

동일 움직임 블록 검출을 이용한 움직임 벡터 복원 기법

김정현 ◦ 박성찬 이귀상
전남대학교 전산학과
전화 : 062-530-0147

Motion Vector Recovery Using Extraction of homogeneous motion blocks

Jung-Hyun Kim Sung-Chan Park Guee-Sang Lee
Dept. of Computer Science, Chonnam National University
E-mail : jhkim@cs.chonnam.ac.kr

Abstract

Bitstreams corrupted by channel errors are not only difficult to be decoded but also propagate error to other part of the bitstreams when highly compressed video is transmitted over channels with noise such as mobile communication channels. In this paper, error concealment algorithm performed in decoder is proposed when errors occur for transmission. Proposed algorithm searches moving area with homogeneous movement in neighbored blocks when motion vectors are damaged, then recovers motion vectors of missing blocks considering where missing blocks are belong to. Experiment result shows that proposed algorithm exhibits better performance in PSNR than existing error concealment method.

I. 서론

이동 통신 채널과 같은 저속 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 미치게 된다.

이러한 문제들에 대한 해결책으로 ARQ(Automatic Retransmission Request), FEC(Forward Error Correction), 그리고 interleaving 기법 등이 제안되었다. 그러나 ARQ는 재전송을 하기까지 Delay time이 발생하게 되고 오히려 채널에 오버헤드를 가중시킬 수 있으며, FEC는 에러 감지나 에러 은닉을 위한 추가 패리티비트들이 필요하게 되고 인터리빙 기법은 계산 시간이 오래 걸리는 문제점들이 있다[1].

따라서 본 논문에서는 수신측에서 에러의 영향을 최소화하는 방법을 제안한다. 즉 손실블록의 주변블록 중에서 같은 움직임을 갖는 블록들로 구성된 동일 움직임 영역을 추출하고, 이 블록들을 이용하여 손실블록의 움직임벡터를 복구한다. 공간적으로 인접한 블록 간에는 높은 움직임 벡터의 상관성이 있기 때문에 손실블록 또한 이 블록들과 움직임이 비슷한 가능성이 높다.

이러한 특성을 이용하여 주변블록들 중 동일한 움직임을 갖는 영역이 있을 때 손실블록 또한 이 영역에 포함되는지를 조사하여 포함되는 경우 이 영역에 속한 블록들의 움직임 벡터를 이용하여 손실블록의 움직임 벡터를 복원하게 된다. 만일 손실블록이 동일 움직임 영역에 포함되지 않고 독립적인 움직임을 갖는 경우에는 손실블록의 상하좌우에 직접적으로 인접한 블록들의 움직임벡터를 이용하여 손실블록의 움직임 벡터를 구하게 된다.

※ 본 연구는 한국과학재단 목적기초연구과제 (2000-1-30300-005-3)에 의하여 수행되었음.

실험은 저대역폭 통신을 위한 비디오 부호화 표준으로 널리 이용되고 있는 H.263 부호화를 기반으로 하였다. 본 논문의 구성은 2장에서는 BMA에 대해 언급하고, 3장에서는 제안된 움직임 벡터 복원기법을 설명한다. 4장에서는 제안된 방법에 대한 실험결과를 제시하고 5장에서는 결론을 맺는다.

II. 기존 연구

BMA는 손실블록의 주변에 바르게 복호화 되어 있는 블록과 손실 블록과의 경계선간의 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 유사해지는 움직임 벡터를 손실된 블록의 움직임 벡터로 추정하여 블록 영상을 복구하게 된다. 즉, 식1에서 $dSum$ 값을 최소로 하는 것을 움직임 벡터로 추정하는 것이다. 여기에서 사용되는 후보 움직임벡터는 다음과 같다.

