

論文 2002-39SP-6-8

## 전송 오류를 은닉하기 위한 옵티컬 플로우 기반의 움직임 벡터 복원

## (Motion Vector Recovery Based on Optical Flow for Error Concealment)

徐在源\*, 扈堯盛\*

(Jae-Won Suh and Yo-Sung Ho)

## 요약

동영상 압축 비트열은 전송 오류에 매우 민감하다. 만일 전송열 패킷이 손실되거나 손상되어 수신되면, 현재 복호하는 화면 뿐만 아니라, 연속적으로 복호될 화면들에도 화질 저하를 초래한다. 오류은닉 기술은 오류가 없이 수신된 영상 정보를 이용하여 손상된 부분을 은폐하여 화질 저하를 최소화시키는 기법이다. 한가지 방법은 손상된 매크로블록의 움직임 벡터를 추정하고 추정된 움직임 벡터를 이용하여 움직임 보상으로써 손상된 부분을 은닉시키는 방법이 있다. 본 논문에서는 매크로블록의 왜곡 모델을 제시하여 손상된 매크로블록의 추정된 움직임 벡터로 움직임 보상하여 은폐하는 방법이 타당함을 증명한다. 오류은닉의 성능을 향상시키기위해 옵티컬 플로우 기반의 움직임 벡터 추정 방법을 제안하고 다른 움직임 벡터 복원에 의한 오류은닉 방법들과 성능을 비교한다. 또한 제안된 방법은 기존의 방법들보다 계산량이 적은 장점이었다.

## Abstract

The compressed video bitstream is very sensitive to transmission errors. If we lost packet or received with errors during the transmission, not only the current frame will be corrupted, but also errors will propagate to succeeding frames. Error concealment is a data recovery technique that enables the decoder to conceal effects of transmission errors by predicting the lost or corrupted video data from the previously reconstructed error free information. Motion vector recovery and motion compensation with the estimated motion vector is a good approach to conceal the corrupted macroblock data. In this paper, we prove that it is reasonable to use the estimated motion vector to conceal the lost macroblock by providing macroblock distortion models. After we propose a new motion vector recovery algorithm based on optical flow fields, we compare its performance to those of conventional error concealment methods. The proposed algorithm has smaller computational complexity than those of conventional algorithms.

**Keywords** : 오류은닉, MPEG-2 동영상 코덱, 움직임 벡터 복원, 옵티컬 플로우

## I. 서론

디지털 동영상 신호를 대역이 제한된 전송 채널에서

효과적으로 전송하거나 저장 매체에 저장하기 위해 H.261, H.263, MPEG-1, MPEG-2와 같은 동영상 압축 기술을 사용한다. 대표적으로 MPEG-2 동영상 압축 표준은<sup>[1]</sup> 고품질을 유지하면서 수십 배 이상의 압축이 가능하기 때문에, 디지털 방송과 같은 응용분야에 널리 사용되지만 복잡한 압축 방법 때문에 전송 오류에 매우 민감하다. 전송 오류는 전송 채널이 불완전하기 때문에 발생되는데, 수신기에서 복원되는 영상의 화질을

\* 正會員, 光州科學技術院 情報通信工學科

(Dept. of Information and Communications, Kwangju Institute of Science and Technology)

接受日: 2002年7月8日, 수정완료일: 2002年10月15日

저하시킨다. 따라서, MPEG-2 동영상 압축 방법을 이용하는 디지털 TV와 고선명 TV 신호의 충실한 송수신을 위해서 전송 오류에 대한 분석과 화질 향상을 위한 보상 기법에 대한 폭넓은 연구가 필요하다.

