DBpia

H.264에서 에러은닉을 위한 Optical Flow기반의 움직임벡터 복원 기법

정회원 김 동 형*, 정 제 창**

A Motion Vector Recovery Method based on Optical Flow for Temporal Error Concealment in the H.264 Standard

Donghyung Kim*, Jechang Jeong** ***Regular Members***

요 약

H.264/AVC 부호화 표준은 이전의 여러 부호화 표준에는 없던 새로운 부호화 도구들이 추가되었으며 이를 통해 보다 높은 압축 효율을 가진다. 추가된 여러 부호화 도구들 중 최소 4x4단위의 움직임 추정은 이전보다 인접한 블록들 사이의 움직임벡터 상관도를 높이는 결과를 가져오며, 이러한 특징은 움직임벡터 복원을 통한 에러은닉 기법에 효율적으로 사용될 수 있다. 본 논문은 옵티컬 플로워에 기반하여 움직임 벡터를 복원하는 에러은닉 방법을 제안한다. 인접한 매크로블록들 사이의 움직임벡터들 사이의 상관도가 이전의 여러 부호화 표준들에 비하여 높은 H.264 부호화 표준의 특징을 이용하여 제안하는 알고리듬은 비교적 정확한 초기값의 설정과 속도벡터를 구하기 위한 영역을 16x16크기로 제한함으로써 복잡도를 경감시킨다. 실험결과 제안하는 알고리듬은 다양한 비트율 및 다양한 매크로블록 오류율에서 이전 동일 블록 복사를 수행하는 시방향 복사 방법뿐만 아니라 JM10.1에 구현되어 있는 에러은닉 기법보다 주관적 및 객관적 화질이 높은 것으로 나타난다.

Key Words : Optical Flow, Error Concealment, H.264/AVC Video Coding, Motion Vector Recovery

ABSTRACT

For the improvement of coding efficiency, the H.264 standard uses new coding tools which are not used in previous coding standards. Among new coding tools, motion estimation using smaller block sizes leads to higher correlation between the motion vectors of neighboring blocks. This characteristic of H.264 is useful for the motion vector recovery. In this paper, we propose the motion vector recovery method based on optical flow. Since the proposed method estimates the optical flow velocity vector from more accurate initial value and optical flow region is limited to 16x16 block size, we can alleviate the complexity of computation of optical flow velocity. Simulation results show that our proposed method gives higher objective and subjective video quality than previous methods.

Ⅰ. 서 론

H.264/AVC 표준은 ITU-T의 VCEG(Video Coding Experts Group)과 ISO/IEC의 MPEG(Moving Picture Expert Group)의 공동 작업으로 제정되었다[1]. H.264/AVC 비디오 부호화 방식은 이전의 비디

※ 본 연구는 한국과학재단 목적기초연구(R01-2003-000-11627-0)지원으로 수행되었습니다.
* 한양대학교 전자통신전파공학과 영상처리 및 신호처리 연구실 (kimdh@ece.hanyang.ac.kr)
** 한양대학교 전자전기컴퓨터 공학부 (jjeong@ece.hanyang.ac.kr)
논문번호 : KICS2005-12-487, 접수일자 : 2005년 12월 5일

www.dbpia.co.kr

오 부호화 표준들(MPEG2, MPEG4 Part2, H.263)에는 없던 새로운 부호화 도구들이 추가되었으며, 이러한 도구들은 H.264 부호화 표준이 이전의 비디오 부호화 표준과 비교하여 보다 높은 압축효율을 갖게 하였다. 새롭게 추가된 부호화 도구들에는 4x4 단위의 정수변환 및 인트라 16x16에서의 하다마드 변환, 블록화 현상의 제거를 위한 루프필터(loop filter), 공간영역에서의 인트라 예측부호화, 1/4 화소 단위의 정확도를 가지는 최소 4x4 단위의 움직임 추정 등이 있다[2]. 이렇게 추가된 여러 새로운 부호화 도구들 중 이전의 부호화 표준들에 비해 보다 작은 블록단위에서의 움직임 추정은 인접한 블록들의 움직임벡터간의 상관도를 높이는 결과를 가져오는데, 이러한 H.264의 특징은 손실된 매크로블록의 움직임벡터 복원 시에 유용하게 사용될 수 있다.

