

경계 방향성을 이용한 손상된 움직임 벡터의 복원 기법

정회원 박 정 윤*, 이 창 우**

Recovering Corrupted Motion Vectors Using Edge Direction

Jung-Yoon Park*, Chang-Woo Lee** *Regular Members*

요 약

통신 기술의 발전으로 무선 채널을 이용한 멀티미디어 서비스를 제공하기 위한 방법이 활발히 연구되고 있다. 무선 채널을 이용한 영상 통신에서 채널상의 오류는 복원되는 영상의 화질에 큰 영향을 미친다. 이러한 채널상의 오류로 인한 화질 열화를 막기 위한 다양한 기법들이 활발히 연구되고 있는데, 그 중 오류 은폐 기법은 부가 정보가 필요 없고 부호화 기법의 변형 없이 적용이 가능하여 최근에 널리 연구되고 있다.

본 논문에서는 동영상 압축 표준안인 H.263에 의해서 압축된 영상이 채널 오류에 의해서 손상되었을 때 주변 블록들의 경계 방향성을 고려해서 손상된 움직임 벡터를 추정하는 새로운 기법을 제안하였다. 이는 각 객체에 속하는 블록들은 유사한 움직임을 갖는다는 특성을 이용한 것으로 방향성을 고려하는데는 각 블록의 첫 번째 AC 계수를 이용하였다. 제안한 방법의 성능을 평가하기 위한 모의 실험 결과 제안된 방법은 기존의 오류 은폐 방법보다 계산량을 크게 줄이면서 PSNR 측면에서 향상된 결과를 보여준다.

ABSTRACT

In wireless communication systems, transmission errors degrade the reconstructed image quality severely. Among various techniques which reduce the degradation, the error concealment technique yields good performance without overheads and the modification of the encoder.

In this paper, we have proposed the technique for recovering the corrupted motion vector using the edge direction of blocks which surround the corrupted block, when the video information is compressed by H.263 standard. We uses the property that each object in video sequences shows the same motion. For the estimation of edge directions, the first AC coefficients of blocks are used. Extensive simulation results show that the proposed technique yields good performance with a reduced computational complexity.

I. 서 론

최근에 개인 휴대 통신 기술이 발전하면서 무선 채널을 이용하여 멀티미디어 서비스를 제공하기 위한 방법이 활발히 연구되고 있다^{1,2}. 특히 멀티미디어 정보 중에서도 동영상 정보는 정보량이 매우 크기 때문에 동영상 정보를 효율적으로 압축하기 위해서 다양한 압축 알고리즘이 제안되었다. 현재 많이 사용되는 동영상 압축 기법에서는 연속되는 프레임간의 시간적 상관성을 이용한 움직임 추정/보상

(motion estimation/motion compensation) 기법, 공간적 상관성을 이용한 DCT(discrete cosine transform) 기법과 허프만 부호화와 같은 가변 길이 부호화(variable length coding : VLC) 등의 알고리즘을 복합적으로 사용한다. 그러나 이런 압축 방법들을 사용하면 전송시 채널 오류에 민감하게 된다. 특히 가변 길이 부호화 기법을 사용한 경우 채널 오류가 발생하면 복호기의 동기가 손실되기 때문에 다시 복호기의 재동기가 이루어지기 전까지의 모든 정보에 오류가 전파될 수 있다. 또한 프레임간 예측

* (주)크로스다임

** 가톨릭대학교 컴퓨터·전자공학부(lcw@www.cuk.ac.kr)

논문번호: K01055-0206, 접수일자: 2001년 2월 6일

부호화 기법인 움직임 보상 기법이 사용된 경우 한 프레임에 오류가 발생하면 이후 연속된 프레임에 오류가 전파되는 특성이 있다. 따라서 채널 오류에 의해 복원되는 영상의 화질이 열화되는 것을 막는 것이 중요하다. 이를 위해서 오류에 강인한 부호화 기를 설계하는 방법, 오류 정정 부호를 사용하는 방법과 별도의 부가 정보 없이 수신 단에서 손상된 데이터를 상관성을 이용해서 오류를 은폐시키는 방법들이 많이 사용되고 있다. 이 중에서 수신단에서 손상된 데이터를 상관성을 이용하여 복원하는 오류 은폐(error concealment) 방법은 별도의 부가 정보가 필요 없고 영상 부호화 기법의 변경없이 적용이 가능한 장점이 있기 때문에 최근에 많이 연구되고 있다^[3-10].

