Journal of Korea Multimedia Society Vol. 12, No. 11, November 2009(pp. 1563-1571)

H.264/AVC의 시간적 오류 은닉을 위한 경계 정합과 후처리 방법

이준우[†], 나상일[††], 원인수[†††], 임대규[††††], 정동석[†††††]

요 약

H.264/AVC의 시간적 오류 은닉을 위한 새로운 경계 정합 방법과 은닉된 영상의 화질 향상을 위한 후처리 방법을 제안한다. 시간적 오류 은닉은 참조 프레임에서 가장 유사한 블록으로 오류가 발생한 블록을 대체시키는 방법이다. 가장 유사한 블록을 찾기 위해 H.264/AVC에서는 오류가 발생한 블록의 바깥 경계의 화소값과 참조 블록의 안쪽 경계의 화소값을 단순 비교한다. 그러나 기존의 방법은 좁은 범위의 화소값 만을 비교하므로 부정확한 블록으로 오류 블록을 대체할 확률이 높다. 본 논문에서는 더욱 정확한 블록으로 오류 블록을 대체하기 위하여 참조 블록의 안쪽 경계의 화소값과 바깥 경계의 화소값을 오류 블록의 바깥 경계의 화소값과 비교하고 후보 움직임 벡터의 최솟값과 최댓값을 기준으로 일정 검색 범위내의 추가적인 후보 움직임 벡터를 설정하는 보다 향상된 경계 정합 방법, 그리고 변형된 디블록킹 필터를 통해 오류 없이 복호된 블록과 오류가 은닉된 블록과의 경계를 부드럽게 하는 후처리 방법을 제안한다. 실험을 통하여 제안된 방법이 기존의 블록 정합 방법보다 최대 약 0.9 dB의 화질 향상을 보여주는 것을 확인하였다.

A Boundary Matching and Post-processing Method for the Temporal Error Concealment in H.264/AVC

Jun Woo Lee[†], Sang Il Na[††], In Su Won[†††], Dae Kyu Lim[††††], Dong Seok Jeong[†††††]

ABSTRACT

In this paper, we propose a new boundary matching method for the temporal error concealment and a post processing algorithm for perceptual quality improvement of the concealed frame. Temporal error concealment is a method that substitutes error blocks with similar blocks from the reference frame. In conventional H.264/AVC standard, it compares outside pixels of erroneous block with inside pixels of reference block to find the most similar block. However, it is very possible that the conventional method substitutes erroneous block with the wrong one because it compares only narrow spatial range of pixels. In this paper, for substituting erroneous blocks with more correct blocks, we propose enhanced boundary matching method by comparing inside and outside pixels of reference block with outside pixels of erroneous block and setting up additional candidate motion vector in the fixed search range based on maximum and minimum value of candidate motion vectors. Furthermore, we propose a post processing method to smooth edges between concealed and decoded blocks without error by using the modified deblocking filter. We identified that the proposed method shows quality improvement of about 0.9dB over the conventional boundary matching methods.

Key words: H.264/AVC(디지털 비디오 코덱), Temporal Error Concealment(시간적 오류 은닉)

※ 교신저자(Corresponding Author) : 정동석, 주소 : 인천시 남구 용현동 253(402-751), 전화 : 032)860-7415, FAX : 032)868-3654, E-mail : dsjeong@inha.ac.kr
접수일 : 2009년 5월 26일, 수정일 : 2009년 8월 21일
완료일 : 2009년 9월 17일
[†] 정회원, 인하대학교 전자공학과 박사과정
(E-mail : junwoo@inha.edu)
[††] 인하대학교 전자공학과 박사과정
(E-mail : leptons@inhaian.net)
[†††] 정회원, 인하대학교 전자공학과 박사과정
(E-mail : woninsu@inha.edu)
[††††] 인하대학교 전자공학과 박사과정
(E-mail : neptune2009@inha.edu)
[†††††] 인하대학교 전자공학과 교수
※ 이 논문은 인하대학교 교내학술연구비의 지원을 받아 수행된 연구임.

1. 서 론

최근 IT 기술의 괄목할 만한 성장으로 인하여 다양한 멀티미디어 서비스에 대한 수요가 나날이 증가하고 있다. 특히 고화질, 고품질의 영상과 같은 고용량의 멀티미디어 데이터에 대한 수요 증가로 인하여 데이터 압축에 대한 중요성은 날로 증가하고 있다. 제한된 압축 성능을 보여주는 기존의 영상 부호화 기술은 점점 증대되는 멀티미디어 데이터의 압축에 한계를 드러내었고 그 압축의 한계를 뛰어넘기 위하여 국제 표준화 기구들의 압축 기술에 대한 연구가 활발히 진행되었다. 그 결실로 H.264/AVC 표준이 완성되었다[1,2].

하지만 좋은 압축 기술을 가지고 영상 데이터를 압축해서 전송한다고 해도 유, 무선 네트워크 환경에서 다양한 원인에 의해 발생되는 오류로 인하여 패킷의 손상을 입는다면 영상 압축과 같이 이전 프레임을 참조로 하는 경우 오류 전파로 인하여 심각한 화질 열화가 발생하게 된다. 그렇기 때문에 H.264/AVC는 전송상에서 발생할 수 있는 오류를 줄이기 위한 다양한 방법을 제공한다. 오류 상황을 탄력적으로 대처하기 위해 부호화 단에서 오류에 대한 내성을 높이는 다양한 방법을 사용하고 이미 오류가 발생한 비트스트림을 받았을 경우에는 복호화 단에서 오류를 은닉하는 다양한 방법이 적용된다[3,4].