MPEG-2가 제정된 이후로 손실된 매크로블록 단위의 움직임 벡터 복원을 통해서 에러를 은닉하는 여러 연구들이 진행되어 왔다. 가장 대표적인 에러은닉방법에는 손실된 블록의 주변화소값을 이용한 BMA(Boundary Matching Algorithm) 기법이 있으며, 이를 수정 보완한 여러 가지 방법들이 제안되었다[3, 4]. 현재 ITU-T에서 권고하고 JM10.1에 구현되어 있는 에러은닉에 사용되는 움직임벡터 복원기법 또한 이러한 BMA 기반의 방법을 사용한다[10]. 하지만 이와 같은 BMA에 기반한 여러 에러은닉 방법들은 적절하지 못한 움직임벡터가 선정된 경우에는 에러를 은닉한 매크로블록과 주변 매크로블록들 사이의 블록화 현상이 발생하여 주관적 화질을 현저히 떨어뜨릴 수도 있다는 단점을 가진다. 이러한 단점을 극복하기 위한 방법으로서 블록단위가 아닌 화소단위의 움직임 벡터의 복원을 통해 에러를 은닉하는 방법이 Zheng에 의해 제안되었으며 이는 주변 매크로블록의 움직임벡터 정보를 이용하여 Lagrange 보간방법을 통한 화소단위의 움직임벡터를 복원해내는 방법이다[5]. 하지만 이러한 에러은닉 기법은 작은 영역에서의 움직임이 발생하였을 경우 높은 성능을 보이지 못한다는 단점을 가진다. 옵티컬 플로워를 이용한 손실된 벡터의 움직임 벡터 복원 방법[6, 7]은 손실된 블록의 인접 화소 단위에서의 옵티컬 플로워 속도 벡터를 구함으로써 보다 이전보다 높은 성능을 나타내지만 높은 복잡도를 갖는다는 단점을 가진다.

본 논문은 옵티컬 플로워를 이용하여 손실된 매크로블록의 움직임 벡터를 4x4단위로 복원하여 에러를 은닉하는 알고리듬을 제안한다. 옵티컬 플로워 기반의 움직임벡터 복원 기법에서의 문제점인 높은 복잡도를 해결하기 위해서 제안하는 알고리듬은 최소 4x4단위에서의 움직임 벡터정보를 포함하는 H.264 부호화 표준의 특성을 이용하여 비교적 정확한 초기값에서 출발함으로써 계산량을 줄임과 동시에 OFR (Optical Flow Region)을 4개의 16x16크기로 제한함으로써 기존보다 복잡도를 크게 경감시킨다.

논문의 구성은 2장에서 옵티컬 플로워의 기본적인 개념을 소개하고 3장에서 옵티컬 플로워 속도벡터를 이용하여 4x4 블록 단위의 움직임 벡터를 복원하는 제안하는 알고리듬을 기술한다. 4장에서는 제안하는 알고리듬과 기존의 H.264에서의 에러은닉 기법과의 비교를 통해서 제안하는 알고리듬의 타당성을 보이고 마지막 절에서 결론을 맺는다.

Ⅱ. 옵티컬 플로워의 개요

2.1 화소값 불변의 제약 조건

옵티컬 플로워(Optical Flow)란 프레임내의 화소들의 위치변화에 대한 속도를 2차원으로 표현한 것을 의미한다[8]. 이는 시간이 지남에 따라 변화하는 화소들은 화소의 위치만 변화될 뿐 그 값은 동일하게 유지된다는 화소값 불변의 제약조건(brightness invariance constraint)에서 출발한다. 이 제약조건은 만일 시간 t에서 (x, y)위치에서의 화소값을 $E(x(t), y(t), t)$라 한다면 식 (1)이 만족됨을 의미한다.

$$E(x(t), y(t), t) = E(x(t+\Delta t), y(t+\Delta t), t+\Delta t) \quad (1)$$

식 (1)의 우변을 Taylor 급수로 전개하면 식 (2)와 같이 정리된다.

$$\left(\frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t}\right)\Delta t + \varepsilon = 0 \quad (2)$$

식 (2)에서 ε은 Taylor 급수의 이차항 이상을 의미하며, 만일 $\Delta t \to 0$이면 식 (2)는 식 (3)과 같이 이원선형방정식(二元線形方程式)으로 간단히 표현된다.

$$E_x u + E_y v + E_t = 0 \quad (3)$$

식 (3)에 포함되어 있는 E_x, E_y 그리고 E_t는 각각 화소값 $E(x(t), y(t), t)$의 x, y 및 t에 대한 일차 편미분 값을 의미하며, $u = \frac{dx}{dt}$와 $v = \frac{dy}{dt}$는 각각 x 와 y 방향으로의 옵티컬 플로워 속도 벡터 즉, 각 화소의 위치 변화에 따른 속도 성분을 나타

www.dbpia.co.kr

낸다. 따라서 만일 두 프레임간의 시간차이를 안다면 옵티컬 플로워 속도 벡터를 구함으로써 다음 프레임에서의 각 화소의 위치를 알 수 있다.

식 (3)을 통해서 화소의 위치 변화에 대한 속도 (u, v)는 x 및 y 방향으로의 화소값의 기울기를 나타내는 기울기 벡터 (E_x, E_y)에 직교하는 직선상에 존재함을 알 수 있으며, 그림 1은 이를 나타낸다.