동영상 정보를 전송하기 위해서는 몇 단계의 처리 과정을 거친다. 먼저 압축된 동영상 데이터는 전송 환경에 적합하도록 패킷으로 만들어 오디오나 순수 데이터 등과 다중화된다. 다중화된 데이터는 전송 오류를 줄이기 위해 오류정정 부호화 기법 (Forward Error Correction Coding) 등을 이용하여 채널 부호화를 거쳐 전송된다. 이 방법은 전송 오류율이 낮게 제한되어 있을 경우에는 적은 추가 정보만으로도 오류를 완벽하게 제거할 수 있지만, 정정 능력을 벗어난 전송 오류가 발생할 경우에는 오히려 오류를 증가시킬 수도 있다. 또한 부가되는 패리티 정보로 인해 전송해야 할 비트수가 증가하므로 전송 용량을 상대적으로 감소시키는 단점이 있다. 채널 부호화를 마친 동영상 압축 비트열에 남아있는 오류는 동영상 복호기에서 오류은닉 (Error Concealment) 기법을 이용하여 전송 오류의 영향을 최소화한다. 오류은닉 방법은 정상적으로 복원된 시공간적 정보를 이용하여 손상된 부분을 원 영상에 가깝게 복구하는 방법이다. 이 방법은 완벽하게 오류를 정정할 수는 없지만, 예민하지 못한 인간의 시각 특성을 이용하여 오류가 발생한 부분을 어느 정도 복원하는 방법이다. 따라서 부호화 과정에서 오류은닉을 위한 아무런 부가 정보도 전송하지 않고 복호기에서 영상 신호가 갖는 중복성과 연속성만을 이용하기 때문에, 할당된 전송 용량을 영상 정보를 전송하는데 모두 사용할 수 있는 장점이 있다.

오류은닉 기법은 화면내의 중복성을 이용하는 공간적 오류은닉 기법과 화면들 사이의 중복성을 이용하는 시간적 오류은닉 기법으로 분류할 수 있다. 공간적 오류은닉 기법은 이웃 화소들 사이에는 밝기값의 갑작스러운 변화가 없다고 가정하고 보간법을 이용하여 손상된 부분을 복원한다. 손상된 매크로블록을 복원하는 방법은 선형 보간법<sup>[3,4]</sup>이나 방향성을 이용한 보간법<sup>[5]</sup> 등이 있다. 이런 방법들은 오류의 형태가 매크로블록 단위로 고립적일 때에는 우수한 결과를 보이지만 실제로 발생하는 연속된 매크로블록 오류 상황에서는 손상된 매크로블록의 좌우 정보를 충분히 이용하지 못하기 때문에 복원 성능이 그렇게 좋지 못하다. 전송 오류가 발생한 매크로블록의 위치를 고립적으로 만들기 위해서

부호화될 매크로블록의 순서를 위와 아래로 엇갈리면서 부호화하여 전송하는 블록 끼워넣기 방법<sup>[6]</sup>을 사용하기도 한다.

시간적 오류은닉 기법은 연속된 화면간의 유사성을 이용하여 현재 화면에서 손상된 부분을 이전 기준 화면에서 가장 비슷한 부분의 값으로 대치시켜서 복원한다. 가장 간단한 방법은 현재 화면의 손상된 위치의 매크로블록의 움직임 벡터가 0이라고 가정하고, 이미 복호된 기준 화면에서 똑같은 위치의 매크로블록 데이터를 복사하는 것이다. 이 방법은 움직임이 적은 정적인 영역에서 우수한 성능을 보이지만, 움직임이 많은 동적인 영역에서는 좋은 결과를 기대할 수 없다. 이런 문제를 해결하기 위해서 손상된 매크로블록의 움직임 벡터 값을 추정하고 움직임을 보상함으로써 오류를 은닉하는 방법들<sup>[7-9]</sup>이 제안되었다. 이런 방법들은 비교적 우수한 오류은닉 성능을 보이지만 복호기에서 전체 영역 탐색을 기반으로 손상된 매크로블록의 움직임 벡터를 추정하기 때문에 계산량에 대한 부담을 가지고 있다.

