

# 동일 움직임 영역기반에 중복블록움직임보상을 적용한 움직임 벡터 복원 기법

손남례\*, 김정현, 박성찬, 이귀상  
전남대학교 전산학과  
e-mail:nrson@cs.chonnam.ac.kr

## Motion Vector Recovery Using OBMC Based on Homogeneous Motion Blocks

NamRye Son\* JhugHyun Kim SungChan Park GueeSang Lee  
Dept. of Computer Science, Chonnam National University

### 요 약

이동 통신 채널과 같이 에러 발생율이 높은 환경에서 부호화된 비디오 스트림을 전송시 발생된 에러는 비디오 화질에 큰 영향을 줄 수 있다. 본 논문에서는 현재 널리 사용되고 있는 H.263의 복호화기에서 전송도중 에러가 발생했을 경우 추가적인 데이터 스트림의 삽입 없이 효율적으로 에러를 은닉할 수 있는 기법에 관하여 연구하였다. 특히 인터프레임 영상에서 손상되거나 손실된 움직임벡터에 대해 시간영역에서 관계가 깊은 이전 프레임에서 손실블록과 같은 위치에 있는 블록의 움직임벡터와 현재 프레임에서 손실블록과 인접한 블록들의 움직임벡터 정보를 이용하여 이 블록들의 움직임 방향을 구하게된다. 이러한 블록의 움직임 방향을 이용하여 동일한 움직임 방향을 갖는 영역을 찾아 손실 블록이 어떤 움직임 영역에 속하는 지를 판단하여 손실된 움직임 벡터를 복원하는 방법을 제안한다. 또한 제안한 알고리즘에 중복블록 움직임 보상(Overlapped Block Motion Compensation)을 적용하여 블록화 현상을 줄였다.

### 1. 서 론

이동 통신 채널과 같은 저대역 통신망에서 비디오 전송을 위해서는 높은 압축율을 갖는 부호화 방법들이 사용된다. 이러한 방법들로 부호화된 비디오가 에러율이 높은 통신 채널로 전송될 때, 비트열에 에러가 발생하면 복호되기 어려울 뿐 아니라 다른 부분까지 영향을 미치게된다. 이러한 문제들에 대한 해결책으로 ARQ(Automatic Retransmission Request), FEC(Forward Error Correction), 그리고 interleaving 기법 등이 제안되었다. 그러나 ARQ는 재전송을 하기까지 Delay time이 발생하게되고 오히려 채널에 오버헤드를 가중시킬 수 있으며, FEC는 에러 감지나 에러 은닉을 위한 추가 패리티비트들이 필요하게 되고 인터리빙 기법은 계산시간이 오래 걸리는 문제점들이 있다[1].

따라서 본 논문에서는 통신 채널상의 에러를 수신측에서는 에러의 영향을 최소화 에러은닉 방법을 제안한다. 즉 손실블록의 주변블록 중에서 같은 움직임을 갖는 블록들로 구성된 동일 움직임 영역을 추출하고, 이 블록들을 이용하여 손실블록의 움직임벡터를 복구한다. 공간적으로 인접한 블록간에는 움직임 벡터의 상관성이 높기 때문에 손

실블록 또한 이 블록들과 움직임이 비슷할 가능성이 높다. 이러한 특성을 이용하여 주변블록들 중 동일한 움직임을 갖는 영역이 있을 때 손실블록 또한 이 영역에 포함되는 지를 조사하여 포함되는 경우 이 영역에 속한 블록들의 움직임 벡터를 이용하여 손실블록의 움직임벡터를 복원하게 된다[5]. 또한 제안한 알고리즘에 중복블록 움직임 보상(OBMC)을 적용하여 현재의 블록 화소값과 주변 블록의 움직임 벡터를 고려한 블록 화소값을 가중평균하여 블록화 현상을 줄였다.