동영상 데이터에서는 시·공간적인 상관성을 이용할 수 있으므로 특히 오류 은폐 기법은 효율적이다. 오류 은폐 기법은 사용하게 되는 영상 정보의 상관성에 따라 시간 예측 오류 은폐(temporal predictive error concealment) 기법과 공간 예측 오류 은폐(spatial predictive error concealment) 기법으로 나눌 수 있다. 시간 예측 오류 은폐 방법은 시간적으로 인접한 프레임의 영상 정보 중에서 적절한 부분을 찾아서 손상된 부분의 정보를 복원하는 것이고, 공간 예측 오류 은폐 방법은 영상의 국부적인 밝기 변화가 연속임을 가정하여 비슷한 밝기 값을 가진 주변 영역의 밝기 값으로부터 손상된 영역의 밝기 값을 추정하는 것이다^[7]. 시간 예측 오류 은폐 기법에서는 이전 프레임에서 현재의 손상된 부분과 가장 유사한 부분을 찾아야 하기 때문에 움직임 벡터가 중요한 역할을 한다. 즉 움직임 벡터는 이전 프레임에서 가장 유사한 부분을 나타내는 정보이므로 손상된 부분의 움직임 벡터를 알 수 있다면 손상된 부분을 잘 복원할 수 있다. 이런 움직임 벡터를 추정하기 위하여 제안된 기존의 방법으로는 주변 블록의 정상적으로 전송된 움직임 벡터 값들의 평균의 취하는 방법과 주변 블록의 움직임 벡터 값들의 중간값을 취하는 방법이 있다. 또한 주변 블록의 움직임 벡터의 가중 평균을 사용하여 화소마다 움직임을 추정하는 기법, 손상된 블록의 경계에서 주변 블록과 가장 잘 융합되도록 하여 움직임 벡터를 추정하는 기법인 BMA(block matching algorithm) 기법도 있다^[5,15]. 공간 예측 오류 은폐 기법은 주변 블록의 밝기 정보를 이용하여 주변 블록과 가장 잘 연결이 되도록 손상된 부분의 DCT 계수나 밝기 값을 복원하는 것으로 일종의 보간

(interpolation) 기법이라 할 수 있다.

본 논문에서는 동영상 표준인 H.263에 의해서 압축된 동영상에서 채널 오류로 움직임 벡터가 손상되었을 때 주변 블록의 특성을 고려해서 움직임 벡터를 추정하는 새로운 기법을 제안하였다. 프레임 내에서 움직임은 각 객체 단위별로 비슷하므로, 손상된 매크로 블록의 움직임이 있다면 주변 매크로 블록들 중에서 같은 객체에 속하는 블록을 찾아 이 블록의 움직임 벡터를 손상된 움직임 벡터로 대체하였다. 같은 객체에 속하는 블록을 찾기 위해서 각 블록의 경계(edge) 방향성을 이용하였다. 제안한 방법의 성능 평가를 위해서 움직임 벡터만 손상되었을 경우와 움직임 벡터와 DCT 계수가 모두 손상되었을 경우에 대해서 모의 실험을 실시하였다. 실험 결과는 기존의 제안된 방법들 중에서 가장 나은 성능을 보이는 BMA와 비슷한 성능을 보였다. 그러나 제안한 방법은 BMA의 많은 계산량을 줄이는 장점이 있다.

본 논문의 구성은 다음과 같다. 제 2장에서는 기존의 오류 은폐 기법에 대해 논하고, 제 3장에서는 새로운 오류 은폐 기법을 제안한다. 제 4장에서는 기존의 오류 은폐 기법과 제안된 오류 은폐 기법을 평가하는 모의 실험과 그 결과를 제시하였고, 제 5장에서는 결론을 제시한다.

II. 오류 은폐 기법

1. 동영상 정보의 오류 은폐

H.263에서는 프레임간의 예측을 통한 시간적 상관성과 DCT를 통한 공간적 상관성을 이용하여 영상을 압축한다. 따라서 채널 오류로 인해 현재 프레임에 열화가 생기면, 그 영향이 이후 프레임에 계속 전파되어 복원 영상의 화질이 크게 저하됨에 따라서 오류 은폐 방법이 요구된다.

오류 은폐 방법에는 영상 정보의 종류에 따라 공간 예측 오류 은폐 방법과 시간 예측 오류 은폐 방법이 있다^[8-10]. 공간 영역에서의 오류 은폐 방법은 같은 프레임내의 블록들 중에서 오류가 일어난 블록의 주변 화소값을 가지고 손실된 블록의 화소값을 추정하는 것이다. 시간 영역에서의 오류 은폐 방법은 이전 프레임의 정보를 이용하여 손상된 부분을 복원하는 것으로서 영상 내에서 움직임이 없는 부분이나 배경같은 부분의 손상에 대해서는 거의 완벽하게 원래의 정보를 복원할 수 있는 장점이 있다. 그러나 움직임이 있는 부분에 대해서는 이전 프

레이에서 현재의 손상된 부분과 가장 유사한 부분을 찾아야 하는데 이에 움직임 벡터가 중요한 역할을 한다. 즉 움직임 벡터는 해당 영역에 대하여 전 프레임에서 가장 유사한 부분을 나타내는 정보이기 때문에 만약 손상된 영역의 움직임 벡터를 알 수 있다면 손상된 영역을 잘 복구할 수 있게 된다.

