임 벡터 유사도 정보를 삽입하기 위해서는 아래에 위치한 매크로블록의 움직임 정보를 참고하기 위해 하나의 매크로블록 줄에 대한 지체를 수반하게 된다. 또한 움직임 벡터의 유사도를 수평과 수직 요소로 나눈 이유는 각각의 정보를 다른 위치에 존재하는 매크로블록의 유사도 정보를 이용하도록 하여 손상된 매크로블록의 복원 정확도를 높이기 위함이다.

4비트의 움직임 벡터 유사도 정보를 인트라 매크로블록의 양자화된 AC계수에 삽입한 후에 나타나는

AC계수에는 무조건 '1'의 LSB 패턴을 만들어주게 된다. 인트라 매크로블록에서 계수의 위치에 따른 상대적인 검출 패턴 대신에 무조건 '1'의 패턴을 삽입한 이유는 LSB 패턴이 '1'이 되는 경우가 '0'으로 만드는 경우보다 원래 데이터의 화질 보존 측면에서 유리하다는 것이 실험을 통해 분석되었기 때문이다.

3. 에러 검출 및 복원

그림 8은 일반적인 부호화기에서 사용하는 구문 분석적인 방법을 통해 채널 에러를 검출할 때 에러가 검출되는 GOB와 그렇지 않은 GOB를 나타낸 그림이다. 구문 분석을 통한 에러 검출 방법은 하나의 8x8 블록에 64개의 DCT 계수가 존재할 경우, 해석이 불가능한 코드워드가 존재할 경우 등이 있는데, 그림 8(a)는 이러한 검출 방법으로 에러가 검출되지 않은 매크로블록이 존재할 경우의 GOB를 나타낸 그림이다. 6번째 매크로블록의 데이터에 에러가 발생했음에도 불구하고 다른 의미로 복호화가 가능하기 때문에 구문 분석 방법을 통해 에러 검출이 불가능한 경우이다.

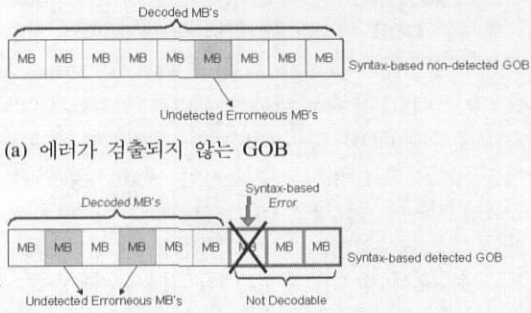


그림 8. 구문 분석으로 에러 검출을 하는 경우

또한 그림 8(b)는 구문 분석 방법을 통하여 에러가 검출된 GOB를 나타낸 것이다. 그림 8(b)의 두 번째와 네 번째 매크로블록은 에러에 의해 손실이 되었지만, 구문 해석을 통한 에러 검출이 되지 않았기 때문에 그대로 복호화되고 있는 것을 나타내고, 일곱 번째 매크로블록은 구문 분석을 통해 에러가 검출된 것을 뜻한다. 일곱 번째 매크로블록 이후의 매크로블록들은 복호화가 불가능하기 때문에 버려지고, 복호화기는 다음 GOB의 시작 코드 정보를 찾아 새롭게 복호화를 시작하게 된다.