실험은 저대역폭 통신을 위한 비디오 부호화 표준으로 널리 이용되고 있는 H.263 부호화를 기반으로 하였다. 본 논문의 구성은 다음과 같이 2장에서는 블록정합기법인 BMA에 대해 언급하고, 3장에서는 실제로 영상의 움직임 벡터가 동일한 방향으로 움직이는지 예측하고 제안한 움직임 벡터 복원기법(HMA: Homogeneous Motion Vector)과 OBMC를 설명한다. 4장에서는 제안된 방법에 대한 실험결과를 제시하고, 마지막으로 5장에서는 결론을 맺는다.

### 2. 기존 연구

BMA(Block Matching Algorithm)는 손실블록의 주변에 빠르게 복호화 되어 있는 블록과 손실 블록과의 경계선간

\* 본 연구는 정보통신부 2000년 대학기초 연구사업지원(071-02)에 의하여 수행되었음.

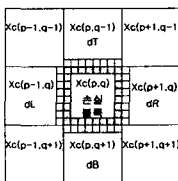
의 유사성을 고려하여 가장 유사한 블록경계선을 갖는 블록으로 손실블록을 대체한다[2][3].

이 방법은 손실된 블록의 움직임 벡터를 후보벡터중 임의의 값으로 정하여 이전 프레임으로부터 손실된 블록을 보상한 후에, 그 보상된 블록의 경계값과 손실된 블록에 인접한 주변블록들의 경계값을 비교한다. 이때 그 경계값들이 가장 유사한 움직임 벡터를 손실된 블록의 움직임 벡터로 추정하여 블록 영상을 복구하게된다. 즉 (식1)에서  $dSum$  값을 최소로 하는 것을 움직임벡터로 추정하는 것이다. 여기에서 사용되는 후보 움직임벡터는 다음과 같다.

- (a) 전 프레임에서 손실블록과 같은 위치에 있는 블록의 MV
- (b) 인접블록의 움직임 벡터의 중간 값
- (c) 인접블록의 움직임 벡터의 평균 값
- (d) 0 (zero 벡터)

$$\begin{aligned}
 dT &= \sum_{j=1}^{row} [X_{MC}(p, q)]_{i,j}^{row} - [X(p, q-1)]_{i,j}^{row} \\
 dL &= \sum_{j=1}^{row} [X_{MC}(p, q)]_{i,j}^{col} - [X(p-1, q)]_{i,j}^{col} \\
 dR &= \sum_{j=1}^{row} [X_{MC}(p, q)]_{i,j}^{col} - [X(p+1, q)]_{i,j}^{col} \quad (식1) \\
 dB &= \sum_{j=1}^{row} [X_{MC}(p, q)]_{i,j}^{row} - [X(p, q+1)]_{i,j}^{row} \\
 dSum &= dT + dB + dL + dR
 \end{aligned}$$

프레임에서  $(p, q)$ 에 위치한 크기가  $n \times n$ 인 블록을  $X(p, q)$ , 이 블록에 해당하는 움직임 벡터를  $V(p, q)$ 라고 했을 때 만일  $V(p, q)$ 가 손상되었을 경우 복원된 움직임 벡터를  $\hat{V}(p, q)$ 라고 하고  $[X_{MC}(p, q)]$ 는 움직임 보상된 값을 나타낸다.  $[X(p, q)]_{i,j}^{col}$ 과  $[X(p, q)]_{i,j}^{row}$ 은 블록  $X(p, q)$ 의  $i$ 번째 column에서의 벡터와  $j$ 번째 row에서의 벡터를 나타낸다. 그림 1은 매크로블록에 대하여 BMA를 적용할 때의 블록 경계 화소들을 보여준다.