본 논문에서는 패킷 기반의 MPEG-2 동영상 비트열의 전송시 발생하는 오류에 대한 은닉 기법을 살펴본다. 특히 정상적으로 수신된 데이터의 시공간적 상관성을 고려한 시간적 오류은닉 기법에 중점을 두었다. 기존의 시간적 오류은닉 방법들의 장단점을 분석하고 복호기의 계산량을 줄이면서 좋은 성능을 얻을 수 있는 옵티컬 플로우 (Optical Flow)를 이용한 움직임 벡터 추정 방법을 제안한다. 본 논문의 구성은 다음과 같다. 먼저 II장에서는 MPEG-2 비디오 압축 알고리즘과 전송 시스템에 대해서 간략하게 살펴보고, 압축 알고리즘의 구조를 이용한 오류 검출 방법을 소개한다. III장에서는 매크로블록의 왜곡이 차분 영상의 양자화 왜곡, 현재 매크로블록에 오류가 없더라도 참고하는 기준 화면에서 오류가 있기 때문에 발생하는 왜곡, 그리고 현재 오류가 발생하여 움직임 벡터 추정 방법에 의하여 오류은닉 방법을 적용하여 발생하는 왜곡 성분이 있음을 보인다. 설정된 매크로블록 왜곡 모델을 이용하여 손상된 매크로블록의 움직임 벡터를 추정하고 움직임 보상하는 것이 오류를 은닉하기 위해 타당한 방법임을 보인다. IV장에서는 기존의 움직임 벡터 복원 방법들의 장단점을 살펴보고, 계산량을 줄이면서 성능은 우수한 옵티컬 플로우 기반의 움직임 벡터 추정 방법을 제안한다. V장에서는 기존에 제안된 방법들과 본 논문에서 제안하는 방법을 S/W로 구현하고 여러 실험 영상을

대상으로 컴퓨터 실험을 통해 각 성능을 비교하였다. 마지막으로 실험 결과를 토대로 오류은닉 기법들에 대한 성능을 평가하고 결론을 맺는다.

II. 시스템 구성과 MPEG-2 비디오 코덱

전송 오류를 고려한 디지털 TV 신호 전송 시스템의 구성도를 <그림 1>에 나타내었다. 각 부호기를 통과한 기초열 (ES: Elementary Stream) 은 전송에 적당한 길이로 패킷화되어 패킷화된 기초열 (PES: Packetized Elementary Stream) 이 된다. 패킷화된 기초열들은 전송열 다중화기 (TS Mux) 에서 전송열 (TS: Transport Stream) 패킷 형태로 만들어지고 전송된다. MPEG-2 표준에서 전송열 패킷의 길이는 4 Byte의 헤더를 포함하여 188 Byte이다. 전송열 패킷 헤더에는 1 비트의 전송오류표시 (Transport-error-indicator) 항목이 있는데, 그 값이 1로 설정되어있으면 전송 오류가 발생했음을 알 수 있다. 비디오 복호기는 오류가 발생한 패킷 데이터를 받았을 때, 패킷 내부의 얼마나 많은 데이터가 정상인지 또한 어디에서 오류가 발생했는지 정확히 모르기 때문에, 오류가 발생한 패킷 데이터는 손실되었다고 취급한다. 따라서 복호기는 다음 재동기가 설정될 때까지 연속적으로 수신되는 데이터의 복호가 불가능하다. MPEG-2 비디오 복호기는 최소 동기 단위가 슬라이스이기 때문에 연속된 매크로블록 데이터 손실 형태의 오류가 발생한다.

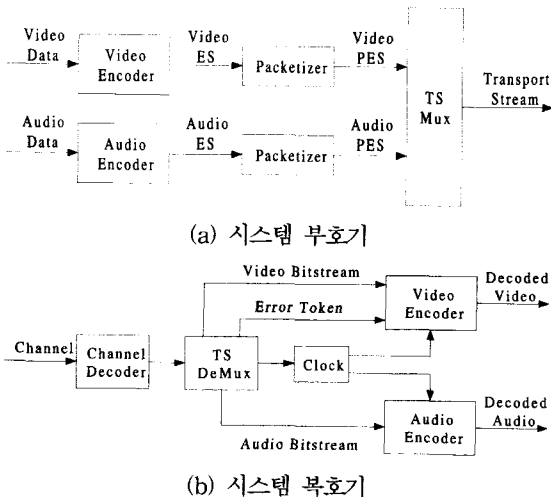


그림 1. 오류가 있는 채널을 고려한 MPEG-2 TS 시스템  
Fig. 1. MPEG-2 TS System for Noisy Channel.