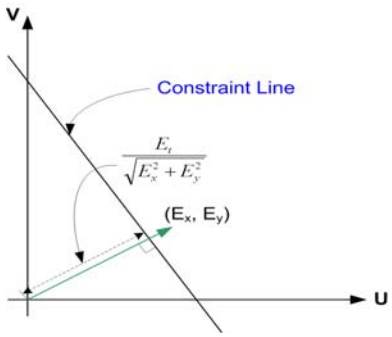


그림 1. 화소값 불변의 제약 조건
Fig. 1. Brightness invariance constraint

2.2 평탄 제약 조건

화소값의 불변에 대한 제약조건만으로는 옵티컬 플로워 속도벡터 (u, v)가 존재할 수 있는 직선의 방정식을 구할 수 있을 뿐 화소의 위치변화에 대한 정확한 (u, v)를 구할 수 없으며, 정확한 속도벡터 (u, v)를 구하기 위해서는 추가적인 제약조건이 필요하다. 추가적인 제약조건으로 주로 사용되는 것이 벡터 (u, v)의 평탄 제약조건(smoothness constraint)이며, 이는 식 (4)와 같이 옵티컬 플로워 속도벡터의 x 및 y 성분에 대한 라플라시안(Laplacian) 값들에 대한 제곱의 합을 최소로 하는 제약 조건을 의미한다.

$$(u,v) = \underset{u,v}{\arg\min}(\nabla^2 u + \nabla^2 v) = \underset{u,v}{\arg\min}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \quad (4)$$

2.3 일차 편미분 및 속도벡터의 라플라시안값의 근사화

일차 편미분의 근사값을 구하기 위한 대표적인 방법으로서 그림 2와 같이 x, y 및 t 방향으로 인접한 여덟 개의 화소값을 이용하여 입방체의 중심점에서의 E_x, E_y 그리고 E_t를 구하는 방법을 들 수 있다. 그림 2에서 j, i, k는 각각 영상좌표에서의 x, y, t에 대응한다. 이렇게 일차 편미분의 근사값을 구함에 있어 단일 화소위치에서의 기울기가 아닌 2x2x2 단위의 입방체에서의 기울기의 평균으로 근사화하는 이유는 잡음의 영향을 최소화하기 위함이다. 따라서, x, y 및 t방향으로의 일차 편미분에 대한 근사값은 식 (5~7)과 같다.

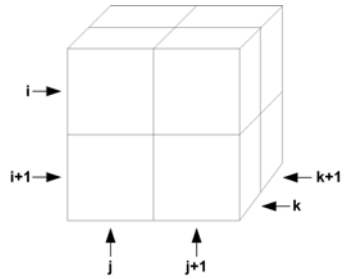


그림 2. 일차 편미분의 근사화
Fig. 2. Approximation of the 1D partial derivatives

$$E_x = \frac{1}{4}\{(E_{i,j+1,k} - E_{i,j,k}) + (E_{i+1,j+1,k} - E_{i+1,j,k}) + E_{i,j+1,k+1} - E_{i,j,k+1}) + (E_{i+1,j+1,k+1} - E_{i+1,j,k+1})\} \quad (5)$$

$$E_y = \frac{1}{4}\{(E_{i+1,j,k} - E_{i,j,k}) + (E_{i+1,j+1,k} - E_{i,j+1,k}) + (E_{i+1,j,k+1} - E_{i,j,k+1}) + (E_{i+1,j+1,k+1} - E_{i,j+1,k+1})\} \quad (6)$$

$$E_t = \frac{1}{4}\{(E_{i,j,k+1} - E_{i,j,k}) + (E_{i+1,j,k+1} - E_{i+1,j,k}) + (E_{i,j+1,k+1} - E_{i,j+1,k}) + (E_{i+1,j+1,k+1} - E_{i+1,j+1,k})\} \quad (7)$$

또한 속도 벡터 (u, v)의 라플라시안 값의 근사화하는 식 (8)과 같으며, $\bar{u}$와 $\bar{v}$는 고려하고 있는 화소위치에 인접한 화소위치에서의 속도벡터들에 대한 가중평균(weighted average)을 의미한다. 그림 3은 이를 적용하였을 때 특정 위치에서의 속도벡터 (u, v)의 라플라시안 값을 구하기 위해 사용되는 주변 화소의 위치와 각 위치에서의 가중치를 나타낸다.

1/12	1/6	1/12
1/6	-1	1/6
1/12	1/6	1/12

그림 3. 속도벡터의 라플라시안 값의 근사화
Fig. 3. Approximation of the Laplacian of flow vector

www.dbpia.co.kr

$$\nabla^2 u \approx (\overline{u_{i,j,k}} - u_{i,j,k}),\ \nabla^2 v \approx (\overline{v_{i,j,k}} - v_{i,j,k}) \quad (8)$$

$$\begin{aligned} &where,\\ &\overline{u_{i,j,k}} = \frac{1}{6}\{u_{i-1,j,k}+u_{i,j+1,k}+u_{i+1,j,k}+u_{i,j-1,k}\}\\ &\quad+\frac{1}{12}\{u_{i-1,j-1,k}+u_{i-1,j+1,k}+\\ &\quad u_{i+1,j+1,k}+u_{i+1,j-1,k}\}\\ &\overline{v_{i,j,k}} = \frac{1}{6}\{v_{i-1,j,k}+v_{i,j+1,k}+v_{i+1,j,k}+v_{i,j-1,k}\}\\ &\quad+\frac{1}{12}\{v_{i-1,j-1,k}+v_{i-1,j+1,k}+v_{i+1,j+1,k}+\\ &\quad v_{i+1,j-1,k}\} \end{aligned}$$