[그림 1] BMA에 사용되는 경계 선택 방법

### 3. 제안된 움직임 벡터 복원 기법

#### 3.1 블록의 움직임 방향과 크기

블록의 움직임은 블록의 움직임 방향과 크기를 고려해야 하는데 이것은 블록의 움직임벡터를 이용하여 얻을 수 있다. 블록의 움직임 크기는 움직임 벡터의 x좌표와 y좌표를 유클리디언 공식에 적용시켜 얻을 수 있고, 블록의 움직임 방향은 움직임 벡터의 기울기를 구하여 이것을 8개의 방향 중 하나로 맵핑시킴으로써 구할 수 있다.

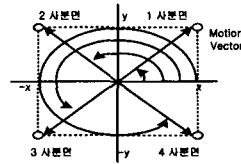
(식2)와 (식3)은 각각 움직임벡터의 크기와 각도를 구하는 방법이고, x, y는 각각 움직임벡터의 x좌표와 y좌표를 의미한다. (식3)을 이용하여 얻은 움직임벡터의 각도를 그

림 2 처럼 좌표평면상에서 움직임 벡터의 x,y좌표가 어떤 사분면에 위치하느냐에 따라 움직임벡터의 기울기를 구할 수 있다. 참조 논문[7]과 같이 구해진 사분면의 위치에 따른 움직임벡터의 기울기와 기울기 값에 따라 각도의 범위에 맞추어 8가지 방향 중 하나로 설정함으로써 블록의 움직임 방향을 구하였다.

$$d = \sqrt{x^2 + y^2} \quad (식2)$$

$$Radian = \arcsin\left(\frac{|y|}{\sqrt{x^2 + y^2}}\right) \quad (식3)$$

$$MV\_Angle = Radian \times 180$$



[그림 2] 사분면의 위치에 따른 움직임벡터의 기울기

공간적으로 인접한 블록간에는 움직임 벡터의 상관성이 높기 때문에 손실블록 또한 이 블록들과 움직임이 비슷할 가능성이 높다. 따라서 영상에 대한 MV 값이 인접한 블록과 동일 방향 움직임을 증명하기 위하여 영상에 대한 MV 값을 MB단위로 그림 3과 같이 화면에 화살표로 표시하였다.



(a) forman의 40번째 프레임 (b) suzi의 15번째 프레임

[그림 3] 영상을 MB단위로 MV출력

#### 3.2 제안 알고리즘

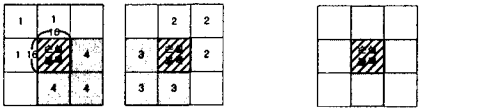
동일 움직임 영역을 찾기 위해 손실블록의 주변블록들을 미리 그림 4의 (a)처럼 4개의 영역으로 구분한다. 각 영역에 포함된 블록들의 움직임 방향과 크기를 위와 같은 방법으로 조사하여 만일 한 영역내에 포함된 모든 블록들의 움직임 방향이 같고, 블록의 움직임 크기간 차이가 임계값 이하일 경우 이 영역을 동일 움직임 영역(HMA)으로 인식한다. 동일 움직임 영역을 찾은 후 이 영역에 포함된 블록을 참조하여 손실블록의 움직임 벡터를 구하게되는데, (식4)와 같이 참조블록들의 움직임벡터 평균을 구하여 평균과의 차분 값이 최소인 것을 손실블록의 움직임 벡터로 복원한다. 여기에서  $MV_i$ 는 블록들의 움직임 벡터를 나타낸다.

$$Ave = \left(\sum_{i=0}^3 MV_i\right) / 3 \quad (식4)$$

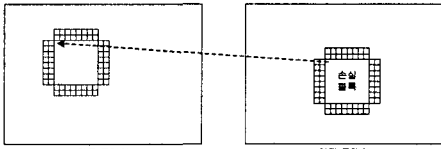
$$Argmin_{i=1 \dots 3} (MV_i - Ave)$$

손실블록이 실제 이 움직임영역과 같은 움직임을 갖는지 판단하기 위해 움직임이 독립적일 때와 비교한다. 이때

비교방법은 각각의 방법으로 복원되었을 때 왜곡 (Distortion)을 이용한다. 왜곡은 그림 6와 같이 손실블록에 인접한 주변블록 픽셀과 복원된 움직임벡터를 이용하여 참조 프레임에서 손실블록의 움직임 보상 위치를 찾아 이 블록의 주변픽셀의 값을 비교하여 이 픽셀들의 차분의 합으로 계산한다[8]. 이때 움직임이 독립적일 때의 움직임 벡터는 그림 5의 (b)에서 진하게 색칠된 블록을 참조하여 이 블록들의 중간값으로 복원하게 된다.