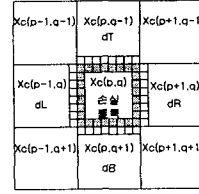
- (a) 전 프레임에서 손실블록과 같은 위치에 있는 블록의 움직임 벡터
- (b) 인접블록의 움직임 벡터의 중간 값
- (c) 인접블록의 움직임 벡터의 평균 값
- (d) 0 (zero 벡터)

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

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 움직임벡터를 $\hat{V}(p, q)$ 라하고 $[X_{MC}(p, q)]$ 는 움직임 보상된 값을 나타낸다. $[X(p, q)]_{l,j}^{col}$ 과 $[X(p, q)]_{1,j}^{row}$ 은 블록 $X(p, q)$ 의 l 번째 column에서의 벡터와 l 번째 row에서의 벡터를 나타낸다.

$$\begin{aligned}
 [X(p, q)]_{1,j}^{col} &= (x_{11}, x_{12}, x_{13}, \dots, x_{1n}) \\
 [X(p, q)]_{1,j}^{row} &= (x_{11}, x_{21}, x_{31}, \dots, x_{n1})
 \end{aligned}$$

그림 1은 매크로블록에 대하여 BMA를 적용할 때의 블록 경계 화소들을 보여준다.



(그림 1). BMA에 사용되는 경계 선택 방법

III. 제안된 움직임 벡터 복원 기법

3.1 블록의 움직임 방향과 크기

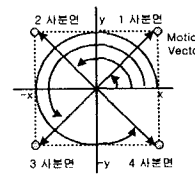
블록의 움직임은 블록의 움직임 방향과 크기를 고려해야 하는데 이것은 블록의 움직임벡터를 이용하여 얻을 수 있다. 블록의 움직임 크기는 움직임 벡터의 x좌표와 y좌표를 유클리디언 공식에 적용시켜 얻을 수 있고, 블록의 움직임 방향은 움직임 벡터의 기울기를 구하여 이것을 8개의 방향 중 하나로 맵핑시킴으로써 구할 수 있다.

식 2와 3은 각각 움직임벡터의 크기와 각도를 구하는 방법이다. 이때 x, y는 각각 움직임벡터의 x좌표와 y좌표를 의미한다.

$$d = \sqrt{x^2 + y^2} \quad (식 2)$$

$$\begin{aligned}
 Radian &= \arcsin\left(\frac{|y|}{\sqrt{x^2 + y^2}}\right) \\
 MV_Angle &= Radian \times 180 \quad (식3)
 \end{aligned}$$

식 3을 이용하여 얻은 움직임벡터의 각도를 그림 2에 나타난 것처럼 좌표평면상에서 움직임 벡터의 x좌표, y좌표가 어떤 사분면에 위치하느냐에 따라 수정함으로써 움직임벡터의 기울기를 구할 수 있다. 계산 방법은 표 1과 같다[4].



(그림 2). 사분면의 위치에 따른 움직임벡터의 기울기

(표 1). 움직임 벡터 좌표의 위치를 고려한 기울기 계산

움직임 벡터의 위치	움직임 벡터의 기울기
1 사분면	Gradient = MV_Angle
2 사분면	Gradient = 180 - MV_Angle
3 사분면	Gradient = 180 + MV_Angle
4 사분면	Gradient = 360 - MV_Angle

이렇게 구해진 기울기 값에 따라 표 2에 나와있는 각도의 범위에 맞추어 8가지 방향 중 하나로 설정함으로써 블록의 움직임 방향을 구할 수 있다.

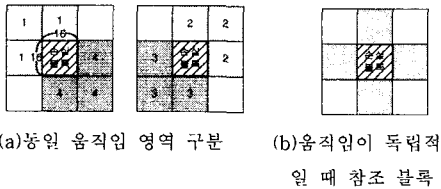
(표 2). 움직임 벡터의 기울기와 블록의 방향과의 관계

움직임 벡터의 기울기	블록의 움직임 방향
Angle = 0°	0
0 < Angle < 22.5 and 337.5 < Angle < 360	1
22.5 < Angle < 67.5	2
67.5 < Angle < 112.5	3
112.5 < Angle < 157.5	4
157.5 < Angle < 202.5	5
202.5 < Angle < 247.5	6
247.5 < Angle < 292.5	7
292.5 < Angle < 337.5	8

3.2 제안 알고리즘

동일 움직임 영역을 찾기 위해 손실블록의 주변블록들을 미리 그림 3의 (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 크기 차이가 임계값 이하일 경우 이 영역을 동일 움직임 영역으로 인식한다.