본 논문에서는 H.264/AVC의 오류를 은닉하는 방법에 초점을 맞추었다. 오류를 은닉하는 방법에는 공간, 시간적인 중복성을 이용한 방법이 있다. 공간적 중복성을 이용하여 오류를 은닉하는 방법으로는 오류가 발생한 매크로 블록 주변의 오류 없이 복호된 매크로 블록을 이용하여 경계선의 방향을 고려한 보간법을 이용해 오류가 발생한 매크로 블록을 은닉하는 방법이 있다[5]. 공간적 중복성을 이용하는 방법은 일반적으로 화면 내 부호화 프레임이나 움직임 벡터가 큰 프레임에서 주로 사용된다. 시간적 중복성을 이용하여 오류를 은닉하는 방법은 오류가 발생한 매크로 블록과 이전 프레임으로부터 동일한 위치의 매크로 블록을 그대로 복사해오는 프레임 복사 방법이 있고 이전 프레임으로 동일한 위치의 매크로 블록의 움직임 벡터를 그대로 현재 매크로 블록에 적용시키는 움직임 복사 방법이 있다[6]. 또한 오류가 발생한 매크로 블록 주변 블록의 움직임 벡터가 가리키는 참조 프레임의 블록들 중에서 현재 프레임의 경계값의 차이가 가장 작은 블록으로 오류가 발생한 블록을 대체하는 Boundary Matching Algorithm(BMA) 방법[6] 이 있고 선명한 경계선이 많은 영상에서 BMA 보다 좋은 성능을 나타내는 Overapping Boundary Matching Algorithm(OBMA) 방법이 있다[7].

본 논문에서는 기존의 BMA 방법과 OBMA 방법을 혼합하고 기존의 방법보다 많은 후보를 설정하여 오류를 은닉한 후 에러가 발생한 경계선을 필터 처리를 통해 부드럽게 하여 보다 좋은 오류 은닉 성능을 나타내는 방법을 제안한다.

2. H.264 영상 파일 전송

H.264/AVC 표준은 다양한 유/무선 네트워크 환경에서 유연하게 적용되도록 H.264 영상은 영상 부호화 처리 그 자체를 다루는 영상 부호화 계층(Video Coding Layer, VCL)과 부호화된 정보를 전송하고 저장하는 하위 시스템과의 사이에 있는 네트워크 추상 계층(Network Abstraction Layer, NAL)의 분리된 구조로 구성한다. 이와 같이 분리하여 정의하는 목적은 영상 부호화 계층의 압축 특징과 네트워크 추상 계층의 전송 특징 사이를 구별하기 위해서이다[8,9].

영상 부호화 계층이란 부호화 단계를 거쳐서 나온 압축 스트림 즉, 부호화된 영상 데이터를 나타내는 연속적인 비트들을 의미하고 이것을 전송하거나 저장하기 전에 다양한 네트워크에 적합하도록 네트워크 추상 계층 단위(NAL unit, NALU)로 변환된다. 각 네트워크 추상 계층 단위는 압축된 영상 데이터 또는 헤더 정보에 해당하는 데이터인 Raw Byte Sequence Payload(RBSP)를 포함한다. 압축된 영상 파일은 그림 1과 같은 구조로 표현된다. H.264/AVC

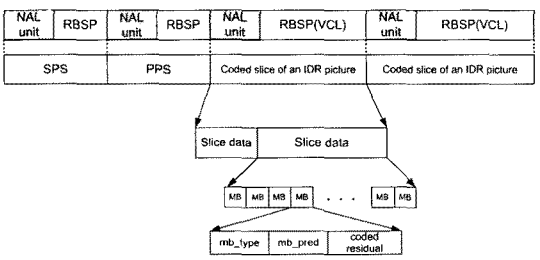


그림 1. H.264/AVC 영상 파일의 구조

는 파라미터 셋(parameter set)이라는 개념을 도입하고 있는데, 각 파라미터 셋은 많은 부호화된 픽처에 적용될 수 있는 정보를 포함하고 있다. 그림 1의 영상 파라미터 셋(sequence parameter set, SPS)은 전체 영상에 적용되는 프레임 번호, 복호화에 사용되는 참조 프레임의 수, 복호되는 픽처의 넓이와 높이, 그리고 순차주사 또는 비월주사 부호화의 선택과 같은 정보들을 포함한다. 또한 그림 1의 픽처 파라미터 셋(picture parameter set, PPS)은 선택된 영상 파라미터 넷의 식별자, 가변 길이 부호화 또는 문맥 기반 적응적 이진 산술 부호화를 선택하기 위한 플래그, 초기 양자화 파라미터 등을 포함한다. 일반적으로 하나 또는 그 이상의 영상 파라미터 셋과 픽처 파라미터 셋이 슬라이스 헤더와 슬라이스 데이터가 복호되기 전에 복호기로 전송된다. 이와 같이 파라미터 셋은 없어서는 안 될 중요한 정보를 포함하고 있다. NAL 유닛을 전송하는 방법은 표준안에 정의되어 있지 않지만, 주로 패킷 기반 네트워크(packet-based network)를 통해 전송된다. 패킷 기반 네트워크에서 각 NAL 유닛은 서로 독립적인 패킷에 실릴 수 있으며 복호화 단계 이전에 올바른 순서로 구성되어야 한다[3]. 하지만 실시간 프로토콜(Real Time Protocol, RTP)과 사용자 데이터그램 프로토콜(User Datagram Protocol, UDP)과 같은 다양한 전송 방식을 통하여 압축 스트림을 전송할 때 예기치 못한 오류로 인하여 하나 또는 그 이상의 패킷이 손실된다면 수신단에서 복호한 영상의 화질 열화를 피할 수 없게 된다.