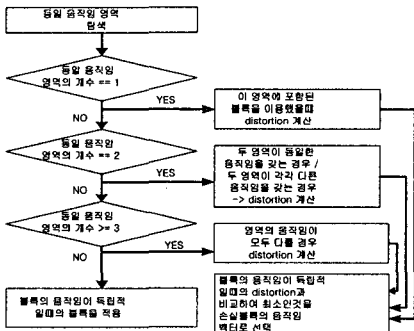


(a)동일 움직임 영역 구분 (b) 움직임이 독립적일 때 참조 블록  
[그림 5] 동일 움직임영역구분과 참조 블록

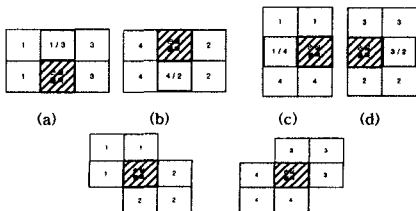


[그림 6] 제안방법에서 사용되는 주변블록의 픽셀

그림 6는 제안 알고리즘 순서도를 간단히 나타낸 것이다. 그림 7의 (a)~(f)는 동일 움직임이 2개인 경우이고 이때 (e)와 (f)는 손실블록을 특정 그룹에 포함시키기 어렵기 때문에 참조하는 블록들이 블록의 움직임이 독립적일 때와 같게 한다[6].



[그림 6] 제안 알고리즘 순서도

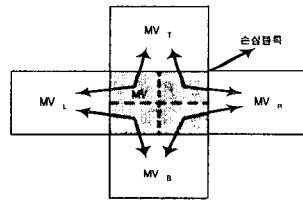


[그림 7] 동일 움직임 영역이 2개일 때 참조블록

### 3.3 OBMC(Overlapped Block Motion Compensation)

대부분의 영상 부호화 방법은 효과적인 부호화를 위하여 영상을 16×16 크기로 분할하여 움직임 벡터를 구하고, 영상을 8×8 크기 단위로 부호화하기 때문에, 수신측에서 복호된 영상은 이러한 블록 경계가 왜곡되는 블록화 현상이 필연적으로 발생하게 된다. 따라서 OBMC를 적용한 에러내성 기법에서는 현재의 블록 화소값과 주변 블록의 움직임벡터를 고려한 블록 화소값을 가중 평균하여 블록화 현상을 줄인다[4].

N×N 크기의 손실블록은 네 개의 N/2×N/2 서브블록 (Subblock)으로 분할되며, 각 서브블록은 세 개의 예측된 N/2×N/2 블록의 가중치 평균에 의해서 통합된다. 첫 번째 N/2×N/2 블록은 손실된 블록을 위해 동일 움직임영역 기반 복원기법으로 추정된 움직임벡터를 이용하여 보상된 블록이고, 두 번째는 손실된 블록에 수평으로 이웃한 N×N 블록의 움직임벡터에 의해서 보상된 N/2×N/2 블록이며, 세 번째는 손실된 블록에 수직으로 이웃한 N×N 블록의 움직임벡터에 의해서 보상된 N/2×N/2 블록이다.