동일 움직임 영역을 찾은 후 이 영역에 포함된 블록을 참조하여 손실블록의 움직임 벡터를 구하게되는데, 식 4와 같이 참조블록들의 움직임벡터 평균을 구하여 평균과의 차분 값이 최소인 것을 손실블록의 움직임 벡터로 복원한다. 여기에서 MV_i 는 블록들의 움직임 벡터를 나타낸다.



(그림 3). 동일 움직임영역구분과 참조 블록

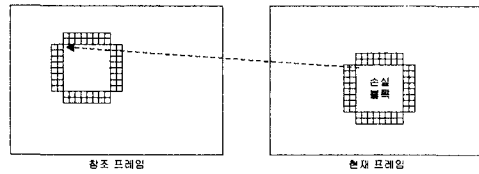
$$Ave = \left(\sum_{i=1}^3 MV_i \right) / 3 \quad (식 4)$$

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

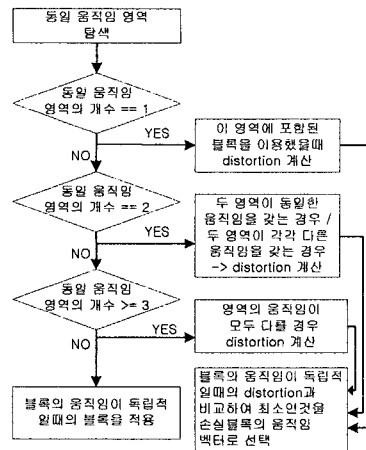
손실블록이 실제 이 움직임영역과 같은 움직임을 갖는지 판단하기 위해 움직임이 독립적일 때와 비교한다. 이때 비교방법은 각각의 방법으로 복원되었을 때 distortion을 이용하게된다. distortion은 그림 4와 같이 손실블록에 인접한 주변블록의 픽셀과 복원된 움직임 벡터를 이용하여 참조 프레임에서 손실블록의 움직임 보상 위치를 찾아 이 블록의 주변픽셀의 값을 비교하여 이 픽셀들의 차분의 합으로 계산한다. 이때 움직임이 독립적일 때의 움직임 벡터는 그림 3의 (b)에서 진하게 색칠된 블록을 참조하여 이 블록들의 중간값으로

복원하게 된다.

이러한 방법으로 동일 움직임 영역을 이용하여 움직임 벡터를 복원했을 때와 블록의 움직임이 독립적인 경우 움직임 벡터를 복원했을 때 각각의 distortion 값을 비교하여 값이 더 작은 쪽을 손실블록의 움직임 벡터로 복원하게 된다.



(그림 4). 제안방법에서 사용되는 주변블록의 픽셀



(그림 5). 제안 알고리즘 순서도

그림 5는 제안 알고리즘의 순서를 간단히 나타낸 것이다. 먼저 동일 움직임이 있는지 조사한 후 만일 동일 움직임 영역이 있는 경우 영역의 개수에 따라 처리 방법이 달라진다.

동일 움직임 영역이 1개인 경우 이 영역에 포함된 블록을 이용하여 움직임 벡터를 복원하고 동일 움직임 영역이 2개인 경우에는 먼저 두 움직임 영역이 같은 움직임을 갖는지 조사한다. 두 영역의 움직임이 같은 경우 나올 수 있는 경우를 그림 6에 나타냈다. 그림 6에서 두 움직임 영역은 실선과 점선으로 구분되며 주변블록들 중 진하게 색칠된 블록들이 손실블록의 움직임 벡터를 복원하기 위해 사용되는 블록들이다. 그러나 그림 6에서 (e)와 (f)의 경우 손실블록을 특정 그룹에 포함시키기 어렵기 때문에 참조하는 블록들이 블록의 움직임이 독립적일 때와 같게된다. 두 영역의 움직임이 같지 않은 경우, 그리고 동일 움직임 영역이 3개 이상이고 각 영역의 움직임이 모두 다른 경우에는 각각의 영역을 이용하여 복원하게 된다.