3. 기존의 오류 은닉 기술

다양한 네트워크 환경에서 영상 데이터를 전송할 경우 여러 가지 원인에 의하여 패킷의 손실이 발생할 수 있다. H.264/AVC에서는 이와 같이 손실된 패킷으로 인한 화질 저하를 막기 위하여 다양한 방법을 제공한다. 그 중 시간적 오류 은닉 기술(Temporal Error Concealment, TEC)은 연속되는 프레임간의 유사성을 이용하여 오류가 발생한 매크로 블록을 은닉하는 방법이다.

3.1 영 움직임 벡터(Zero Motion Vector, ZMV)

연속되는 프레임 간에는 높은 시간적 중복성이 있다. 영 움직임 벡터 방법은 이러한 시간적 중복성을 이용하여 오류가 발생한 매크로 블록의 움직임 벡터를 전부 영으로 설정한다. 즉, 오류가 발생한 매크로 블록을 이전 프레임의 동일한 위치의 매크로 블록으로 채워 넣는다[6]. 움직임이 거의 없는 영상의 경우에 현재 프레임과 이전 프레임과의 차이는 미미하므로 영 움직임 벡터 방법을 사용하게 되면 좋은 성능을 보이게 된다. 하지만 움직임이 많은 영상에서는 이전 프레임과 현재 프레임간의 차이가 커지기 때문에 이 방법을 사용하면 좋은 결과를 얻기가 힘들다.

3.2 경계 정합 방법(Boundary Matching Algorithm, BMA)

H.264/AVC에서 기본적으로 사용하고 있는 시간적 오류 은닉 방법으로서 오류가 발생한 주변 블록의 움직임 벡터를 참조하여 오류가 발생한 매크로 블록의 움직임 벡터 값을 설정하는 방법이다. 경계 정합 방법은 오류가 발생한 매크로 블록 주변의 움직임 벡터를 이용하여 참조 프레임으로부터 얻어온 후보 블록들을 오류가 발생한 위치에 대체시킨다. 그리고 가장 인접한 주변 매크로 블록의 가장자리 화소값의 차이를 누적한 값이 최소가 되는 후보 블록으로 오류가 발생한 위치에 최종적으로 대체시키는 방법이다[10].

식 (1)은 경계 정합 방법을 이용하여 오류 누적 값(Sum of Absolute Difference, SAD)을 구하는 식이다.

$$
\begin{aligned}
&SAD_{BMA} = \\
&\sum_{x=x_0}^{x_0+N-1}\left[\left|P_{x,y_0-1}-P^{r}_{x+v_x,y_0+1+v_y}\right| + \left|P_{x,y_0+N}-P^{r}_{x+v_x,y_0+N-1+v_y}\right|\right] \\
&+\sum_{y=y_0}^{y_0+N-1}\left[\left|P_{x_0-1,y}-P^{r}_{x_0+1+v_x,y+v_y}\right| + \left|P_{x_0+N,y}-P^{r}_{x_0+N-1+v_x,y_0+v_y}\right|\right]
\end{aligned}
\tag{1}
$$

여기에서 N은 매크로 블록의 크기 즉, 16이다. (x_0, y_0)는 오류가 발생한 매크로 블록의 좌측 상단 화소의 위치이고, (v_x, v_y)는 후보 움직임 벡터 값이다. 그리고 $P_{x,y}$와 $P^{r}_{x,y}$는 각각 현재의 화소값과 참조 프레임의 화소값이다.

경계 정합 방법은 인접한 서로 다른 위치의 화소값의 차이를 누적 시키므로 오류가 발생한 블록의 경계가 부드러울 경우 좋은 성능을 나타낸다.

3.3 중복 경계 정합 방법(Overlapping Boundary Matching Algorithm, OBMA)

중복 경계 정합 방법은 기존의 경계 정합 방법과

오류가 발생한 매크로 블록 주변의 움직임 벡터를 이용하여 참조 프레임으로부터 얻어온 후보 블록들을 오류가 발생한 위치에 대체시킨다는 것 까지는 동일하지만, 차이 값을 누적하는 방법이 다르다. 중복 경계 정합 방법은 주변 블록의 후보 움직임 벡터를 이용하여 대체된 매크로 블록의 최인접 외각 화소값과 현재 프레임의 오류가 발생한 블록의 외각 화소값들의 차이를 누적한 값이 최소가 되는 후보 블록으로 오류가 발생한 블록을 대체시킨다[7].

식 (2) 는 중복 경계 정합 방법을 이용하여 오류 누적 값을 구하는 식이다.

$$SAD_{OBMA} = \sum_{x=x_0}^{x_0+N-1}\left[\left|P_{x,y_0-1}-P^{r}_{x+v_x,y_0-1+v_y}\right| + \left|P_{x,y_0+N}-P^{r}_{x+v_x,y_0+N+v_y}\right|\right] + \sum_{y=y_0}^{y_0+N-1}\left[\left|P_{x_0-1,y}-P^{r}_{x_0-1+v_x,y+v_y}\right| + \left|P_{x_0+N,y}-P^{r}_{x_0+N+v_x,y_0+v_y}\right|\right] \quad (2)$$

그림 2는 위에서 설명한 기존의 경계 정합 방법과 중복 경계 정합 방법의 누적 차이값을 구하는 방법을 나타내었다. 중복 경계 정합 방법은 기존의 경계 정합 방법과는 다르게 동일한 위치의 화소값의 차이를 누적시키므로 오류가 발생한 프레임의 경계선이 뚜렷할 경우 좋은 성능을 나타낸다.