[그림 8] OBMC에서 사용되는 주변블록의 움직임벡터 선택방법

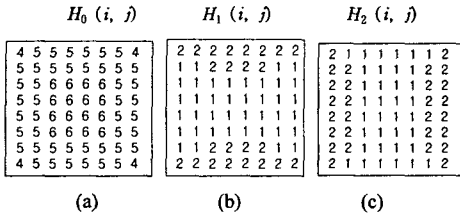
그림 8은 중복 블록 움직임 보상에 사용되는 움직임벡터의 선택 방법을 보여주며,  $MV_U$ ,  $MV_D$ ,  $MV_L$ ,  $MV_R$ 는 각각 손실된 블록의 위, 아래, 왼쪽, 오른쪽에 이웃한 매크로블록의 움직임벡터를 의미한다. 손실된 블록은 크기가  $N/2 \times N/2$ 인 4개의 작은 블록으로 나누어 그림 8에 나타난 움직임벡터 선택 방법에 따라 중복 블록 움직임 보상이 이루어진다. 동일 움직임영역을 이용하여 구한 움직임 벡터를  $MV^0$ , 위 또는 아래 블록의 움직임 벡터를  $MV^1$ , 왼쪽 또는 오른쪽에 위치한 블록의 움직임 벡터를  $MV^2$ 라 하고,  $MV=(MV_x, MV_y)$ 라고 정의한다. 이때  $MV^0, MV^1, MV^2$ 에 의해서 보상된 블록의 (i, j) 위치의 각 화소값  $q(i, j), r(i, j), s(i, j)$ 를 구하는 방법은 (식5)와 같다.

$$\begin{aligned}
 q(i, j) &= p(i + MV_x^0, j + MV_y^0) \\
 r(i, j) &= p(i + MV_x^1, j + MV_y^1) \\
 s(i, j) &= p(i + MV_x^2, j + MV_y^2)
 \end{aligned}
 \tag{식5}$$

$$P(x, y) = (q(i, j) \times H_0(i, j) + r(i, j) \times H_1(i, j) + s(i, j) \times H_2(i, j) + 4) / 8
 \tag{식6}$$

(식5)에 의해서 얻어진 화소값들을 이용하여 중복 블록 움직임 보상된 화소값  $P(i, j)$ 는 (식6)을 이용하여 구할 수

있다. 이때  $H_0(i, j)$ ,  $H_1(i, j)$ ,  $H_2(i, j)$ 는 중복 블록 움직임 보상에 사용되는 가중치 행렬을 나타내며, 그림 9는 손실된 블록이 16×16일 때의 가중치 행렬의 예를 보여주고 있다.



(a) (b) (c)  
[그림 9] OBMC에 사용되는 가중치 행렬

- (a) 현재 luminance 블록의 MV와 예측을 위한 가중치
- (b) 현재 블록의 상·하에 위치한 luminance 블록의 MV와 예측을 위한 가중치
- (c) 현재 블록 좌·우에 위치한 luminance 블록의 MV와 예측을 위한 가중치

#### 4. 실험 결과

실험은 각 176×144 pixels/lines의 Suzie, Carphone 영상에 대해 수행하였다. 한 프레임은 9개의 GOB로 구성되어있고, 한개의 GOB는 11개의 매크로블록으로 구성되어 있다. 따라서 실제로 부호화된 영상은 GOB 단위로 에러가 나타날 수 있어 다음 GOB의 Resync가 나타날때까지 한 개의 GOB 손실이 있으므로 즉 11개의 매크로블록 모두 손실이 되기 때문에 실험데이터를 GOB 손실에 대한 영상을복원하는 실험을 하였다. (표1)은 각 영상의 29번째 프레임에 GOB의 손실이 있을 때 각각 실험한 데이터의 PSNR이다. 에러가 발생했을 때 에러를 하지 않고 복원한 경우와 BMA, HMA에 각각 OBMC를 적용한 실험 데이터 결과이다. 그림 10에서는 GOB의 3,5,7번째 손실이 있을 때 Suzie 영상을 이용하여 복원한 실험한 결과이다.