2.4 속도 벡터의 계산

속도 벡터 (u, v)는 두 개의 제약조건에 대한 오차를 최소화하는 (u, v)를 구함으로써 계산할 수 있다. 먼저 첫 번째 제약조건인 화소값 불변 제약조건에서의 오차는 식 (9)와 같다.

$$\varepsilon_b = E_x u + E_y v + E_t \quad (9)$$

또한 두 번째 제약 조건인 평탄제약조건에 대한 오차는 식 (10)과 같다.

$$\varepsilon_c^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \quad (10)$$

따라서 전체 오차는 식 (11)과 같으며 전체 오차를 최소화하는 (u, v)를 선택함으로써 속도벡터를 구할 수 있다.

$$\varepsilon^2 = \iint (\alpha^2 \varepsilon_b^2 + \varepsilon_b^2)\, dx\, dy \quad (11)$$

여기에서 α는 두 제약조건으로부터의 오차에 대한 가중치를 나타낸다. 식 (11)로부터 전체 오차를 최소화하는 속도 벡터 (u, v)는 식(12)를 만족한다[8].

$$\begin{aligned} E_x^2 u + E_x E_y v &= \alpha^2 \nabla^2 u - E_x E_t \\ E_x E_y u + E_y^2 v &= \alpha^2 \nabla^2 v - E_y E_t \end{aligned} \quad (12)$$

최종적으로 식 (12)의 라플라시안 값에 근사값을 나타내는 식 (8)을 대입하여 정리함으로써 식 (13)과 같이 속도벡터 (u, v)를 구할 수 있다.

$$\begin{aligned} u &= (+(\alpha^2 + E_y^2)\overline{u} - E_x E_y \overline{v} - E_x E_t)/(\alpha^2 + E_x^2 + E_y^2) \\ v &= (-E_x E_y \overline{u} + (\alpha^2 + E_x^2)\overline{v} - E_y E_t)/(\alpha^2 + E_x^2 + E_y^2) \end{aligned} \quad (13)$$

Ⅲ. 제안하는 알고리듬

3.1 OFR의 정의 및 속도벡터의 초기값 선정

옵티컬 플로워를 이용한 움직임 벡터 복원을 위해서 제안하는 알고리듬은 먼저 OFR을 정의한다. OFR의 크기는 속도 벡터의 정확성 및 연상량과 직접 관련 되어 있으며, OFR의 크기가 크면 보다 정확한 속도벡터를 구할 수 있으나 연상량이 많아지고, 반대로 작은 크기의 OFR은 상대적으로 적은 연산량을 가지지만 속도벡터의 정확성이 떨어진다. 하지만 만일 속도벡터의 초기값이 비교적 정확하다면 작은 크기의 OFR에서도 충분히 정확한 속도벡터를 구해낼 수 있다.

H.264의 여러 부호화 특성 중 최소 4x4단위의 움직임 추정으로 인하여 각 매크로블록은 이전보다 세밀한 움직임 정보를 포함하며, 이는 속도벡터의 초기값을 이전의 여러 부호화 표준들과 비교하여 보다 정확히 유추할 수 있음을 의미한다. 따라서 제안하는 알고리듬은 이러한 H.264의 부호화 특성을 이용하여 움직임 벡터 복원에 사용되는 OFR을 손실된 매크로블록의 상·하·좌·우에 위치한 16x16 크기로 한정하였으며, 그림 4는 이를 나타낸다.

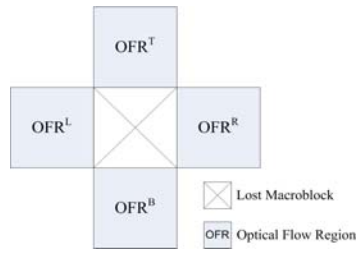


그림 4. 제안한 알고리듬에서 사용한 OFRs
Fig. 4. OFRs used in the proposed algorithm

제안하는 알고리듬은 그림 4에서와 같이 정의된 OFR에서의 속도 벡터 (u, v)를 구하기 위해서 Gaussian Seidel 방법[9]을 사용하여 반복적으로 속도 벡터를 갱신해나가는 방법을 사용하였으며, 이는 식 (14)와 같다.

$$\begin{aligned} u^{n+1} &= \overline{u^n} - E_x[E_x \overline{u^n} + E_y \overline{v^n} + E_t]/(\alpha^2 + E_x^2 + E_y^2) \\ v^{n+1} &= \overline{v^n} - E_y[E_x \overline{u^n} + E_y \overline{v^n} + E_t]/(\alpha^2 + E_x^2 + E_y^2) \end{aligned} \quad (14)$$