4. 새로운 경계 정합 방법

본 논문에서는 블록의 경계가 부드러운 영상에서 좋은 성능을 보이는 기존의 경계 정합 방법과 블록의 경계가 뚜렷한 영상에서 보다 좋은 성능을 보이는 중복 경계 정합 방법을 혼합하는 방법을 제안한다. 이와 같은 방법을 사용하면 부드러운 영상이나 블록의 경계가 뚜렷한 영상에 치우치지 않는 결과를 얻을 수 있다. 또한 제안된 방법을 이용하여 후보 움직임 벡터의 최솟값과 최댓값으로 범위를 설정한 후 범위 내의 모든 블록 중 가장 작은 누적 오차 값을 나타내는 블록으로 오류가 발생한 블록을 대체시킨다. 그리고 최종적으로 은닉된 매크로 블록과 오류 없이 복호화 된 매크로 블록간의 부자연스러운 경계를 후처리 과정을 통하여 부드럽게 하는 방법을 제안한다.

4.1 기존의 경계 정합 방법과 중복 경계 정합 방법을 혼합한 방법

그림 3과 같이 기존의 경계 정합 방법으로 후보

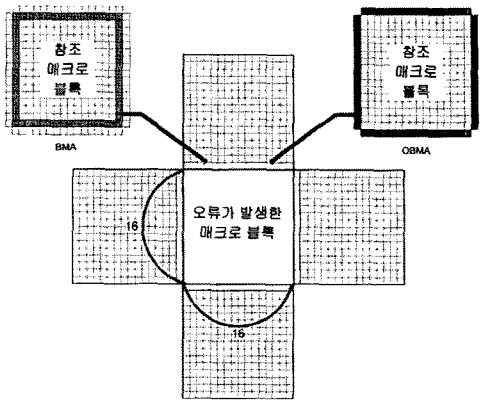


그림 2. 기존의 경계 정합 방법과 중복 경계 정합 방법의 누적 차이값을 구하는 방법

움직임 벡터가 가리키는 참조 프레임의 매크로 블록의 안쪽 화소값과 현재 프레임의 오류가 발생한 매크로 블록의 바깥쪽 화소와 차이 값을 누적시키고 중복 경계 정합 방법으로 참조 프레임으로부터 얻어온 매크로 블록의 바깥쪽 화소값과 현재 프레임의 오류가 발생한 매크로 블록의 바깥쪽 화소와의 차이 값을 추가적으로 누적시킨다. 제안된 방법으로 차이값을 누적 시키는 식은 식 (3)에 나타내었다.

$$SAD_{Proposed} = \sum_{x=x_0}^{x_0+N-1}\left[\left|P_{x,y_0-1}-P^{r}_{x+v_x,y_0+1+v_y}\right| + \left|P_{x,y_0+N}-P^{r}_{x+v_x,y_0+N-1+v_y}\right|\right] + \sum_{y=y_0}^{y_0+N-1}\left[\left|P_{x_0-1,y}-P^{r}_{x_0+v_x,y+v_y}\right| + \left|P_{x_0+N,y}-P^{r}_{x_0+N-1+v_x,y_0+v_y}\right|\right] + \sum_{x=x_0}^{x_0+N-1}\left[\left|P_{x,y_0-1}-P^{r}_{x+v_x,y_0-1+v_y}\right| + \left|P_{x,y_0+N}-P^{r}_{x+v_x,y_0+N+v_y}\right|\right] + \sum_{y=y_0}^{y_0+N-1}\left[\left|P_{x_0-1,y}-P^{r}_{x_0-1+v_x,y+v_y}\right| + \left|P_{x_0+N,y}-P^{r}_{x_0+N+v_x,y_0+v_y}\right|\right] \quad (3)$$

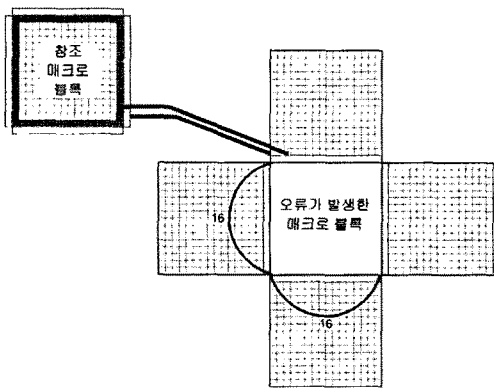


그림 3. 기존의 방법을 혼합한 방법

일반적으로 영상 내의 한 프레임에는 뚜렷한 경계선과 부드러운 경계선이 동시에 존재하게 되는데 이와 같은 방법을 이용하면 뚜렷한 경계선에 약한 기존의 경계 정합 방법과 부드러운 경계선에 약한 중복 경계 정합 방법을 상호 보완할 수 있게 된다.

4.2 범위를 설정하여 후보의 수를 늘리는 방법

H.264/AVC에서는 발생된 오류를 시간적 오류 은닉 기술을 이용하여 은닉할 경우 후보 움직임 벡터 9개를 다음과 같이 정하게 된다. 그림 4와 같이 오류가 발생한 매크로 블록 주변의 위, 아래, 오른쪽, 왼쪽 4개의 매크로 블록의 오류 블록과 가장 인접한 각각 2개씩의 8×8 블록의 움직임 벡터 V는 식 (4)와 같이 정해지게 된다.

$$V = Mean(V(1),\ V(2),\ V(3),\ V(4)) \tag{4}$$

$V(1)$, $V(2)$, $V(3)$, $V(4)$는 그림 4에 나타낸 인접한 8×8 블록 내의 4개의 4×4 블록의 움직임 벡터를 나타낸다. 이렇게 정해진 8개의 8×8 블록의 움직임 벡터가 후보 움직임 벡터가 된다. 그리고 오류가 발생한 블록의 영 움직임 벡터로 나머지 하나의 후보를 정한다. H.264/AVC에서는 이렇게 설정된 9개의 후보 움직임 벡터 중에서 대체할 블록을 찾기 때문에 후보의 개수가 적어서 오류가 발생한 블록을 부정확한 블록으로 대체할 가능성이 있다. 따라서 9개의 후보 움직임 벡터로부터 각각 x, y벡터 값의 최솟값과 최댓값을 구한 후 그 값을 이용하여 범위를 설정한 후 범위 내의 추가적인 후보 움직임 벡터를 설정한다. 먼저 x, y벡터 값의 최솟값과 최댓값은 식 (5)와 같이 구한다.