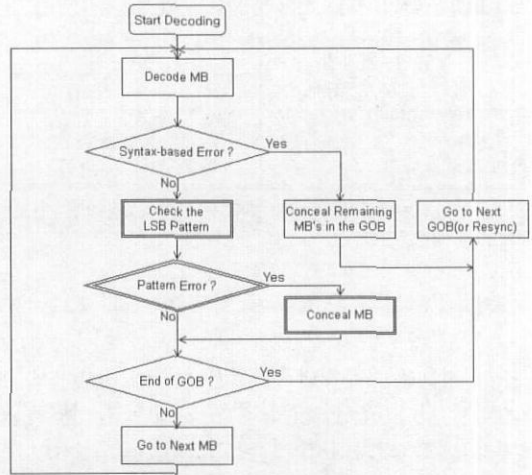


그림 9. 제안하는 에러 검출 및 은닉 방법의 순서도

그림 9는 본 논문에서 제안하는 매크로블록 단위의 에러 검출 및 은닉 방법에 대한 순서도를 나타낸 것이다. 우선 구문 분석 방법을 통해 검출된 매크로블록과 그 이후의 같은 GOB 내의 매크로블록은 기존의 부호화기에서 처리한 바와 같이 복호 자체가 불가능하기 때문에 버려지게 된다. 하지만 구문 분석을 통해 검출은 되지 않았지만 에러 검출 패턴 정보의 비교를 통해 에러가 검출된 매크로블록의 경우는 이후에 전송되는 데이터들이 계속적으로 복호화가 가능하기 때문에, 에러가 검출된 매크로블록만 에러 은닉 과정을 수행하고, 이후의 데이터는 계속해서 복호화를 수행하게 된다. 즉, 그림 8(a)의 여섯 번째 매크로블록 및 그림 8(b)의 두 번째, 네 번째 매크로블록의 경우 에러 검출 패턴의 비교를 통해 에러에 의한 손실 여부를 판단할 수 있게 되기 때문에 이 부분에 나타난 에러를 은닉하여 복호화된 화질의 향상을 꾀할 수 있게 된다. 물론 에러 검출 패턴을 통해 손상된 매크로블록을 검출해 냈을 경우 그 이후의 매크로블록도 함께 버리는 방법도 택할 수 있으나, 이 경우 복원 가능한 매크로블록까지 소실되어 복원된 영상의 화질이 더 나빠지는 경우가 발생할 수 있다.

표 2는 그림 9의 방법을 이용하여 검출된 에러에 의해 손실된 매크로블록을 복원할 때, 인트라 매크로블록에 삽입했던 4비트의 움직임 벡터 유사도 정보를 이용하기 위해 우선 순위를 정한 표이다. 손실된 매크로블록의 상단 또는 좌측에 존재하는 매크로블록이 인트라 매크로블록일 경우에만 해당이 되며, 두

표 2. 움직임 벡터 선택의 우선 순위

| 삽입된 움직임 벡터 유사도 정보 |   | 상단 매크로블록 |     |
|-------------------|---|----------|-----|
|                   |   | 1        | 0   |
| 좌측 매크로블록          | 1 | 위쪽       | 왼쪽  |
|                   | 0 | 위쪽       | 위쪽* |

\*은 일반적인 에러 은닉방법에서 위쪽의 움직임 정보를 택하는 것을 뜻함

개의 이웃한 매크로블록이 모두 인트라 모드로 코딩이 된 경우는 움직임 유사도 정보를 이용하여 움직임 정보를 유추해낼 수 없기 때문에 (0,0)으로 복원하게 된다. 우선 유사도의 우선 순위를 상단에 위치한 매크로블록에 주게 되고, 이 매크로블록이 아래쪽에 위치한, 즉 현재 손실된 매크로블록과의 수평·수직 움직임 정보 유사도 2비트를 확인하여 '1'로 되어 있는 경우 이 매크로블록의 움직임 벡터를 손실된 현재 매크로블록의 움직임 정보로 사용하게 된다. 만약 상단 매크로블록의 유사도 정보가 '0'으로 되어있을 경우 좌측에 존재하는 매크로블록이 갖고 있는 현재 매크로블록과의 움직임 벡터 유사도 정보를 확인하여 '1'로 되어 있는지를 확인하여 움직임 벡터를 가져오게 된다. 만약 이웃한 두 개의 매크로블록으로부터 가져온 유사도 정보가 모두 '0'일 경우 기존의 일반적인 복호화기에서 사용하는 위쪽에 위치한 매크로블록의 움직임 벡터를 가져오게 된다. 본 논문에서 제안하는 움직임 벡터 유사도 정보를 이용하여 손실된 매크로블록을 복원하는 경우, 움직임 벡터의 수평·수직 요소를 각각 다른 매크로블록에서 가져올 수 있기 때문에 이전보다 정확한 매크로블록의 복원이 가능하게 되는 이점이 있다.

IV. 실험 결과 및 분석

본 실험은 인텔 펜티엄III 550Mhz, 마이크로소프트 윈도우즈 2000 운영체제 시스템 환경 하에서 수행되었다. H.263 TMN 8의 유클레어 방법을 이용하여 각각 92Kbps, 128Kbps, 192Kbps의 bitrate에 따라 실험하였고, 실수형 DCT/IDCT가 아닌 정수형 DCT/IDCT로 개선된 H.263 TMN 3.0 부호화기와 복호화기를 사용하였다<sup>[7]</sup>. 176×144 크기의 QCIF, Foreman 300 프레임 영상을 Skip 2로 하여 초당 10장의 영상을 부호화하였다. 또한 본 실험을 위해 움직임 벡터 정보인 MVD와 양자화된 DCT 계수인

TCOEFF의 VLC 코드에 Binary Symmetric Error(BSC)를 삽입하였다<sup>[4]</sup>.