2. 주변 블록의 움직임 벡터를 이용한 움직임 벡터 추정 방법

가장 간단한 움직임 벡터 추정 방법으로는 손실된 매크로 블록의 움직임 벡터를 0이라 가정하는 것이 있다. 즉, 이전 프레임과 현재 프레임의 움직임이 거의 없다고 가정하는 것으로 이전 프레임의 같은 위치의 데이터로 대체하는 것이다^{11,12}. 이것은 움직임이 거의 없는 영상이나 배경 같은 영역에서는 좋은 성능을 기대할 수 있지만, 움직임이 많은 영상에서는 성능이 떨어진다. 움직임 벡터를 추정하는 다른 방법으로는 손실된 매크로 블록의 위치와 같은 위치의 이전 프레임에서의 움직임 벡터로 대체하는 것이 있다. 이것은 화면들간의 움직임이 일정한 경우에는 좋은 성능을 보인다. 또 다른 방법으로 움직임 벡터들의 공간적 상관도를 이용하는 것이 있다. 즉, 화면 내 개체들의 움직임은 대개 같은 방향을 가지므로 손실된 매크로 블록의 움직임 벡터를 추정하기 위해 그림 1과 같이 주위의 움직임 벡터들의 평균을 구한다¹³. 움직임 벡터들의 공간적 상관도를 이용하는 다른 방법으로는 주위의 움직임 벡터들의 중간값으로 대체하는 방법이 있다.

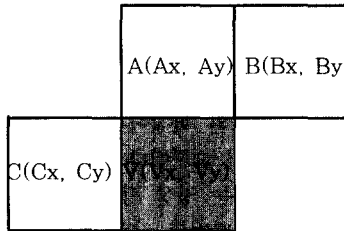


그림 1. 주변 블록의 움직임 벡터를 이용한 움직임 벡터 추정

3. 블록 경계 정합 방법(BMA)

블록 정합 알고리즘(block matching algorithm)은 공간적 상관성을 이용하여 손실된 블록 주변에 정확히 복호화된 블록들의 경계 화소값과 가장 잘 일치할 것으로 보이는 이전 프레임의 블록을 추정하는 방법이다^{15,14}. 블록 정합을 할 때 탐색 영역 안의 모든 블록에 대해 정해진 평가 함수를 적용하여

최소의 오차를 나타내는 블록의 위치를 움직임 벡터로 산정하는 방식을 FBMA(full search BMA)이라 한다¹⁵. FBMA 방식은 정확한 움직임 벡터를 찾는 데는 가장 적합한 방법이지만 계산량이 많아 수행 속도가 느리다는 단점이 있다¹⁶. 이를 보완하기 위한 고속 알고리즘으로 3SS(three step search), 4SS(four step search), CBOSA(center-biased orthogonal search algorithm)등이 있다^{17,18}.

본 논문에서는 제안하는 방법과의 비교를 위해 가장 성능이 좋은 FBMA를 사용하였다. FBMA 방법은 손실된 블록 주변의 블록과 이전 프레임의 탐색 영역 내에 있는 블록을 비교한다. 즉, 식 (1)의 MSE(mean square error) 값인 e를 구한 뒤 가장 작은 값을 가지는 데이터를 손실된 블록의 최적 움직임 벡터로 간주한다.

$$e = \sum_{i,j} (P'_{i,j} - P'_{i,j})^2 \tag{1}$$

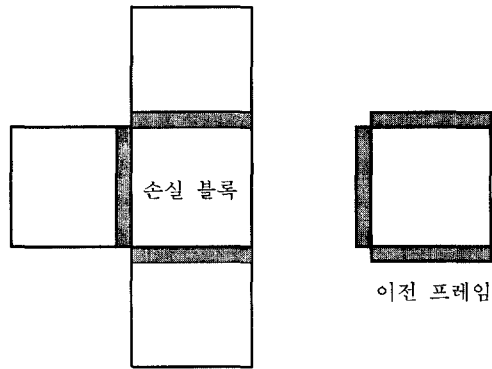


그림 2. 블록 경계 정합 방법

식 (1)의 $P'_{i,j}$ 는 그림 2의 왼쪽 그림의 색칠한 부분인 현재 프레임의 손실 블록 주변의 화소 값이고, $P'_{i,j}$ 는 그림 2의 오른쪽 그림의 색칠한 부분인 이전 프레임의 매크로 블록 주변의 화소 값이다.

4. 공간 예측 오류 은폐 기법

공간 예측 오류 은폐 기법은 주변 영역의 밝기 정보를 이용해서 주변 영역과 가장 잘 조화되도록 손상된 부분의 DCT 계수나 밝기 값을 복원하는 것으로 일종의 보간(interpolation) 기법이라 할 수 있다. 따라서 손상된 영역 내부의 정확한 정보를 주변 영역의 정보만으로 복원하는 것은 불가능하기 때문에 어느 정도 열화된 결과를 얻게 된다. 그러나 이

방법은 프레임내 영상이나 정지 영상의 경우, 움직임이 심한 경우, 프레임 간격이 큰 경우 등 시간 예측 오류 은폐 기법을 적용할 수 없는 경우에도 적용이 가능한 장점이 있다. 현재 제안되어 있는 기법은 손상된 영역의 밝기 값을 직접 복원하는 방법^{19,10}과 DCT 계수를 복원하는 방법¹⁸의 두 가지로 나눌 수 있다.