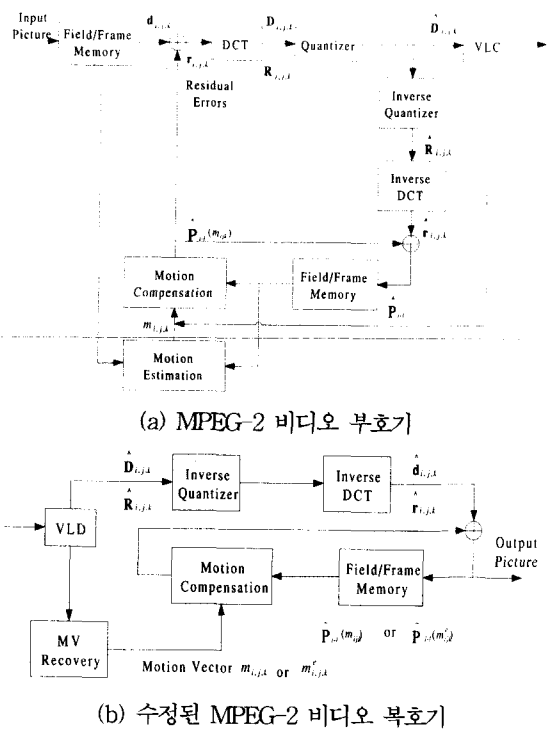


그림 2. 수정된 MPEG-2 비디오 코덱  
Fig. 2. Modified MPEG-2 Video Codec.

<그림 2(a)>처럼, MPEG-2 비디오 압축 방법은 시공간적인 예측 구조를 이용한다. 움직임 보상과 움직임 예측을 사용한 차분 영상의 부호화는 시간적 상관성을 줄이고 이산여현 변환 (Discrete Cosine Transform) 과 양자화를 수행하여 공간적 상관성이 제거한다. 이렇게 시간적, 공간적 상관성이 제거된 데이터를 Huffman 부호화 방법과 같은 가변길이 부호화 (Variable Length Coding) 를 적용하여 통계적 상관성을 제거하여 효율적으로 데이터를 압축한다. <그림 2(b)>는 오류를 은닉하는 방법을 고려한 MPEG-2 비디오 복호기의 구성도이다. 수신된 압축 비트열을 복호할 때, 움직임 벡터 정보가 없다면 움직임을 보상할 수 없기 때문에 화면의 재생이 불가능하다. 하지만 차분 영상에 대한 정보가 없더라도 손상된 매크로블록의 움직임 정보를 알 수 있어서 움직임 보상을 할 수 있다면 매크로블록 경계에서 블록경계 효과와 같은 현상이 나타날 수 있지만 화면 재생이 불가능하도록 치명적이지 않다. 본 논문에서는 이 점을 이용하여 복호기에서 손상된 데이터가 수신되더라도 복호할 매크로블록의 움직임 벡터를 추정하고 움직임 보상을 하여 손상된 화면을 복원하는 방법을 제안한다.

오류는닉과 밀접한 관계가 있는 오류 검출은 매크로블록의 주소값을 이용한다. 각각의 매크로블록은 고유값을 가지며, 이 값들은 화면의 왼쪽 상단의 매크로블록부터 오른쪽 하단의 매크로블록으로 0부터 차례대로 증가한다. 따라서 복호할 때 정상적으로 복호된 매크로블록의 주소값을 저장하였다가, 전송 오류로 인해 복호가 진행되지 못하고 다음의 매크로블록 슬라이스 헤더에서 동기가 재설정된다면, 저장된 매크로블록 주소값의 다음 위치에서 오류가 발생했다고 가정할 수 있다.

### III. 매크로블록 왜곡 모델

본 논문에서 제안하는 왜곡 모델에서는 TS 패킷에 오류가 발생하면 전송중에 손실되었다고 가정한다.  $S_{i,j}$ 를  $i$ -번째 프레임의  $j$ -번째 슬라이스가 왜곡된 경우라고 하면, 해당 슬라이스가 손상될 확률은 식 (1)과 같이 정의될 수 있다.

$$\Pr(S_{i,j}) = \sum_{n=1}^{N_{i,j}} C_n P_b^n (1-p_b)^{n-k} = 1 - (1-p_b)^{N_{i,j}}$$

$$i = 1, \dots, L \quad N \quad 1 \leq j \leq J, \text{ and } J = 1, 2 \quad (1)$$

여기서  $p_b$ 는 비트 오류 확률,  $N_{i,j}$ 는  $i$ -번째 프레임의  $j$ -번째 슬라이스 내의 총 비트 수,  $J$ 는 프레임 내의 슬라이스 개수, 그리고  $N$ 은 프레임의 개수이다.