1. LSB패턴 정보를 삽입하였을 때의 PSNR

표 3은 채널 에러 검출 및 손실된 매크로블록 복원을 위한 움직임 벡터 유사도 정보를 양자화된 AC계수의 LSB에 삽입하였을 때와 그렇지 않았을 때의 PSNR을 bitrate에 따라 비교한 표이다. 식(1)의 pos\_start를 11로 하였을 때, 약 0.4~0.9dB의 화질 저하가 발생한 것을 확인할 수 있다.

표 3. LSB 패턴을 삽입하였을 때의 PSNR [dB]

| Bitrate (Kbps) | Foreman      |           | News         |           |
|----------------|--------------|-----------|--------------|-----------|
|                | No Embedding | Embedding | No Embedding | Embedding |
| 64             | 31.09        | 30.53     | 35.41        | 34.98     |
| 96             | 33.07        | 32.46     | 38.05        | 37.50     |
| 128            | 34.36        | 33.71     | 39.61        | 38.78     |
| 160            | 35.60        | 34.83     | 40.37        | 39.50     |
| 192            | 36.65        | 35.77     | 40.86        | 39.96     |

2. 에러 검출 성능

표 4는 GOB 단위의 구문분석을 통한 에러 검출 방법의 성능을 비교하기 위한 표로써 Foreman 영상 100프레임 전체에 대하여 에러가 삽입된 GOB 수와 검출된 GOB 수를 나타낸다. 일반적인 복호화기는 구문분석 방법만을 이용하여 채널 에러를 검출하게 되는데, 이 경우 GOB 단위에서 약 50% 안팎의 검출율을 보이고 있음을 확인할 수 있다. 이것은 구문 분석 방법을 통한 에러 검출 성능이 그리 뛰어나지 않고, 많은 에러가 그대로 복호화 된다는 것을 의미한다.

표 4. GOB 단위의 구문 분석 방법의 에러 검출 성능

| Error Rate | 64Kbps |     |             | 128Kbps |     |             | 192Kbps |     |             |
|------------|--------|-----|-------------|---------|-----|-------------|---------|-----|-------------|
|            |        |     | 검출율 (B/A) % |         |     | 검출율 (B/A) % |         |     | 검출율 (B/A) % |
|            | A      | B   |             | A       | B   |             | A       | B   |             |
| 1E-3       | 460    | 233 | 50.6        | 808     | 412 | 51.0        | 880     | 534 | 60.7        |
| 8E-4       | 378    | 193 | 51.1        | 704     | 346 | 49.1        | 852     | 481 | 56.5        |
| 6E-4       | 297    | 143 | 48.1        | 589     | 278 | 47.2        | 770     | 398 | 51.7        |
| 4E-4       | 206    | 100 | 48.5        | 426     | 191 | 44.8        | 620     | 298 | 48.1        |
| 2E-4       | 104    | 50  | 48.1        | 222     | 101 | 45.5        | 335     | 158 | 47.2        |
| 1E-4       | 52     | 25  | 48.1        | 111     | 52  | 46.8        | 174     | 79  | 45.4        |

A : 에러가 삽입된 GOB수

B : 구문 분석을 통한 에러가 검출된 GOB 수

표 5는 제안한 방법을 이용하여 에러가 있는 것으로 검출된 매크로블록의 수를 나타내는 표이다. 이는 그림 9에서 구문 분석에 의해 검출되지 않지만 에러에 의해 손실된 매크로블록을 검출한 회수를 의미한다.

표 5. 제안한 방법을 이용하여 검출된 손실된 매크로블록의 수 (100프레임)

| Error Rate | 64Kbps | 128Kbps | 192Kbps |
|------------|--------|---------|---------|
| 1E-3       | 153    | 377     | 689     |
| 8E-4       | 108    | 309     | 603     |
| 6E-4       | 73     | 220     | 417     |
| 4E-4       | 54     | 145     | 256     |
| 2E-4       | 21     | 64      | 133     |
| 1E-4       | 10     | 28      | 58      |