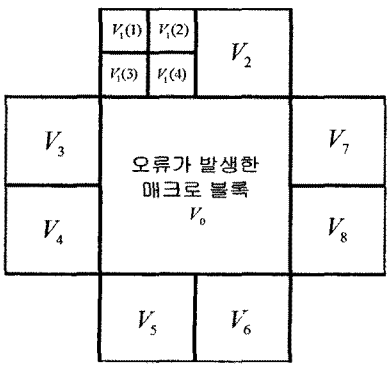


그림 4. 후보 움직임 벡터

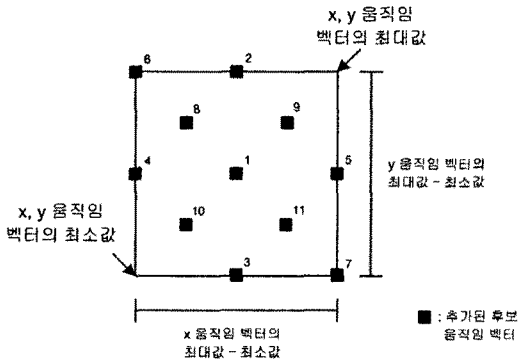


그림 5. 추가적인 움직임 벡터 후보

식 (5)로 부터 얻어진 범위 내에서 그림 5와 같이 추가적인 후보 움직임 벡터를 설정한다. 추가적인 후보 움직임 벡터는 식 (6)에 나타내었다.

$$\begin{aligned}
&Range\ x_{\min} = \min\{V_0(M_x),\ V_1(M_x),\ldots,V_7(M_x),\ V_8(M_x)\}\\
&Range\ y_{\min} = \min\{V_0(M_y),\ V_1(M_y),\ldots,V_7(M_y),\ V_8(M_y)\}\\
&Range\ x_{\max} = \max\{V_0(M_x),\ V_1(M_x),\ldots,V_7(M_x),\ V_8(M_x)\}\\
&Range\ y_{\max} = \max\{V_0(M_y),\ V_1(M_y),\ldots,V_7(M_y),\ V_8(M_y)\}
\end{aligned} \tag{5}$$

$$\begin{aligned}
&(M_{x1}, M_{y1}) = \left(\frac{\max M_x - \min M_x}{2}, \frac{\max M_y - \min M_y}{2}\right)\\
&(M_{x2}, M_{y2}) = \left(\frac{\max M_x - \min M_x}{2}, \max M_y\right)\\
&(M_{x3}, M_{y3}) = \left(\frac{\max M_x - \min M_x}{2}, \min M_y\right)\\
&(M_{x4}, M_{y4}) = \left(\min M_x, \frac{\max M_y - \min M_y}{2}\right)\\
&(M_{x5}, M_{y5}) = \left(\max M_x, \frac{\max M_y - \min M_y}{2}\right)\\
&(M_{x6}, M_{y6}) = \left(\min M_x, \max M_y\right)\\
&(M_{x7}, M_{y7}) = \left(\max M_x, \min M_y\right)\\
&(M_{x8}, M_{y8}) = \left(\frac{M_{x2} - \min M_x}{2}, \frac{\max M_y - M_{y1}}{2}\right)\\
&(M_{x9}, M_{y9}) = \left(\frac{\max M_x - M_{x2}}{2}, M_{y8}\right)\\
&(M_{x10}, M_{y10}) = \left(M_{x8}, \frac{M_{y1} - \min M_y}{2}\right)\\
&(M_{x11}, M_{y11}) = \left(M_{x9}, M_{y10}\right)
\end{aligned} \tag{6}$$

범위내의 가장 가운데 부분 하나와 가운데를 중심으로 한 십자가 모양으로 네 개, x 움직임 벡터의 최댓값과 y 움직임 벡터의 최솟값의 위치와 x 움직임 벡터의 최솟값과 y 움직임 벡터의 최댓값의 위치의 두 개, 그리고 내부의 추가적인 네 개, 총 11개의 추가적인 후보 움직임 벡터를 설정한다. 총 20개의 후보 움직임 벡터의 위치의 블록들을 현재 프레임과

비교하여 누적 차이 값이 최소가 되는 블록으로 현재 프레임의 오류가 발생한 블록을 대체시킨다. 이와 같이 추가적인 후보 움직임 벡터를 설정하게 되면 기존의 방법보다 더욱 많은 후보들 가운데서 대체할 블록을 찾기 때문에 보다 정확한 블록을 선택하게 되고 결과적으로 기존의 방법보다 높은 오류 은닉 성능을 보이게 된다.

4.3 변환된 디블록킹 필터(Deblocking Filter)를 이용한 후처리

제안된 방법을 통하여 후보 매크로 블록으로 오류가 발생한 매크로 블록을 대체한다고 해도 원본과 동일한 블록이 아니기 때문에 아무리 많은 후보 매크로 블록을 가지고 오류가 발생한 매크로 블록을 대체한다고 하여도 오류가 발생하지 않은 주변 블록과 대체된 블록간에는 블록 왜곡 현상이 존재하게 된다. 이러한 블록 왜곡 현상을 줄이기 위하여 오류를 은닉한 영상에 변환된 디블록킹 필터를 적용하는 방법을 제안한다.