그림 2에서  $(i, j, k)$ 는  $i$ -번째 프레임의  $j$ -번째 슬라이스의  $k$ -번째 매크로블록을 의미하며 두꺼운 글씨체는 2차원 블록 데이터임을 뜻하고  $i$ 는 부호화 순서이다.  $d_{i,j,k}$ 는 현재 부호화 매크로블록 데이터,  $D_{i,j,k}$ 는  $d_{i,j,k}$ 의 DCT 변환 값,  $\hat{D}_{i,j,k}$ 는  $D_{i,j,k}$ 값이 양자화된 값이고  $\hat{d}_{i,j,k}$ 는 양자화된 데이터가 역 변환 DCT를 통과한 값이다.  $r_{i,j,k}$ 는 차분 영상 데이터이고,  $R_{i,j,k}$ 는 차분값의 DCT 변환 값, 는 DCT 변환 값의 양자화된 값이고,  $m_{i,j,k}$ 는 해당 매크로블록의 움직임 벡터이다.  $w$ 는 움직임 보상을 위한 부호기와 복호기의 화면 저장 공간이다. 복호기에 기준화면에 오류가 발생해서 이미 오류를 은닉했을 수도 있기 때문에 복호기와 부호기의 화면 저장 공간의 표기는 다르다.  $\mathbf{P}_{t-1}(m_{i,j,k})$ 는 움직임 벡터  $m_{i,j,k}$ 를 이용해 움직임 보상된 매크로블록 데이터이다.

인트라 (intra) 매크로블록의 왜곡  $e'_{i,j,k}$ 는 식 (2)처럼 표현될 수 있다. 즉, 복호기에서 손상된 데이터를 받는다면 오류는닉된 데이터  $\tilde{d}_{i,j,k}$ 가  $\hat{d}_{i,j,k}$ 를 대신한다.

$$e'_{i,j,k} = E[(d_{i,j,k} - \hat{d}_{i,j,k})^2](1 - \Pr(S_{i,j})) + E[(d_{i,j,k} - \tilde{d}_{i,j,k})^2]\Pr(S_{i,j}) \quad (2)$$

식 (2)의 우변 첫번째 항은 양자화 오류를 나타낸다. 두번째항은 양자화 왜곡값과 양자화 데이터와 오류는닉된 데이터의 차분값이 상관성이 없다고 가정하면 다음과 같이 분해될 수 있다.

$$E[(d_{i,j,k} - \tilde{d}_{i,j,k})^2] = E[(d_{i,j,k} - \hat{d}_{i,j,k})^2] + E[(\hat{d}_{i,j,k} - \tilde{d}_{i,j,k})^2] \quad (3)$$

식 (3)을 이용하여 식 (2)를 정리하면 식 (4)처럼 정리된다.

$$e'_{i,j,k} \approx E[(d_{i,j,k} - \hat{d}_{i,j,k})^2] + E[(\hat{d}_{i,j,k} - \tilde{d}_{i,j,k})^2]\Pr(S_{i,j}) \quad (4)$$

손실된 데이터를 대신할  $\tilde{d}_{i,j,k}$ 를 구하는 방법은 여러 가지가 있을 수 있는데, I-화면에서는 움직임 벡터 정보가 없기 때문에 손상된 매크로블록의 주변 화소값을 이용하여 공간적 보간법으로 복원할 수 있다. P-화면이나 B-화면에서는 손상된 매크로블록의 주변 매크로블록의 움직임 정보를 이용하거나 화소값들을 이용하여 움직임 예측 방법으로 손상된 매크로블록의 움직임 벡터를 추정하고 움직임 보상하여 왜곡값을 최소화할 수 있다.