### 3. 손실된 매크로블록의 복원

표 6은 일반적인 구문 분석 방법을 통해 에러를 검출하고, 일반적인 복원 방법인 손실된 매크로블록의 상단에 위치한 매크로블록의 움직임 벡터를 가져온 경우(Syntax-based), 구문 분석 방법과 더불어 에러 검출 패턴만을 이용한 경우(Proposed A), 움직임 벡터 유사도 정보를 함께 사용한 경우(Proposed B)를 각각 적용하였을 때, 복원된 영상의 PSNR을 나타낸 표이다. 일반적인 구문 분석 방법에 비해 PSNR 수치가 눈에 띄게 향상되지 않은 것은 채널 에러의 발생이 불규칙적이기 때문에 에러가 발생하지 않은 영상에 대해서도 실험 결과에 반영하였기 때문이다. 또한 에러율이 매우 높은 경우는 제안하는 방법을 사용한다고 하더라도 바로 이전에 전송된 영상 또한 에러에 의해 변질되었을 확률이 높기 때문에 에러를 검출할 수 있는 능력이 높아졌음에도 불구하고 이를 복원하기 위한 참조 영상이 부족하여, 효과적으로 에러를 복원할 수 없었다. 하지만 에러 검출 능력이 향상된 만큼 제안된 방법이 일반적인 구문 분석 방법에 비해 화질이 평균적으로 향상되었음을 표 6에서 보여주고 있다. 표 6의 Proposed A와 Proposed B 역시 큰 차이는 없지만, Proposed B가 약간 높은 수치를 보여주고 있다. 이 역시 불규칙하게 발생하는 에러에 대해 Proposed A의 에러 복원 능력보다 Proposed B의 복원 능력이 더 뛰어나다는 것을 알 수 있다.

표 6. 일반적인 복호화기와 제안하는 복호화기의 PSNR 성능 (10회 실험 평균)

[dB]

| Error Rate | 64Kbps       |          |       | 128Kbps      |          |       | 192Kbps      |          |       |
|------------|--------------|----------|-------|--------------|----------|-------|--------------|----------|-------|
|            | Syntax-based | Proposed |       | Syntax-based | Proposed |       | Syntax-based | Proposed |       |
|            |              | A        | B     |              | A        | B     |              | A        | B     |
| 1E-3       | 21.42        | 21.73    | 21.81 | 22.34        | 23.17    | 23.33 | 20.29        | 21.68    | 21.66 |
| 8E-4       | 21.92        | 22.09    | 22.13 | 22.45        | 22.71    | 22.75 | 21.05        | 22.54    | 22.59 |
| 6E-4       | 21.96        | 22.50    | 22.53 | 23.78        | 24.23    | 24.45 | 22.67        | 25.53    | 25.52 |
| 4E-4       | 21.98        | 22.05    | 22.06 | 24.49        | 25.80    | 26.10 | 22.96        | 24.00    | 24.01 |
| 2E-4       | 22.52        | 22.59    | 22.60 | 28.47        | 29.42    | 29.60 | 25.65        | 25.99    | 26.06 |
| 1E-4       | 22.70        | 22.67    | 22.69 | 30.16        | 30.46    | 30.51 | 28.18        | 28.35    | 28.42 |

Syntax-based : 일반적인 구문 분석을 통한 에러 검출 방법  
 Proposed A : 움직임 벡터 유사도 정보를 사용하지 않고 에러 검출 패턴만을 이용한 경우  
 Proposed B : 움직임 벡터 유사도 정보와 에러 검출 패턴을 모두 이용한 경우

다음의 그림 10과 11은 제안한 방법이 효과적으로 적용된 예를 나타내는 그림이다.



(a) 구문 분석 에러 검출 방법



(b) 제안한 에러 검출 및 은닉 방법

그림 10. 구문 분석 에러 검출 방법과 제안한 에러 검출 및 은닉 방법의 비교



그림 10(a)는 기존의 일반적인 복호화기에서 사용하는 구문 분석 방법을 통한 에러 검출을 한 후, 에러가 발생한 매크로블록의 상단에 위치한 움직임 벡터를 이용하여 손실된 매크로블록을 복원한 그림이다. 검출되지 않은 에러에 의해 매크로블록 전체가 잘못 복호화된 것을 확인할 수 있다. 그림 10(b)는 제안한 에러 검출 패턴 및 움직임 벡터 유사도 정보를 이용하여 복호화한 영상이다. 에러 검출 능력 향상으로 눈에 띄는 화질 향상 효과를 확인할 수 있다.



(a) 제안한 에러 검출 방법



(b) 제안한 에러 검출 및 은닉 방법

그림 11. 움직임 벡터 유사도 정보 이용의 예

그림 11은 제안한 인터 매크로블록에서의 움직임 벡터 유사도 정보를 이용하여 손실된 매크로블록을 복원하였을 경우(b)와 이용하지 않은 경우(a)를 비교한 그림이다. 그림 11(a)는 VLC 데이터에 포함된 특정 패턴을 이용하여 에러를 체크하여 에러가 발생한 블록의 위쪽에 인접한 블록의 움직임 벡터를 가지고 복원하여 그림 10(a)에서 나타나는 것과 같은 블록 형태의 에러를 감춘 것이다. 블록 형태로