이때 각 OFR에서의 속도벡터 (u, v)에 대한 초기값은 손실된 블록에 인접한 4개의 4x4크기의 블록에 대한 움직임 벡터들의 평균값을 사용한다. 속도벡터 (u, v)의 초기값 설정을 위해 사용되는 움직임 벡터는 그림 5에 나타나 있는 것과 같으며 이를 이용한 각 OFR에서의 속도벡터 초기값은 식 (15)와 같다.

www.dbpia.co.kr

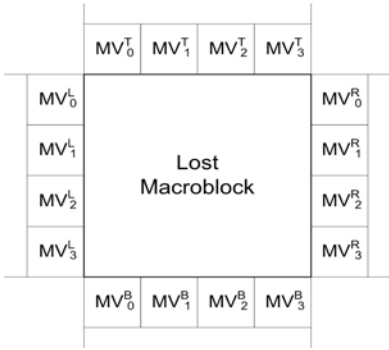


그림 5. 속도벡터의 초기값 설정에 사용되는 주변 움직임 벡터
Fig. 5. Motion vectors used for estimation of the initial value of velocity vector

$$(u,v)^{T}_{initial}=\sum_{i=0}^{3}MV^{T}_{i},\ (u,v)^{B}_{initial}=\sum_{i=0}^{3}MV^{B}_{i}$$
$$(u,v)^{L}_{initial}=\sum_{i=0}^{3}MV^{L}_{i},\ (u,v)^{R}_{initial}=\sum_{i=0}^{3}MV^{R}_{i} \quad (15)$$

3.2 움직임 벡터의 복원

제안한 알고리듬은 손실된 매크로블록의 움직임 벡터를 4x4 크기의 블록 단위로 복원하며, 네 개의 독립된 OFR로부터 계산된 속도 벡터 중 손실된 매크로블록에 바로 인접한 화소 위치, 즉 64개의 화소 위치에서의 속도벡터가 움직임 벡터 복원을 위해 사용되어진다. 또한 손실된 매크로블록의 움직임 벡터는 바깥쪽의 블록에서부터 안쪽블록순서로 복원되며, 그림 6은 이를 나타낸다.

그림 6에서 $(u,v)^{T,B,L,R}_{i}$은 각각 상하좌우에 위치한 OFR으로부터 계산된 (u,v)를 말하며, i는 바깥쪽의 4x4블록에 인접한 네 개의 화소위치에서의 속도벡터에 대한 평균을 의미한다.

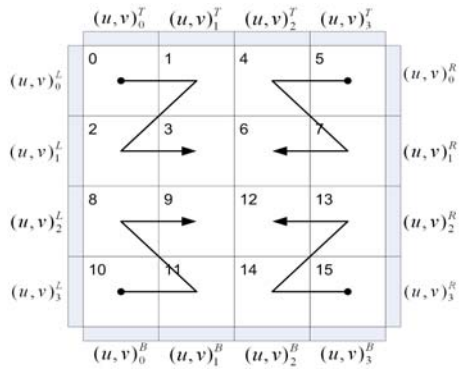


그림 6. 움직임 벡터 복원에 사용되는 속도벡터 및 움직임 벡터 복원 순서
Fig. 6. Velocity vector used for motion vector recovery and the recovery ordering

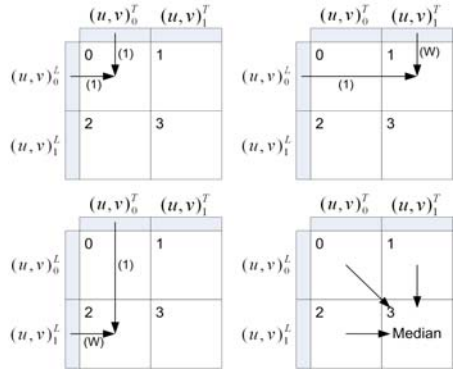


그림 7. 블록 0~블록 3의 움직임 벡터 복원 방법((·) 가중치)
Fig. 7. Motion vector recovery method of block 0 to 3 ((·) weight)

제안하는 움직임벡터 복원기법의 기술(記述)을 위해 먼저 그림 6의 블록0부터 블록3까지의 움직임 벡터 복원 방법을 살펴보면 각 블록은 상측과 좌측의 옵티컬 플로워 속도벡터를 이용하여 복원하게 되며, 이때 거리에 따라 서로 다른 가중치를 부여하는데, 그림 7은 이를 나타낸다.