인터 (inter) 매크로블록의 왜곡  $e''_{i,j,k}$ 는 식 (5)처럼 표현될 수 있다.

$$e''_{i,j,k} = E\{[(r_{i,j,k} + \hat{\mathbf{P}}_{t-1}(m_{i,j,k})) - (\hat{r}_{i,j,k} + \tilde{\mathbf{P}}_{t-1}(m_{i,j,k}))]^2\}(1 - \Pr(S_{i,j})) + E\{[r_{i,j,k} + \hat{\mathbf{P}}_{t-1}(m_{i,j,k}) - \tilde{\mathbf{P}}_{t-1}(m_{i,j,k})]^2\}\Pr(S_{i,j}) \quad (5)$$

$i$ 는 현재 프레임 번호이다. 오류가 없을 때에는 첫번째 항처럼 차분영상의 양자화 왜곡값과 복호기와 부호기 사이에 차이가 있을 수 있기 때문에 발생하는 왜곡값을 갖는다. 오류가 발생했을 경우에는 두번째 항처럼 복호에 필요한 모든 정보를 모르기 때문에, 해당 매크로블록의 움직임 벡터 정보를 추정하고 기준화면으로부터 움직임 보상하여 왜곡을 최소화할 수 있다.

다. 손상된 매크로블록의 움직임 벡터값을 예측하기 위해서 P-화면에서는  $l$ 은 1 또는  $M$ 이고 B-화면에서는 1과  $M-1$  사이의 값이다.  $M$ 은 I-화면과 P-화면사이나 P-화면들 사이의 화면개수이다. 즉, 인터 매크로블록의 움직임 벡터 정보는 시간적으로 앞이나 뒤의 기준화면 중에서 가까운 기준화면을 이용하여 추정하는 것이다.

위 수식을 좀더 의미있는 항들로 정리하기 위해서 차분 영상의 양자화 왜곡값과 움직임 벡터 정보를 이용한 움직임 보상 정보의 불일치 정도가 상관성이 없다고 가정하면 식 (5)의 첫번째 항은 식 (6)처럼 정리될 수 있다.

$$E[(\hat{\mathbf{P}}_{i-l}(m_{i,j,k}) - \tilde{\mathbf{P}}_{i-l}(m_{i,j,k}))^2](1 - \Pr(S_{i,j})) + E[(\mathbf{r}_{i,j,k} - \hat{\mathbf{r}}_{i,j,k})^2](1 - \Pr(S_{i,j})) \quad (6)$$

또한 차분 영상의 양자화 왜곡값과 오류는은닉으로 인해 발생하는 왜곡이 서로 상관성이 없다고 가정하면 식 (5)의 두번째 항은 식 (7)처럼 분리될 수 있다.

$$E[(\hat{\mathbf{r}}_{i,j,k} + \hat{\mathbf{P}}_{i-l}(m_{i,j,k}) - \tilde{\mathbf{P}}_{i-l}(m'_{i,j,k}))^2]\Pr(S_{i,j}) + E[(\mathbf{r}_{i,j,k} - \hat{\mathbf{r}}_{i,j,k})^2]\Pr(S_{i,j}) \quad (7)$$

식 (6)과 식 (7)을 이용하여 식 (5)를 정리하면 식 (8)처럼 표현할 수 있다.

$$e_{i,j,k}^p \approx E[(\mathbf{r}_{i,j,k} - \hat{\mathbf{r}}_{i,j,k})^2] + E[\hat{\mathbf{P}}_{i-l}(m_{i,j,k}) - \tilde{\mathbf{P}}_{i-l}(m_{i,j,k})]^2 \Pr(S_{i,j}) + E[\hat{\mathbf{r}}_{i,j,k} + \hat{\mathbf{P}}_{i-l}(m_{i,j,k}) - \tilde{\mathbf{P}}_{i-l}(m'_{i,j,k})]^2 \Pr(S_{i,j}) \quad (8)$$

즉, 인터 매크로블록의 왜곡은 차분 영상의 양자화 왜곡, 현재 화면에 오류가 없더라도 기준 화면에 오류가 있었던 부분을 오류는은닉 처리를 했기 때문에 발생할 수 있는 왜곡, 그리고 현재 오류가 발생하여 움직임 벡터 추정 방법에 의해 오류는은닉 기법을 적용하여 발생하는 왜곡 성분이 있음을 알 수 있다. 결국, 현재 오류가 발생한 매크로블록의 움직임 벡터를 정밀하게 추정하면 인터 매크로블록의 왜곡은 줄일 수 있다.