III. 방향성을 고려한 오류 은폐 방법

1. 경계 방향성을 고려한 움직임 벡터 추정

움직임 벡터 복원 방법 중에서 손실된 움직임 벡터를 주변 블록의 움직임 벡터의 평균 또는 중간값으로 대체하는 방법은 주변 블록들이 손실된 블록과 같은 객체일 경우는 좋지만 손실된 블록은 움직임이 있는데 주변 블록이 배경이거나 하여 움직임이 없을 경우는 성능이 좋지 않다. 그래서 본 논문에서는 이러한 경우에 성능을 향상시키기 위한 움직임 벡터 추정 방법을 제안한다.

제안된 방법의 수행 단계는 다음과 같다.

1. 주변 블록의 경계 방향성을 고려하여 현재 블록과 같은 객체에 속하는 블록을 추정한다.

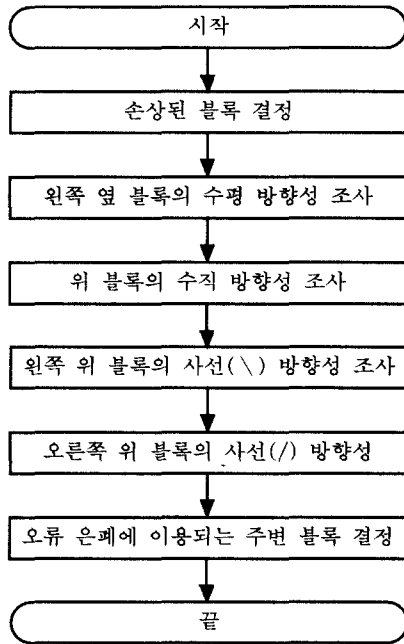


그림 3. 오류 은폐에 이용되는 주변 블록 결정 흐름도

2. 같은 객체로 추정된 블록이 하나일 경우 그 블록의 움직임 벡터로, 2 개 이상이면 그 블록들의 평균으로, 그리고 같은 객체로 추정되는 블록이 하나도 없다면 주변 네 개의 블록들의 움직임 벡터를 이용하여 이전 프레임과 BMA를 사용하여 손실 블록의 움직임 벡터를 추정한다.

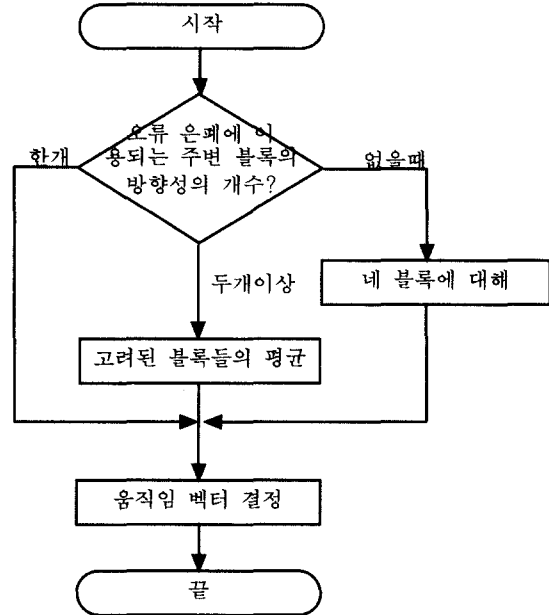


그림 4. 움직임 벡터 결정 흐름도

그림 3과 그림 4는 제안된 알고리즘의 흐름도이다. 그림 3은 주변 블록들의 경계의 방향성을 고려하여 현재 블록과 같은 객체에 속하는 블록을 추정하는 것으로 1단계에 해당된다. 그림 4는 고려되는 블록들을 이용하여 움직임 벡터를 추정하는 알고리즘의 흐름도로, 2단계에 해당된다. 다음 2절과 3절에서 각 단계에 대한 자세한 설명을 한다.

2. 주변 블록의 방향성 검사

경계 방향성을 고려하는데 사용한 주변 블록은 그림 5와 같이 손실된 매크로 블록 주변의 4개의 블록들이다. 그리고 경계 방향성 고려에는 그림 6에서와 같이 H_1 , V_1 의 두 개의 AC 계수를 고려하였다. 즉 (H_1, V_1) 과 그림 7의 13개의 대표 분류 벡터(class vector)와의 거리가 최소가 되는 것을 찾아서 방향성을 결정하였다^{19,20}. 그림 7의 13개의 분류 벡터는 경계의 방향성을 나타내주는 것으로 5개의 방향성으로 분류될 수 있다. 0번 분류 벡터는 방향성이 없는 블록으로 배경과 같은 블록을 나타내