그림 7에서 보는 바와 같이 블록 0의 경우 상측과 좌측 속도벡터의 가중평균(weighted average)값을 이용하여 움직임벡터를 복원하며 두 지점간의 거리가 같기 때문에 각 속도벡터에 대한 가중치는 1의 값을 갖는다. 블록 1의 경우에는 상측의 속도벡터에 가중치 $W(>1)$를 그리고 좌측의 속도벡터에는 1의 가중치를 주어 평균을 취한다. 블록 2의 경우에도 1의 경우와 유사하며 다만 좌측의 속도벡터에 가중치 W를 부여한다는 점에서만 상이하다. 마지막으로 블록 3의 경우 이미 움직임 벡터가 복원된 블록0~블록2의 움직임벡터에 대한 중간값을 취함으로써 움직임 벡터를 복원한다. 이와 같은 과정을 식으로 표현하면 식(16~19)와 같다.

$$MV_0=\frac{1}{2}((u,v)^{T}_{0}+(u,v)^{L}_{0}) \quad (16)$$

$$MV_1=\frac{1}{1+W}(W\cdot(u,v)^{T}_{1}+(u,v)^{L}_{0}) \quad (17)$$

$$MV_2=\frac{1}{1+W}((u,v)^{T}_{0}+W\cdot(u,v)^{L}_{1}) \quad (18)$$

$$MV_3=Median(MV_0, MV_1, MV_2) \quad (19)$$

나머지 블록들(블록 4~블록 15)에 대해서도 이와 동일한 과정으로 4x4 블록단위로 움직임 벡터를 복원하며, 이미 전술한바와 같이 움직임벡터의 복원은 바

www.dbpia.co.kr

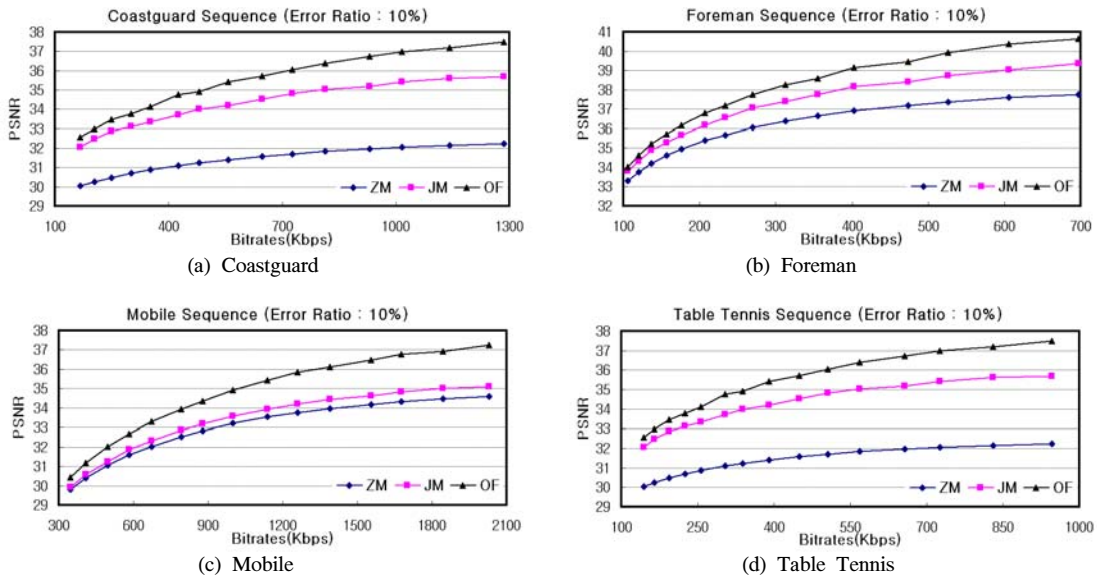
깥쪽의 블록에서부터 안쪽 방향으로 복원해 나아간다.

Ⅳ. 실험 및 결과 분석

제안하는 알고리듬의 타당성을 보이기 위한 실험은 비교적 움직임의 정도가 큰 네 개의 QCIF 크기를 테스트 시퀀스를 대상으로 하였으며, 이는 비교적 움직임이 적은 시퀀스의 경우 이전 프레임에서의 동일 위치 블록으로 에러를 은닉하는 시방향 복사 방법(Zero Motion) 또는 기존의 참조 소프트웨어에서의 방법(Joint Model)만으로도 충분히 좋은 성능을 보이기 때문이다. 실험에 사용된 시퀀스들은 Coastguard, Foreman, Mobile, Table Tennis이며 이들 시퀀스의 최초 100프레임을 IPPP 구조로 부호화 한 결과를 복호기의 입력으로 사용하였다. 제안하는 알고리듬은 H.264 참조 소프트웨어(JM 10.1[10]) 에 적용하여 구현하였으며, 비교를 위하여 시방향 복사 기법과 기존의 H.264 참조 소프트웨어에 구현된 방법을 사용한 결과를 제안하는 방법의 결과와 함께 비교하였다. 또한 옵티컬 플로워 속도 벡터를 구하기 위해 사용된 최대 반복회수는 32회를 사용하였으며, 식 (14) 및 식 (17~18)에서의 α값 및 W 값은 각각 10과 2를 사용하였다.

그림 8은 실험에 사용된 네 개의 시퀀스에 대해서 매크로블록의 오류율(誤謬率)이 10%인 경우 서로 다른 비트율에 따른 객관적 화질을 나타낸다. 실험의 결과에서 볼 수 있는 바와 같이 제안하는 알고리듬은 시방향 복사 방법뿐만 아니라 H.264 참조 소프트웨어에 구현되어 있는 에러은닉 방법과 비교하여 모든 비트율에서 높은 객관적 화질을 나타냄을 알 수 있다.