#### IV. 움직임 벡터 추정 방법

시간적 오류는은닉 기법은 연속된 화면간의 유사성을

이용하여 현재 화면에서 손상된 부분을 기준 화면에서 가장 비슷한 부분의 값으로 대체시켜서 복원한다. 따라서 손상된 매크로블록의 움직임 벡터의 추정이 우선되어야 한다.

##### 1. 이웃 매크로블록의 움직임 벡터의 평균을 이용한 방법

우선 손상된 매크로블록 주변 매크로블록의 움직임 벡터 정보를 이용할 수 있다.<sup>[7]</sup> 즉, 화면의 개체들은 대개 같은 방향으로 움직이는 성질을 이용해서 손실된 매크로블록의 수직으로 이웃한 매크로블록의 움직임 벡터의 평균을 이용한다. 만약 상하 매크로블록이 인터로 부호화되어 모두 움직임 벡터를 가지고 있을 경우에는, 그들의 평균이 손실된 매크로블록의 움직임 벡터로 이용된다. 만약 둘 중 하나의 매크로블록만 움직임 벡터를 가졌을 경우에는, 그것을 손실 매크로블록의 움직임 벡터로 간주한다. 불행히도 어느 하나도 움직임 벡터를 갖지 못했을 경우에는, 손실 매크로블록의 움직임 벡터를 0으로 간주한다. 이 방법은 위와 아래의 매크로블록 모두가 움직임 벡터를 가졌을 때 평균값을 적용하였을 경우에만 우수한 성능을 나타내지만, 그렇지 않은 경우에는 매크로블록 경계 부분에서 눈에 거슬리는 현상이 발생한다.

##### 2. 확장 영역을 이용한 움직임 벡터 복원 방법

손상된 매크로블록의 움직임 벡터를 추정하기 위해서 손상된 매크로블록 경계에 맞닿은 화소값들은 손상된 매크로블록과 상관성이 높은 점을 이용한 방법들이 있다.  $E(x, y, t)$ 가  $t$  시간에  $(x, y)$  위치의 화소값을 의미한다.  $E_C(x, y, t)$ 는 움직임 보상 데이터를 의미하며  $E_R(x, y, t)$ 은 차분 영상 데이터를 의미한다.  $m$ 은 움직임 벡터이고  $m_x$ 와  $m_y$ 는 움직임 벡터의  $x$ 와  $y$ 의 성분이다. 기준화면과의 시간적인 차이가 1이라면 복호기에서 재생된 화소값은 식 (9)와같이 표현될 수 있다.

$$E(x, y, t) = E_C(x, y, t) + E_R(x, y, t) = E(x + m_x, y + m_y, t - 1) + E_R(x, y, t) \quad (9)$$

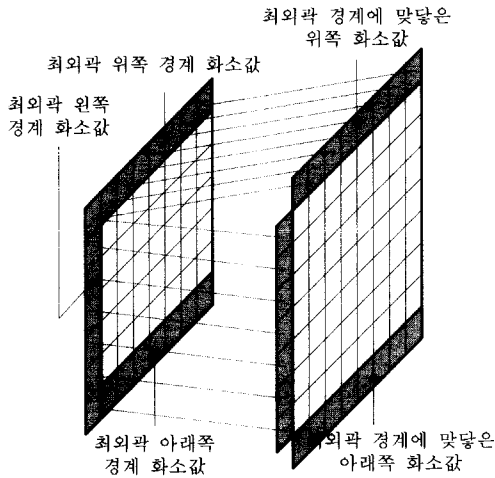
영상 블록의 크기가  $N \times N$ 이고 복호기에서 움직임 벡터 정보와 차분 영상 정보를 알지 못한다면 손상된 매크로블록 데이터는 식 (10)과 같이 추정된 움직임 벡터를 이용하여 예측될 수 있다.