고, 1~4번 분류 벡터는 대각선의 방향성이 있는 블록이다. 5, 7번 분류 벡터는 수평 방향성이 있는 블록을 나타내고, 6, 8번 분류 벡터는 수직 방향성이 있는 블록이며, 9~12번 분류 벡터는 강한 대각선의 방향성을 가진 블록이다. 만약 블록의 AC 계수를 (H_1, V_1) 라 하고, 그림 7의 한 대표 벡터를 (X_1, Y_1) 이라 한다면, 움직임 벡터와 그림 7의 대표 분류 벡터와의 거리를 식 (2)를 사용하여 구해서 거리가 가장 작은 값을 갖는 대표 분류 벡터로 방향성을 결정하는 것이다.

$$d = \sqrt{(H_1 - X_1)^2 + (V_1 - Y_1)^2} \quad (2)$$

예를 들면 위의 식을 이용하여 d를 구했는데 d 값을 가장 작게 만드는 분류 벡터가 1번이었다면 이 블록은 1와 같은 대각선의 방향성을 가지고 있다고 결정한다.

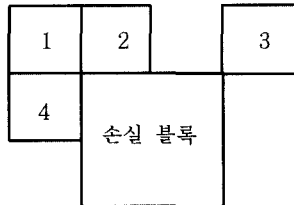


그림 5. 방향성을 고려한 블록들

	V_1						
H_1							

그림 6. DCT 계수

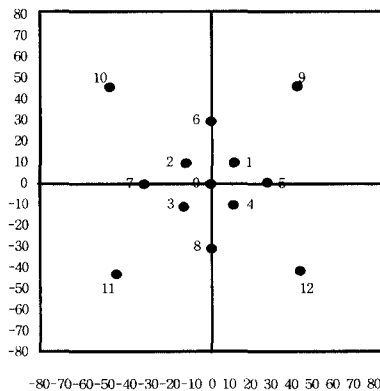


그림 7. V_1, H_1 특성 도메인에서 대표 분류 벡터

3. 움직임 벡터 추정

손실 블록의 경계 방향성과 비슷한 방향성을 가졌다고 고려할 수 있는 주변 블록의 경계 방향성은 그림 5의 1번 블록에서는 1와 같은 대각선 방향, 2번 블록은 수직방향, 3번 블록은 1와 같은 대각선 방향, 4번 블록은 수평방향이다. 예를 들면 1번 블록에서 만일 수평성분의 경계 방향성이 있었다면, 이것은 손실 블록과는 상관성이 적었을 것이다. 그 경계 방향성을 고려해 본다면, 1번 블록의 상하 블록보다 좌우 블록과 움직임이 더 많이 연결될 것이다. 다시 말하면, 만일 손실 블록이 수평 성분의 경계 방향성이 있었다고 한다면 이 블록의 움직임은 위아래 블록들의 움직임 벡터에 의해 추정된 블록보다 옆 블록의 움직임 벡터로 추정된 블록이 원래 블록과 더 일치할 것이다. 이 방법을 이용하여 각각의 블록에 맞는 방향성이 있는지를 찾아서 블록에 맞지 않는 블록은 제외하고, 맞는 블록들만을 고려하여 손실된 블록의 움직임 벡터를 추정하는 것이다. 만일 고려되는 블록이 1개일 경우는 그 블록의 움직임 벡터로 손실된 블록의 움직임 벡터를 추정하지만, 여러 개일 경우는 해당되는 블록들의 평균으로 하였다. 즉 2개 블록에서 경계의 방향성이 고려되면 이 2개 블록의 움직임 벡터의 평균으로 하고, 3개의 블록이 고려된다면 3개 블록의 움직임 벡터 평균으로 손실된 블록의 움직임 벡터를 추정한다. 그러나 주변 블록들이 다 경계 방향성을 갖지 않을 경우는 주변 블록들의 움직임 벡터로 손실된 블록을 추정하고 추정된 블록의 경계 블록들을 이용하여 BMA 알고리즘을 수행하여 이중 가장 잘 일치하는 블록의 움직임 벡터로 손실된 블록의 움직임 벡터를 추정하였다.

IV. 모의 실험 및 결과

1. 움직임 벡터 손실

앞장에서 제안한 오류 은폐 방법의 성능을 평가하기 위하여 다음과 같이 10 frames/sec QCIF 형식의 foreman 영상과 carphone 영상을 대상으로 32 kbits/s와 64 kbit/s의 전송률에서 모의 실험하였다. 본 논문에서 제안한 경계 방향성을 고려한 움직임 벡터 추정 방법을 평가하기 위하여 기존의 방법인 주위 블록들의 움직임 벡터의 평균으로 대체하는 방법(average), 주위 블록들의 움직임 벡터의 중간값으로 대체하는 방법(median), 블록 경계 정합 방법

(BMA) 등의 기존의 방법에 대한 실험도 같이 수행하였다. 또한 성능 평가 척도로서 식 (3)의 PSNR(Peak signal-to-noise ratio)을 사용하였다.

$$PSNR = 10 \log \left(\frac{255^2}{\frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} D(i,j)} \right) \quad (3)$$

$$D(i,j) = |x(i,j) - x^{recon}(i,j)|^2$$

이번 절에서는 움직임 벡터만이 손실되었다는 가정에 대하여 모의 실험을 하였다. 그림 8은 foreman 영상에 대해 32 kbits/s의 전송률을 사용한 경우에 한 프레임에서 두 개의 매크로 블록이 손실되었을 때의 결과이고, 그림 9는 carphone 영상에서 64 kbits/s의 전송률을 사용한 경우에 한 프레임에서 세 개의 매크로 블록이 손실되었을 때 모의 실험한 결과이다. 그림 10과 그림 11은 carphone 영상의 원 영상과 실험 결과 영상이다.