두드러지는 에러는 보이지 않지만 영상의 인물이 쓰고 있는 안전모의 테두리 부분이 올바르게 복원되지 않았다. 그림 11(b)는 에러가 발견된 블록을 복원함에 있어 위에 인접한 움직임 벡터를 그대로 사용하는 것이 아니라, VLC 코드에 에러 검출 패턴과 함께 넣었던 이웃 블록의 움직임 벡터 유사도 정보를 이용하여 복원한 것이다. 그림 11(a)의 어긋난 안전모 테두리 부분이 부드럽게 복원되어 움직임 벡터 유사도 정보가 중요하게 적용되었음을 확인할 수 있다.

## V. 결 론

본 논문은 비디오 채널 에러에 의해 손상된 압축 데이터의 효율적인 복원을 위해 워터마크를 삽입하여 전송 에러를 검출하는 방법을 제안하고 있다. 채널 잡음에 의해 손상된 압축 데이터가 복호화 되었을 때, 경우에 따라 매우 큰 화질 저하를 발생시킨다. 이를 위해 채널 잡음을 효과적으로 검출하고 손상된 부분을 복원하기 위해 양자화된 DCT 계수에 에러 검출 및 손상된 부분의 복원을 위한 워터마크를 삽입하여 이를 이용하는 방법을 제안하였다. 기존의 구문 분석 방법의 경우 검출율이 50% 정도에 그쳤으나, 이를 이용할 경우, 기존의 구문 분석을 이용한 에러 검출 방법에서 검출할 수 없었던 에러들을 대부분 검출 할 수 있었다. 또한 에러에 의해 손상된 매크로블록을 복원하는데 있어, 움직임 벡터 유사도 정보를 이용하여 복원함으로써 화질의 향상을 가져올 수 있었다. 제안하고 있는 방법은 무선 채널 환경 하에서 동영상 데이터 통신에 유용하게 활용될 수 있다.

## 참 고 문 헌

- [1] Yao Wang, Stephan Wenger, Jiangtao Wen, and Aggelos K. Katsaggelos, "Error Resilient Video Coding Techniques", IEEE Signal Processing Magazine, Vol. 17, No. 4, pp.61-82, July 2000.
- [2] Weiping Li, Jens-Rainer Ohm, Mihaela van der Schaar, Hong Jiang, Shipeng Li, "MPEG-4 Video Verification Model version 17.0", ISO/IEC JTC1/SC29/WG11 N3515, July 2000.

[3] Yao Wang, Qin-Fang Zhu, "Error Control and Concealment for Video Communication: A Review", IEEE Proceedings, Vol. 86, No. 5, pp.974-997, May 1998.

[4] Wee Sun Lee, M. R. Pickering, M. R. Frater, and J. F. Arnold, "Error resilience in video and multiplexing layers for very low bitrate video coding systems", IEEE Journal on Selected Areas in Communications, Vol. 15, No. 9, pp.1764-1774, Dec. 1997.

[5] Minghua Chen, Yun He, Reginald L. Lagendijk, "Error detection by fragile watermarking", Proceedings of Picture Coding Symposium 2001, pp.287-290, April 2001.

[6] J. Song and K. J. R. Liu, "A data embedding scheme for H.263 compatible video coding", Proceedings of 1999 IEEE International Symposium on Circuits and Systems (ISCAS), vol.4, pp.390-393, 1999.

[7] Ying-Jui Chen, Soontorn Oraintara, Truong Nguyen, "Video compression using integer DCT", Proceedings of 2000 International Conference on Image Processing, vol.2, pp.844-847, 2000

전 병 우(Byeungwoo Jeon)

정회원



1985년 2월 : 서울대학교 전자공학과 졸업(공학사)

1987년 2월 : 서울대학교 전자공학과 졸업(공학석사)

1992년 12월 : Purdue Univ. School of Elec. 졸업(공학박사)

1993년 ~ 1997년 8월 : 삼성전자 신호처리 연구소 선임 및 수석 연구원

1997년 9월 ~ 현재 : 성균관대학교 정보통신공학부 부교수

<관심분야> 멀티미디어, 영상압축, 영상인식

박 운 기(Woonki Park)

정회원



2000년 2월 : 성균관대학교 전기전자및컴퓨터공학부 졸업(공학사)

2002년 2월 : 성균관대학교 전기전자및컴퓨터공학부 졸업(공학석사)

2002년 3월~현재 : 삼테크아이

엔 씨 기술연구소 주임연구원

<관심분야> 영상처리, 영상압축, 멀티미디어통신