기존의 디블록킹 필터를 오류가 은닉된 영상에 적용하게 되면 오류 없이 복호된 매크로 블록의 화소값도 변화하기 때문에 오류가 은닉된 매크로 블록에만 필터링 효과를 적용한다.

그림 6은 오류 없이 복호된 매크로 블록과 오류가 은닉된 매크로 블록 사이의 수직 또는 수평 경계 양쪽의 4개의 샘플 즉, $p0, p1, p2, p3$ 그리고 $q0, q1, q2, q3$

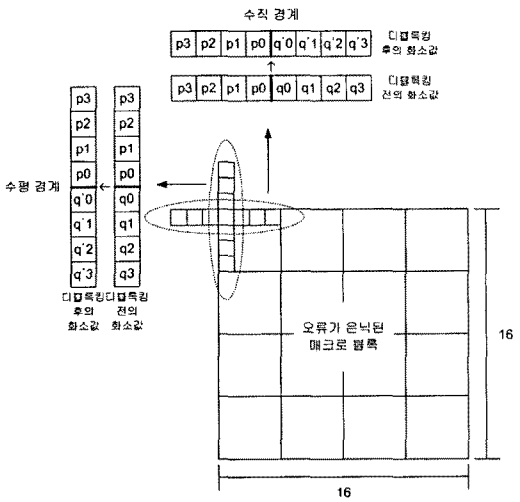


그림 6. 변환된 디블록킹 필터

을 나타내고 있다. 그림 6에서 나타내었듯이 디블록킹 필터 처리를 통하여서 오류가 은닉된 매크로 블록의 위, 아래, 좌, 우 가장자리의 최대 3줄의 화소값을 변화시켜 줌으로써 오류가 은닉된 매크로 블록과 주변 매크로 블록 사이를 부드럽게 해준다. 필터링의 세기는 현재의 양자화 값, 인접한 블록의 부호화 모드 등에 의해 결정된다.

4.3.1 경계 세기(Boundary Strength)

필터링 과정의 선택은 경계 세기와 경계 주위의 영상 샘플의 변화(gradient)에 의해 좌우된다. 경계 세기 파라미터(boundary strength parameter) bS는 표 1의 규칙에 따라 선택된다[3].

이러한 규칙을 적용하게 되면, 화면 내 부호화 매크로 블록의 경계 또는 부호화된 계수들을 포함하는 블록 사이의 경계와 같이 심한 블록 왜곡 현상이 발생할 것 같은 위치의 필터 세기가 강해진다.

4.3.2 필터 결정(Filter Decision)

$(p2, p1, p0, q0, q1, q2)$로부터의 샘플 그룹들은 다음의 경우에만 필터링 된다.

1) $bS > 0$이고

2) $|p0-q0| < \alpha$이고 $|p1-p0| < \beta$ 이고 $|q1-q0| \le \beta$.

α와 β는 표준안에서 정의된 임계값(threshold)으로, 현재 블록의 양자화 파라미터의 평균에 따라 증가한다. 양자화 파라미터가 작을 경우, 경계 주위의 매우 작은 변화 외에는 이미지의 특성에 기인한 것이므로 보존되어야 하므로 임계값 α와 β는 낮다. 양자화 파라미터가 큰 경우 블록 왜곡은 α와 β가 높을 때 더 심해지므로 더 많은 경계 샘플이 필터링 되어야 한다.

표 1. bS 값 정의

p와 화면 내 부호화의 관계	bS 값
p 블록이 화면 내 부호화 매크로 블록에 속하고 매크로 블록 경계에 위치	$bS=3$ (가장 강한 필터링)
p 블록이 화면 내 부호화 매크로 블록에 속하고 매크로 블록 경계에 위치하지 않음	$bS=2$
p 블록이 화면 내 부호화 매크로 블록에 속하지 않고 더 나아가 직교 변환 계수를 가짐	$bS=1$
p 블록이 화면 내 부호화 매크로 블록에 속하지 않고 직교 변환 계수를 가지지 않는다.	$bS=0$ (필터링 안함)

4.3.3. 필터 구현(Filter Implementation)

1) $bS<3$일 경우

4-tap 필터가 입력 $p1, p0, q0, q1$에 대해 적용되어 필터링된 출력 $q'0$을 생성한다. $|q2-q0|$이 임계값 β보다 작으면 또 다른 4-tap 필터가 입력 $q2, q1, q0, p0$에 적용되어 필터링된 출력 $q'1$이 생성된다.

2) $bS=3$ 일 경우

$|q3-q0|<\beta$이고 $|p0-q0|<\partial\,(\alpha/4)$이고 휘도 블록일 경우에는 $q'0$는 $q2, q1, q0, p0$ 그리고 $p1$에 5-tap 필터링을 적용하여 생성된다. $q'1$는 $q2, q1, q0$ 그리고 $p0$에 4-tap 필터링을 적용하여 생성되고 $q'2$는 $q3, q2, q1, q0$ 그리고 $p0$에 5-tap 필터링을 적용하여 생성된다. 그렇지 않으면 $q'0$는 $q1, q0$ 그리고 $p1$에 3-tap 필터링을 적용하여 생성된다[3].

5. 실험 결과

제안된 방법을 평가하기 위해 JVT Model(JM) 참조 소프트웨어 10.1 버전을 사용하였고 실험은 QCIF (176×144) 크기의 Foreman, Bus, Football, Ice 영상을 사용하였다[11]. 영상은 30프레임을 사용하였고 양자화 파라미터는 28로 설정하였다. 효율적인 오류 은닉을 위해 유연한 매크로 블록 순서 방법(Flexible Macroblock Order, FMO)을 사용하여 한 프레임을 두 개의 슬라이스로 나누고 슬라이스에 따른 매크로 블록을 격자 모양으로 부호화 하였다[12]. 그리고 IPPP 의 프레임 순서로 부호화 하였다. 또한 화질 측정 방법으로 PSNR(Peak to Peak Signal to Noise Ratio)을 사용하였다.