그림 8과 그림 9에서 보듯이 기존의 방법들 중에서는 BMA가 가장 우수한 성능을 보인다. 제안한 방법은 다른 기존의 방법들보다는 우수하고 BMA와 비슷한 성능을 보인다. 이렇게 median과 평균으로 움직임 벡터를 추정된 결과가 나쁜 것은 주변 블록들의 특성을 고려하지 않고 움직임 벡터를 추정했기 때문이다. 이에 반해 BMA와 제안한 방법은 주변 블록을 고려하여 비슷한 특성을 가진 블록을 이용하였기 때문에 median과 평균에 비해 성능이 좋게 나타난다. 그런데 본 논문에서는 제안한 방법과의 비교 대상으로 삼은 BMA 방식은 Ⅱ.3절에 나와있는 FBMA 방법을 사용하였다^{5,14)}. 즉, 모든 탐색 영역에 대해 손실된 매크로 블록 주변과 가장 잘 일치하는 식 (1)의 값을 최소로 만드는 데이터를 찾는 것이다. 그러나 이 방법이 계산량이 많은데 비하여 제안한 방법은 표 1과 같이 BMA의 계산량과 비교해 계산량이 많지 않다. N은 윈도우의 크기이고, B는 블록의 크기라하면, BMA는 항상 $N \times N \times 2B$ 의 곱셈을 필요로 한다. 2B는 블록의 네 변 중에서 두 변에 대해서만 이전 프레임의 블록과 비교를 하기 때문이다. BMA와 달리 제안한 방법은 최소일 경우 같은 객체로 추정되는 블록을 분류하는데 $4 \times 13 \times 2$ 로 104번의 곱셈이 필요하고 최대일 경우는 같은 객체로 추정되는 블록을 분류하는 것과 추정되는 블록이 존재하지 않아 4개의 블록에 대해 BMA를 행하는데 따른 $4 \times 13 \times 2 + 4 \times 2B$ 의 곱셈이 필요하다. 본 논문에서는 윈도우의 크기를 32로 했

으므로 N은 32이고 블록은 매크로블록단위로 했으므로 B는 16이다. 이런 경우 최소일 때 BMA가 32768인 것에 비해 제안한 방법이 104로 많은 곱셈수가 감소했다. 최대의 경우 역시 BMA가 32768이고 제안한 방법이 232로 BMA에 비해 많은 곱셈수가 감소했다.

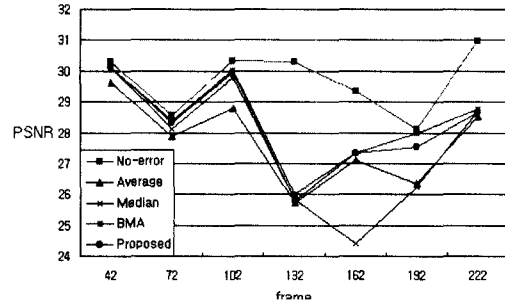


그림 8. PSNR 비교 (Foreman, 한 프레임에서 2 개의 MB 이 손상되었을 때, 32kbps)

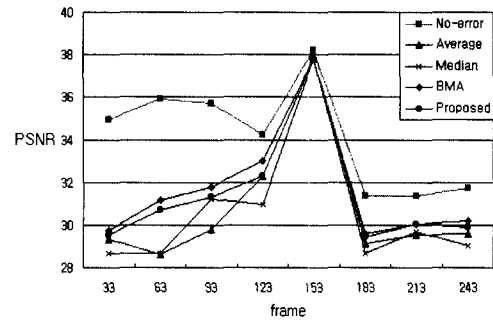
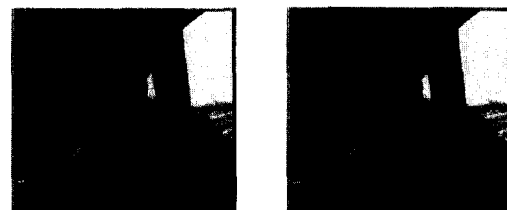


그림 9. PSNR 비교 (Carphone, 한 프레임에서 3개의 MB 이 손상되었을 때, 32kbps)

표 1. BMA와 제안한 방법의 곱셈수 비교

	최소	최대
BMA	32768	32768
제안한 방법	104	232



(a) 원 영상 (b) 블록 손상

그림 10. 원 영상과 손상된 영상(carphone)