또한 표 1은 실험에 사용된 시퀀스에 대해서 네 개의 서로 다른 매크로블록 오류율이 발생하였을 때 에러은닉후의 객관적 화질을 비교하고 있다. 그림 8과 표 1의 결과를 통해서 볼 수 있는 바와 같이 제안하는 알고리듬은 다양한 비트율뿐만 아니라 다양한 매크로블록 오류율에서도 이전의 에러은닉 방법들보다 우수한 객관적 화질을 나타냄을 알 수 있다. 또한 4x4단위의 움직임 벡터 복원은 현재 H.264 참조소프트웨어에서의 16x16 단위의 움직임 벡터 복원에 비해서 보다 세밀한 영역에서의 움직임을 고려할 수 있어 보다 높은 주관적 화질을 나타낸다. 제안하는 알고리듬과 현재의 H.264 참조 소프트웨어에서의 방법 그리고 시방향 블록 복사 방법에 대한 주관적 화질 비교를 그림 9~10에 나타내었다. 그림 9~10에서의 흰색 원안의 부분에서 두드러지게 주관적 화질의 차이를 볼 수 있으며 이상의 결과를 통해서 제안하는 알고리듬은 기존의 에러은닉 방법과 비교하여 객관적 화질뿐만 아니라 주관적 화질 면에서도 우수한 성능을 보임을 알 수 있다.



(a) Coastguard (b) Foreman

(c) Mobile (d) Table Tennis

그림 8. 각 시퀀스에서 다양한 비트율에 따른 PSNR의 비교
Fig. 8. Comparison of PSNRs of each sequence according to bitrates.

www.dbpia.co.kr

표 1. 각 테스트 시퀀스에 대한 매크로블록 오류율에 따른 PSNR
Table 1. The PSNRs of each sequence for variable error ratios

(a) Coastguard

QP	PSNR	Error Ratio 5%	10%	15%	20%
20	ZM	36.57	34.59	33.91	32.70
	JM	38.71	36.65	36.37	35.52
	OF	39.46	38.23	37.77	37.16
22	ZM	35.81	34.08	33.49	32.39
	JM	37.58	36.22	35.75	35.01
	OF	38.11	37.19	36.84	36.32
24	ZM	34.81	33.37	32.87	31.93
	JM	36.15	35.10	34.77	34.18
	OF	36.50	35.81	35.55	35.15

(b) Foreman

QP	PSNR	Error Ratio 5%	10%	15%	20%
20	ZM	38.24	36.91	35.54	34.37
	JM	39.25	38.18	37.12	36.04
	OF	**40.03**	**39.14**	**38.34**	**37.42**
22	ZM	37.53	36.38	35.13	34.07
	JM	38.39	37.40	36.55	35.59
	OF	**38.97**	**38.25**	**37.56**	**36.77**
24	ZM	36.64	35.68	34.62	33.66
	JM	37.33	36.56	35.84	35.02
	OF	**37.78**	**37.19**	**36.61**	**35.96**

(c) Mobile

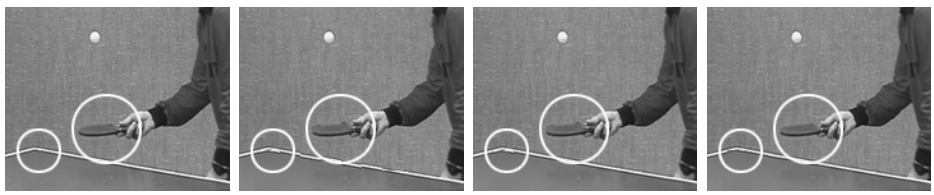
QP	PSNR	Error Ratio 5%	10%	15%	20%
20	ZM	36.11	33.97	33.21	31.99
	JM	36.57	34.44	33.62	32.46
	OF	37.57	36.13	35.09	34.03
22	ZM	35.41	33.54	32.85	31.73
	JM	35.83	33.94	33.19	32.16
	OF	36.61	35.42	34.61	33.64
24	ZM	34.36	32.84	32.27	31.28
	JM	34.67	33.20	32.66	31.74
	OF	35.29	34.35	33.73	32.95

(d) Table Tennis

QP	PSNR	Error Ratio 5%	10%	15%	20%
20	ZM	34.09	31.85	30.76	29.29
	JM	37.23	35.03	34.34	32.98
	OF	**38.34**	**36.39**	**35.72**	**34.57**
22	ZM	33.68	31.58	30.53	29.13
	JM	36.50	34.52	33.96	32.67
	OF	**37.48**	**35.71**	**35.14**	**34.13**
24	ZM	33.17	31.24	30.25	28.93
	JM	35.61	34.00	33.50	32.35
	OF	**36.38**	**34.92**	**34.41**	**33.52**

(a) Original (b) Zero motion (c) JVT Model 10.1 (d) Proposed Method

그림 9. Foreman 영상에 대한 주관적 화질의 비교 (76번째 프레임)
Fig. 9. Comparison of subjective qualities of the foreman sequence (the 76th frame).