그림 7은 결과 영상들의 PSNR 그래프이다. 부드러운 영상과 경계가 뚜렷한 영상 중 어느 한쪽에만 적합한 경계 정합 방법이 아닌 새로운 경계 정합방법을 이용하여 기존의 방법 보다 더 많은 후보들 가운데에서 은닉할 블록을 찾고 또한 후처리를 통하여 은닉한 블록과 기존 블록간의 경계를 부드럽게 해줌으로 인하여 기존의 방법들에 비해 전체적으로 높은 PSNR 값을 갖게 된다. PSNR 값 뿐만 아니라 그림 8과 같이 기존의 오류 은닉 방법들과 제안된 오류 은닉 방법을 적용한 결과 영상을 비교하여 보면 매크로 블록간의 경계가 더 자연스럽다. 영 움직임 벡터 방법, 기존의 경계 정합 방법, 그리고 중복 경계 정합 방법과 비교하였을 때 PSNR 값 뿐만 아니라 주관적

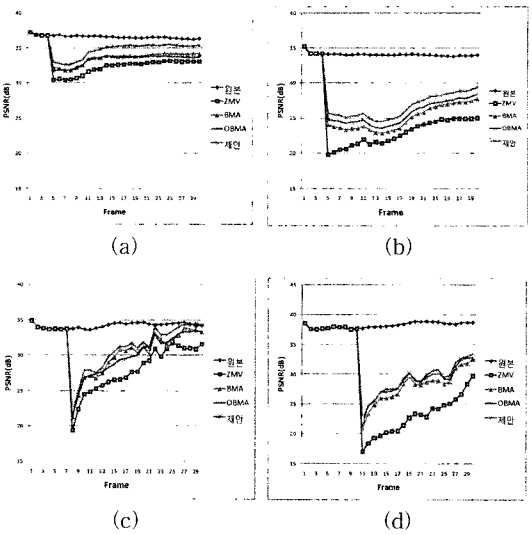


그림 7. 결과 영상들의 PSNR 그래프, (a) Foreman 영상, (b) Bus 영상, (c) Football 영상, (d) Ice 영상.

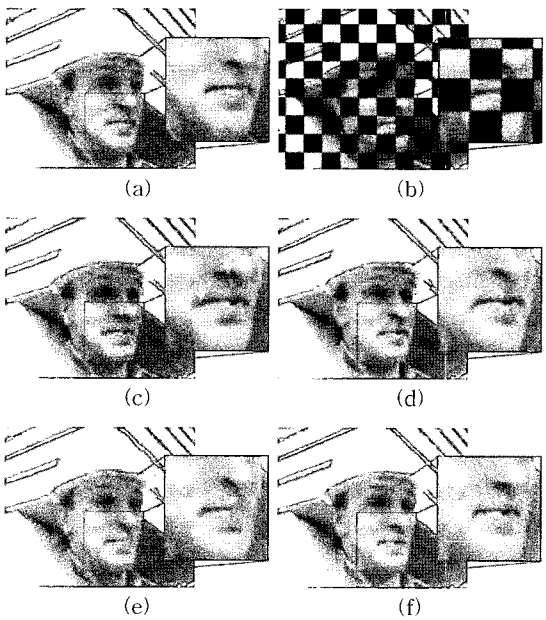


그림 8. Foreman 결과 영상들, (a) 원본 영상, (b) 오류 발생 영상, (c) 영 움직임 벡터 방법, (d) 기존의 경계 정합 방법, (e) 중복 경계 정합 방법, (f) 제안된 방법.

인 평가로도 제안된 방법으로 오류를 은닉했을 경우 더 좋은 성능을 보임을 확인할 수 있다. 그리고 Core 2 Duo 3.0 GHz, 4GB Ram의 PC를 사용하여 기존의 경계 정합 방법과 중복 경계 정합 방법을 사용하여 30프레임을 복호화 하는데 걸리는 시간은 평균 4.53

표 2. 실험 영상에 대한 PSNR(dB) 비교

영상	PSNR(dB)				
	원본	ZMV	BMA	OBMA	제안된 방법
Foreman	36.7526	30.4619	32.2403	31.6725	32.9574
Bus	34.148	19.8151	24.0754	24.8862	25.785
Football	33.7371	19.3658	21.2742	21.1902	21.9719
Ice	37.7675	17.0483	20.9694	21.8445	22.4227

초, 4.69초인 반면 제안된 방법을 사용하였을 때에는 평균 5.39초의 시간이 걸렸다. 약간의 계산량 증가를 통하여 정량적, 주관적 관점의 화질이 향상됨을 알 수 있다.

표 2는 위에 나타낸 각 실험 영상의 오류를 은닉한 프레임의 PSNR 값을 나타내고 있다. 기존의 방법들보다 제안된 방법이 0.6dB에서 0.9dB 더 높은 PSNR 값을 나타냄을 알 수 있다. 그러나 Foreman 영상과 같이 비교적 움직임이 적은 영상에서는 오류 은닉 방법을 사용했을 때 원본에 가까운 PSNR 값을 나타내지만 Football이나 Ice와 같은 움직임이 많은 영상에서는 제안된 오류 은닉 방법을 사용했음에도 원본과 PSNR 값의 차이가 심하게 나타나는 것을 볼 수 있다.