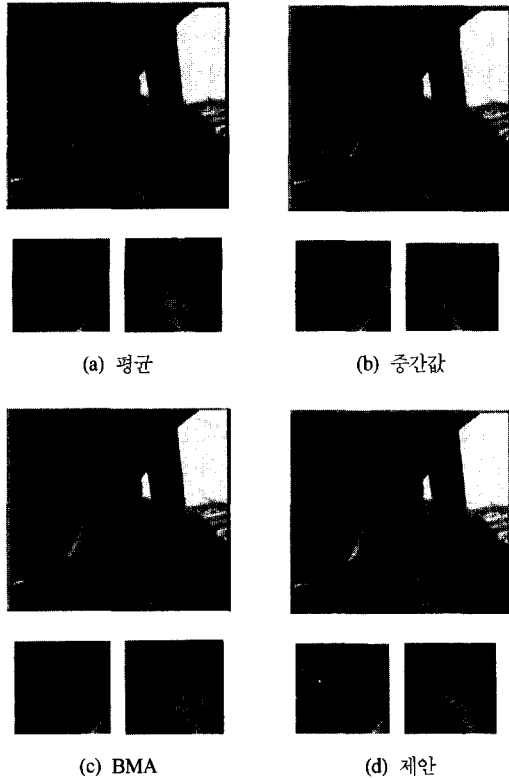


그림 11. Carphone 영상의 72번째 프레임 복원 결과

2. 움직임 벡터와 DCT 계수 모두 손실

앞 절에서 움직임 벡터만이 손실된 경우에 대해 실험한데 비해서 여기서는 움직임 벡터와 DCT 계수가 모두 손실되었을 경우에 대해 모의 실험하였다. 움직임 벡터를 추정하여 추정한 움직임 벡터로 이전 프레임에서 블록을 찾아 블록 전체를 손실된 블록으로 대처하도록 하였다. 그림 12는 carphone 영상에 대해서 한 프레임에서 한 개의 매크로 블록이 손실되었을 경우에 대해 32 kbits/s에서 모의 실험

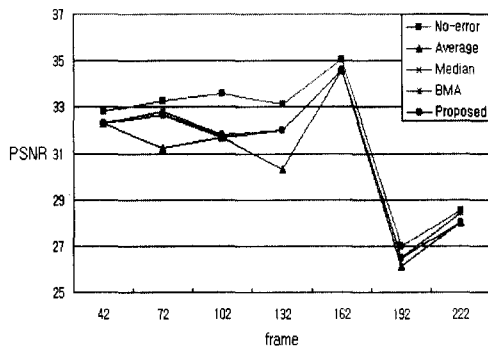


그림 12. PSNR 비교 (Carphone, 한 프레임에서 1 개의 MB 이 손상되었을 때, 32kbps)

한 결과이고, 그림 13은 foreman 영상에 대해 64 kbits/s에서 한 프레임에서 네 개의 매크로 블록이 손실되었을 때의 결과이다. 그림 14와 그림 15는 foreman 영상의 원 영상과 실험 결과 영상이다.

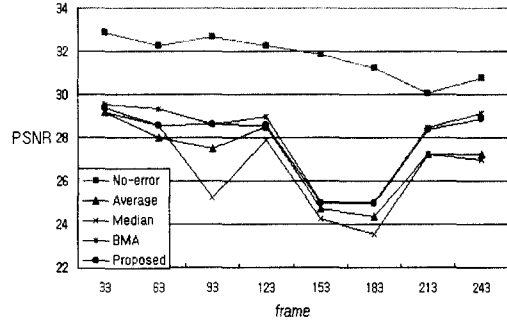
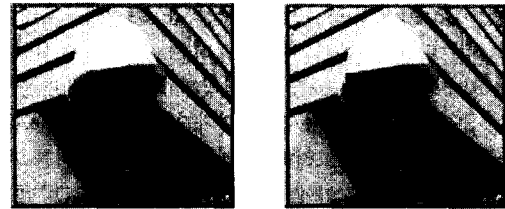


그림 13. PSNR 비교 (Foreman, 한 프레임에서 4 개의 MB이 손상되었을 때, 64kbps)



(a) 원 영상 (b) 블록 손상

그림 14. 원 영상과 손상된 영상 (foreman)

모의 실험 결과 움직임 벡터만 손실되었을 경우와 마찬가지로 BMA와 제안한 방법이 다른 방법들보다 PSNR이 높았다. 하지만 전체적으로 움직임 벡터만 손실되었을 경우보다 PSNR이 낮았다. 이렇게 PSNR이 움직임 벡터만 손실되었을 경우보다 낮게 나온 것은 추정된 움직임 벡터로 이전 프레임에서 블록을 찾아 그대로 대처하였기 때문에 주변 블록과 경계 부분이 잘 맞지 않기 때문이다. 그래도 움직임 벡터를 정확히 찾았을 경우는 시각적으로나 PSNR 측면에서 큰 영향은 없었다. 그러므로 움직임 벡터만 잘 추정하여도 손상된 블록을 잘 복원할 수 있음을 알 수 있다.