(a) Original (b) Zero motion (c) JVT Model 10.1 (d) Proposed Method

그림 10. Table Tennis 영상에 대한 주관적 화질의 비교 (33번째 프레임)
Fig. 10. Comparison of subjective qualities of the table tennis sequence (the 33th frame).

Ⅳ. 결 론

H.264 부호화 표준은 이전의 여러 부호화 표준들과 비교하여 인접한 움직임벡터간의 상관도가 높으며, 이러한 특징은 옵티컬 플로워에 기반한 움직임벡터 복원기법에 효율적으로 사용될 수 있다.

제안하는 알고리듬은 옵티컬 플로워 벡터를 구하기 위해서 Horn이 제안한 화소값 불변의 제약조건 및 평탄제약조건을 이용하였으며, 연산량의 감소를 위해 OFR을 16x16의 크기로 한정하였다. 이후 추정된 옵티컬 플로워 벡터는 손실된 매크로블록내의 4x4블록에 대한 움직임벡터를 복원하는데 사용되며, 각 위치에 따라 서로 다른 가중치를 부여하게 된다. 이러한 4x4 단위의 움직임벡터 복원은 보다 세밀한 부분의 움직임도 추정가능하다는 장점을 가진다.

실험 결과 시방향 블록 복사 방법 및 ITU-T에서

www.dbpia.co.kr

권고하고 현재 H.264 참조 소프트웨어에 구현되어 있는 움직임벡터 복원을 통한 에러은닉 방법과 비교하여 제안하는 알고리듬은 다양한 비트율에서뿐만 아니라 다양한 매크로블록 오류율에서도 보다 높은 객관적 화질을 가지며, 주관적 화질면에서도 이전의 에러은닉방법과 비교하여 높은 성능을 보이는 것으로 나타난다.

참 고 문 헌

〔1〕 JVT G050r1, "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC)," May 2003.

〔2〕 Thomas Wiegand, Gary J. Sullivan, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. Circuits and Systems for Video Technology*, Vol.13, pp.560-576, July, 2003.

〔3〕 Young H. Jung, Yong-goo Kim, and Yoonsik Choe, "Robust error concealment algorithm using iterative weighted boundary matching criterion," *Proc. ICIP*, pp.384-387, 2000.

〔4〕 Ye-Kui Wang, Hannuksela M. Miska, and Viktor Varsa, "The error concealment feature in the H.26L test model," *Proc. ICIP*, Vol.2, pp.729-732, 2002.

〔5〕 Jinghong Zheng and Lap-Pui Chau, "A motion vector recovery algorithm for digital video using lagrange interpolation," *IEEE Trans. on Broadcasting*, Vol.49, pp.383-389, December, 2003.

〔6〕 J. Suh and Y. Ho, "Recovery of motion vectors for error concealment," *IEEE TENCON*, pp.750-753, June, 1999.

〔7〕 Zhi-Heng Zhou and Sheng-Li Xie, "Error concealment based on robust optical flow," *Proc. ICCCAS*, Vol.1, pp.547-550, May, 2005.

〔8〕 B.K.P. Horn and B.G. Schunck, "Determining optical flow," *Artif. Intell.*, 17, 185-203, 1981.

〔9〕 A. Ralston and P. Rabinowitz, *A First Cource in Numerical Analysis*, McGraw-Hill, NewYork, 1978.

〔10〕 JVT Model; *http://bs.hhi.de/~suehring/tml/download/jm101.zip*

〔11〕 정종우, 홍민철, "H.264 표준 동영상 부호화 방식을 위한 순처적 움직임 벡터 오류 은닉 기법," *한국통신학회논문지*, Vol.30, Oct., 2005.

김 동 형 (Donghyung Kim) 정회원

1999년 2월 충북대학교 전자공학과 졸업

2001년 8월 충북대학교 전자공학과 석사

2002년 3월~현재 한양대학교 전자통신전파공학과 박사과정

<관심분야> 영상처리 및 영상압축

정 제 창 (Jechang Jeong) 정회원

1980년 2월 서울대학교 전자공학과 졸업

1982년 2월 KAIST 전기전자공학과 석사

1990년 미국 미시간대학 전기공학과 공학박사

1980년~1986년 KBS 기술연구소 연구원(디지털 TV 및 뉴미디어 연구)

1990년~1991년 미국 미시간대학 전기공학과 연구교수(영상 및 신호처리 연구)

1991년~1995년 삼성전자 멀티미디어 연구소(MPEG, HDTV, 멀티미디어 연구)

1995년~현재 한양대학교 전자전기컴퓨터공학부 교수 (영상통신 및 신호처리 연구실)

1998년 11월 27일 과학기술자상 수상

1998년 12월 31일 정보통신부장관상 표창

<관심분야> 영상처리 및 영상압축

www.dbpia.co.kr