6. 결론 및 향후 연구 과제

본 논문에서는 유무선 네트워크 환경에서 발생될 수 있는 패킷의 손실로 인한 동영상의 화질 열화를 막기 위해 사용되는 오류를 은닉하는 방법 중 시간적 중복성을 이용한 방법에 초점을 맞추었다. 기존의 시간적 중복성을 이용한 오류 은닉 방법인 경계 정합 방법과 중복 경계 정합 방법을 혼합하여 새롭게 경계를 정합하는 방법을 제안하였고 오류가 발생한 블록의 주변 움직임 벡터간의 높은 연관성을 이용하여 기존의 방법보다 많은 후보 움직임 벡터를 설정하여서 원본에 가깝게 은닉될 확률을 높이는 방법을 제안하였다. 또한 은닉된 블록과 기존의 블록들 간의 블록 왜곡 현상을 줄이고 자연스러운 영상으로 은닉하기 위해 후처리 단계로 변형된 디블록킹 필터링 과정을 수행하는 방법을 제안하였다.

기존의 오류 은닉 방법들로 오류를 은닉했을 때와 제안된 오류 은닉 방법으로 오류를 은닉했을 때의 PSNR을 비교해보면 제안된 오류 은닉 방법으로 오류를 은닉했을 때가 기존의 방법으로 오류를 은닉했을 때 보다 약 0.6 dB에서 0.9 dB의 화질 향상을 보여주었다.

본 논문에서는 매크로 블록의 크기를 16×16의 크기로 고정하여서 오류를 은닉하였지만 움직임이 많은 영상에서 최적의 오류 은닉이 불가능하므로 향후 매크로 블록의 단위를 8×8, 더 나아가 4×4 단위까지 세밀하게 나누어서 본 논문의 방법을 적용하는 연구와 오류가 발생한 영상내의 다양한 부호화 모드와 움직임 벡터의 많고 적음에 따라 선택적으로 공간적 오류 은닉 방법을 적용하는 연구가 이루어져야 하겠다.

참 고 문 헌

[1] A. Luthra, G. Sullivan, and T. Wiegand, "Introduction to the special issue on the H.264/AVC video coding standard," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol.13, No.7, pp. 557-559, July 2003.

[2] G. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC advanced video coding standard: Overview and Inton to the Fidelity Range Extensions," *SPIE Conf. on Applications of Digital Image Processing XXVII*, Vol.5558, pp. 53-74, Aug. 2004.

[3] 정제창 역, H.264/AVC 비디오 압축 표준, 홍릉과학출판사, 2006.

[4] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan, "Error Resilience Schemes of H.264/AVC Standard," *Journal of Visual Communication and Image Representation*, Vol.17, pp. 425-450, 2006.

[5] D. Agrafiotis, D. R. Bull and N. Canagarajah, "Spatial error concealment with edge related perceptual considerations," *Signal Processing Image Communication*, Vol.21, pp. 130-142, Feb, 2006.

[6] W. M. Lam, A. R. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," *IEEE International Conference*

on Acoustics, Speech, and Signal Processing, ICASSP-93., Vol.5, 1993.

[7] D. Kim, S. Yang, and J. Jeong, "A new temporal error concealment method for H.264 using adaptive block sizes," *IEEE International Conference on Image Processing*, 2005. ICIP 2005., Vol.3, pp. III- 928-31, 2005.

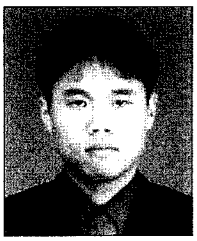
[8] 고성제, 김종옥 역, MPEG-4의 세계, 브레인코리아, 2005.

[9] X. Yang, Z. Ce, L. Zheng Guo, L. Xiao, and L. Nam, "An unequal packet loss resilience scheme for video over the internet," *IEEE Trans. on Multimedia*, Vol.7, pp. 753-765, 2005.

[10] D. Agrafiotis, D. R. Bull and N. Canagarajah, "Enhanced Error Concealment With Mode Selection," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol.16, pp. 960-973, 2006.

[11] JM model 10.1; http://iphome.hhi.de/suehring/tml/JM.10.1.Zip

[12] P. Lambert, W. D. Neve, Y. Dhondt, R. V. D. Walle, "Flexible macroblock ordering in H.264/AVC," Journal of Visual Communication and Image Representation, Vol.17, pp. 358-375, 2006.



나 상 일

1998년 3월~2002년 2월 인하대학교 전자공학과 공학사
2002년 3월~2004년 2월 인하대학교 전자공학과 공학석사
2004년 3월~현재 인하대학교 전자공학과 박사과정
관심분야 : 비디오 카피 디텍션

원 인 수

1999년 3월~2006년 2월 인하대학교 전자공학과 공학사
2007년 3월~2009년 2월 인하대학교 전자공학과 공학석사
2009년 3월~현재 인하대학교 전자공학과 박사과정
관심분야 : 비디오 코덱, 비디오 카피 디텍션

임 대 규

1998년 3월~2002년 8월 인하대학교 전자공학과 공학사
2002년 9월~2004년 8월 인하대학교 전자공학과 공학석사
2009년 3월~현재 인하대학교 전자공학과 박사과정
관심분야 : 비디오 코덱, 컴퓨터 비전

이 준 우

2001년 3월~2006년 2월 인하대학교 전자공학과 공학사
2006년 3월~2008년 2월 인하대학교 전자공학과 공학석사
2008년 3월~현재 인하대학교 전자공학과 박사과정
관심분야 : 비디오 코덱, 비디오 카피 디텍션

정 동 석

1977년 서울대학교 학사
1985년 Virginia Tech 공학석사
1988년 Virginia Tech 공학박사
1988년~현재 인하대학교 전자공학부교수
2000년~2004년 정보전자공동연구소 소장
관심분야 : 컴퓨터 비전, 영상 처리, 포렌식 워터마킹