V. 결론

본 논문에서는 경계 방향성을 이용하여 손상된 매크로 블록의 움직임 벡터를 추정하는 기법을 제안하였다. 즉, 손상된 블록과 비슷한 방향성을 가진

블록을 이용하여 움직임 벡터를 추정하는 기법을 제안하였는데 손상된 매크로 블록의 주변 블록의 경계 방향성을 고려하기 위해서 각 블록의 AC계수를 이용하여 각 블록의 방향성을 분류하였다.

제안한 알고리즘의 성능을 평가하기 위해서 다양한 오류 은폐 기법에 대하여 모의 실험을 통하여 주관적, 객관적 복원 화질을 평가하였다. 평가 결과 기존의 기법 중에서 가장 우수한 성능을 보이는 BMA와 비슷한 성능을 보였다. BMA가 계산량이 많은 것을 감안하면, 제안한 방법이 BMA보다 계산량이 적으면서 BMA와 비슷한 결과를 보이기 때문에 효율적인 방법으로 판단된다. 본 논문을 바탕으로 블록 손실로 에러가 전파될 때의 오류 은폐 기법에 대한 연구가 필요하다고 생각된다.

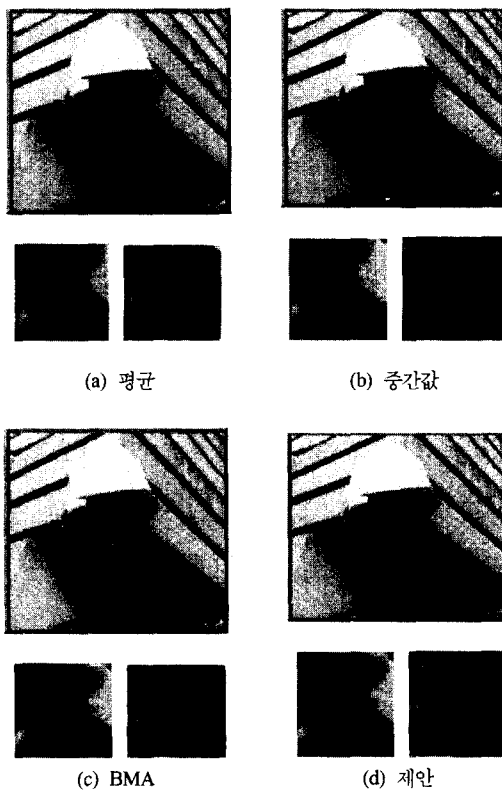


그림 15. foreman 영상의 93번째 프레임 복원 결과

참 고 문 헌

[1] P. Bahi, B. Girod, "Wireless video," *IEEE Commun. Mag.*, pp.92~93, June 1998.
 [2] ITU-T Rec. H.324, "Terminal for low bitrate multimedia communication," 1995.

[3] D. W. Redmil and N. G. Kingsbury, "The EREC : An error-resilient technique for coding variable-length blocks of data," *IEEE Trans. Image Processing*, vol.5, pp. 565-574, Apr. 1996.
 [4] M.-J. Chen, L.-G. Chen, and R.-M. Weng, "Error concealment of lost motion vectors with overlapped motion compensation," *IEEE Trans. Circuits Syst. video Technol.*, vol.7, no.3, pp.560~563, June 1997.
 [5] W. M. Lam, A. R. Reibman and B. Liu, "Recovery of lost or erroneously received motion vector," *ICASSP*, vol.5, pp.417~420, 1993.
 [6] W. M. Lam, A. R. Reibman, "An error concealment algorithm for images subject to channel errors," *IEEE Trans. on Image Processing*, vol.4, pp.533~542, May 1995.
 [7] 박종욱, "영상 정보 통신을 위한 전송 오류 복구 기법," 서울대학교 박사 학위 논문, Feb. 1998.
 [8] Y. Wang and Q. Zhu, "Signal loss recovery in DCT-based image and video codecs," in *Proc. of SPIE Visual Communications and Image Processing '91*(Boston, MA, USA.), pp.667~678, Nov. 1991.
 [9] X. Lee, Y. -Q. Zhang, and A. Leon-Garcia, "Information loss recovery for block-based image coding techniques - a fuzzy logic approach," *IEEE Trans. Image Processing*, vol.4, pp.259~273, Mar. 1995.
 [10] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projections onto convex sets," *IEEE Trans. Image Processing*, vol. 4, pp.470~477, Apr. 1995.
 [11] A. Narula and J. Lim, "Error concealment techniques for an all-digital high-definition television system," *SPIE Visual Communication and Image Processing*, pp.304~315, Nov. 1993.
 [12] S. Aign and K. Fazel, "Temporal and spatial error concealment techniques for hierarchical MPEG-2 video codec," *IEEE International Conference on Communication*, vol.3, pp.1178~1783, June 1995.