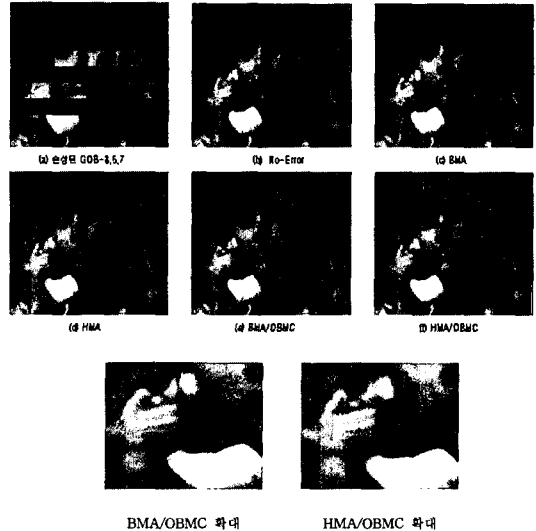
[표 1] 176×144 크기 영상의 GOB 손실에 따른 PSNR 비

영상	GOB Loss Num	no-EC	BMA	HMA	BMA/OBMC	HMA/OBMC
Suzie	5	57.95	58.32	60.39	58.37	60.66
	5,6	56.62	56.96	58.91	57.02	59.05
	5,7	55.60	55.82	56.66	55.85	56.72
	3,5,7	53.64	53.52	54.96	53.56	55.02
Carphone	5	56.30	56.02	60.18	56.30	60.39
	5,6	49.32	48.85	51.67	49.06	51.85
	5,7	52.87	51.84	59.49	52.87	60.25
	3,5,7	46.21	46.04	49.47	46.87	49.94

#### V. 결론

이동 통신 채널과 같은 저대역 통신망에서는 비디오 전송을 위해서는 높은 압축율을 갖는 부호화 방법들이 사용되는데, 본 논문에서는 저대역폭 통신을 위한 비디오 부호

화 표준인 H.263 부호화를 기반으로 실험하였다. 본 논문에서는 전송도중 손상된 움직임 벡터의 복원기법을 제안하고 실험하였다. 제안알고리즘은 기존의 BMA에 비하여 PSNR과 시각적인 화질면에서 우수한 성능을 보임을 알 수 있었고 제안방법으로 복원한 영상에 OBMC 기법을 추가하여 복원했을 경우 블록화 현상을 더 감소시킬수 있었다.



[그림 10] Suzie(176×144) 영상의 실험 결과 영상

#### 참고 문헌

- [1] J.w. Suh, Y.S. Ho, "Recovery of motion vectors for error concealment", IEEE Region 10 Conference Vol. 1, pp. 750-753, 1995
- [2] W.M. Lam, A.R. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors", Proc. ICASSP, 5, pp. 417-420, 1993
- [3] Y. Wang and, Qin-Fan Zhu, and Leonard Shaw, "Maximally smoothness image recovery in transform
- [4] M.-J. Chen, L.-G. Chen, and R.-M. Weng, "Error Concealment of Lost Motion Vectors with Overlapped Motion Compensation," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 7, No. 3, pp. 560-563, June 1997.
- [5] M.J. Chen, L.G. Chen, and R.M. Weng, "Error concealment of lost motion vectors with overlapped motion compensation", IEEE Trans. on circuits and systems for video technology, Vol 7, No. 3, June 1997
- [6] 김정현, 박성찬, 이귀상, "H.263 비디오 통신을 위한 움직임 벡터 복원", 제13회 신호처리학회, Vol 13, No.1, pp.841, 2000.
- [7] J.Y. Nam, J.S. Seo, J.S. Kwak, M.H. Lee, "New Block-Matching Algorithm for Motion Estimation Based on Prediction Direction Information", Visual Communications and Image Proc. 2000, Proc. of SPIE, pp. 1250-1260, 2000.
- [8] Yu, H. R. Wu, and S. Yu, "Lost motion vector recovery for digital video communication," Visual Communications and Image Proc. 2000, Proc. of SPIE, pp. 1250-1260, 2000.