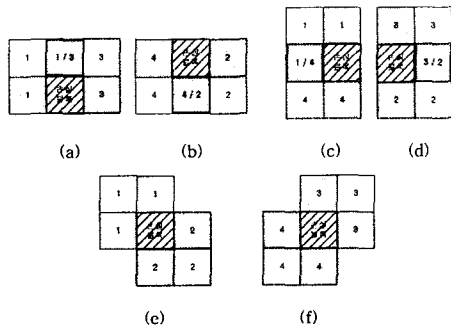


그림 6. 동일 움직임 영역이 2개일 때 참조블록

이렇게 각 움직임 영역을 이용하여 복원했을 때의 distortion과 움직임이 독립적일 때의 distortion을 비교하여 가장 작은 값을 쪽을 손실블록의 움직임벡터로 복원한다.

IV. 실험 결과

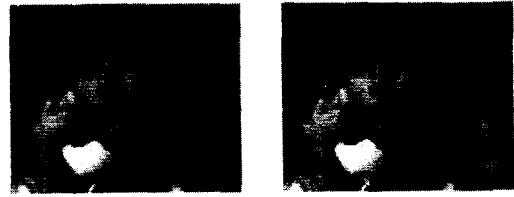
실험은 각 176×144 pixels/lines의 Suzie, Carphone, Mother&daughter 영상에 대해 수행하였다. 표 4는 각 영상의 29번째 프레임에서 에러가 발생했을 때 에러를 닦을 하지 않고 복원한 경우와 BMA와 제안된 기법을 각각 적용하여 복원했을 때 결과 영상의 PSNR을 나타낸다. 그림 7에서는 블록 손실율이 20%일 때 Suzie 영상을 이용하여 실험한 결과를 보인다.

표 3. 176×144 크기 영상의 에러율에 따른 PSNR 비교

영상	Block Loss Rate	no-EC	BMA	제안방법
Suzie	5%	45.70	53.32	54.75
	10%	37.85	42.03	43.83
	15%	37.42	41.24	42.92
	20%	36.92	40.82	42.04
Mother & daughter	5%	41.98	48.67	49.42
	10%	40.08	47.01	48.31
	15%	38.68	45.52	46.42
	20%	37.12	43.79	45.72
Carphone	5%	40.41	42.08	43.87
	10%	35.96	37.54	39.39
	15%	36.39	39.21	39.89
	20%	36.85	39.48	40.32



(a) 에러 영상 (b) no-EC 영상



(c) BMA (d) 제안 방법
그림 7. Suzie(176×144) 영상의 실험 결과 영상

V. 결론

본 논문에서는 전송도중 손상된 움직임 벡터의 복원 기법을 제안하고 실험하였다. 제안알고리즘은 기존의 BMA에 비하여 PSNR과 시각적인 화질면에서 우수한 성능을 보임을 알 수 있었다.

기존의 BMA는 다음 두 가지 문제점을 내포하고 있다. 첫째로 블록 정합 방법이 블록 인접 화소의 상관성을 이용하기 때문에 종종 실제 움직임벡터와 오차가 있는 움직임 벡터를 추정하게 되고 두 번째는 정확한 움직임 벡터를 추정했어도 복원된 영상의 경계에 왜곡 현상이 존재하게 된다.

제안된 방법에서 동일 움직임 영역 추출을 이용하여 움직임 벡터를 복원하고 손실블록의 경계 바깥쪽 픽셀들을 이용하여 움직임 예측과 비슷한 효과를 가져옴으로써 첫 번째 문제점을 어느 정도 보완할 수 있었다. 두 번째 문제점 또한 현재의 블록 화소값과 주변 블록의 움직임 벡터를 고려한 블록 화소값을 가중 평균하여 블록화현상을 줄이는 OBMC(Overlapped Block Motion Compensation)[5]를 이용하여 해결 할 수 있을 것이다.

참고문헌

- [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 coding", IEEE Trans. Commn. vol. 41, pp.1544-1551, 10, 1993
- [4] J.Y. Nam, J.S. Seo, J.S. Kwak, MH Lee, "New block-matching algorithm for motion estimation based on prediction information", SPIE Visual Communication and Image Processing 2000, VOL. 4067, pp.1212-1220
- [5] M.J. Chen, L.G. Chen, and R.M. Weng, "Error concealment of lost motion vectors with overlapped motion compensation", IEEE Transactions on circuits and systems for video technology, Vol 7, No.3, 6, 1997