$$\hat{E}(x, y, t) = E(x + m_x^e, y + m_y^e, t - 1) \quad (10)$$

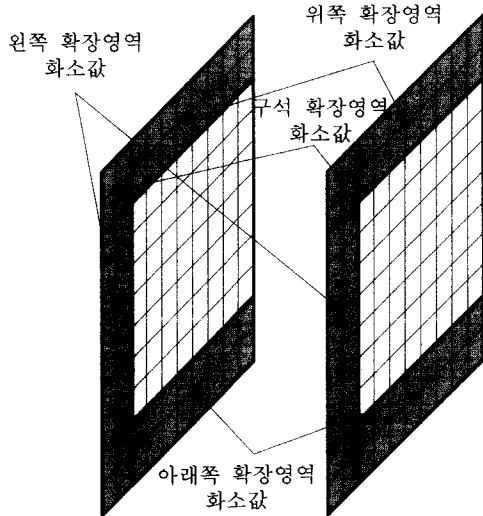
$\text{and } y < x_N \quad 1 \leq y < y_N$

여기서  $(m_x^e, m_y^e)$ 는 복호기에서 추정된 움직임 벡터 정보이고  $(x_1, y_1)$ 은 영상 블록의 가장 왼쪽이면서 위쪽의 좌표이다.

경계 정합 알고리즘 (BMA: Boundary Matching Algorithm)은 <그림 3(a)>처럼 손상된 매크로블록과 맞닿은 정상 복원된 화소값들과 기준 화면에서 블록의



(a) 경계 정합 알고리즘(Boundary Matching Algorithm)



(b) 복호기 움직임 예측 방법(Decoder Motion Vector Estimation)

그림 3. 일반적인 움직임 벡터 복원 방법들  
Fig. 3. Conventional Motion Vector Recovery Methods.

최외곽 화소값들을 이용한다.<sup>[8]</sup> 먼저 식 (11)처럼 제한된 탐색 영역 (SR: Search Range) 내에서 위쪽 ( $B_A$ ), 왼쪽 ( $B_L$ ), 아래쪽( $B_B$ )의 경계부분에서 화소들의 차이값의 제곱의 합을 구한다. 그리고 이 값이 최소값을 갖는 블록 데이터가 손상된 매크로블록 데이터를 대신한다. 이 방법의 중요한 변수는 탐색 영역의 크기의 설정이다. 크면 클수록 손실 매크로블록의 움직임 벡터 후보가 많아져서 좋은 결과를 기대할 수 있지만, 수행 시간이 늘어나는 단점이 있고, 탐색 영역이 너무 커도 오히려 더 나쁜 결과를 초래할 수도 있다. 또한 손상된 매크로블록의 최외곽 경계 화소값들만을 이용하였는데, 움직임 벡터를 추정하기에 충분하지 못한 경우도 있다.

$$B_A = \sum_{x=x_1}^{x_N} (E(x, y_1, t-1) \Big|_{x,y \in SR} - E(x, y_1, t))^2$$

$$B_B = \sum_{x=x_1}^{x_N} (E(x, y_N, t-1) \Big|_{x,y \in SR} - E(x, y_N + 1, t))^2$$

$$B_L = \sum_{y=y_1}^{y_N} (E(x_1, y, t-1) \Big|_{x,y \in SR} - E(x_1 - 1, y, t))^2 \quad (11)$$

움직임 벡터 추정값을 더욱 정밀하게 구하기 위해서 <그림 3(b)>처럼 경계부분에서 사용되는 화소값들을 수를 가변적으로 변화할 수 있도록 한 복호기 움직임 벡터 예측 방법 (DMVE: Decoder Motion Vector Estimation) 이 있다.<sup>[9]</sup> 이 방법 역시 식 (12)와 같이 세 방향에서 화소들의 차이값의 제곱의 합을 구한 값들의 합이 최소값을 갖는 블록 데이터가 손상된 매크로블록 데이터를 대신한다.

$$D_A = \sum_{y=y_1-k}^{y_1-1} \sum_{x=x_1-k}^{x_N} (E(x, y, t-1) \Big|_{x,y \in SR} - E(x, y, t))^2$$

$$D_B = \sum_{y=y_N+1}^{y_N+k} \sum_{x=x_1-k}^{x_N} (E(x, y, t-1) \Big|_{x,y \in SR} - E(x, y, t))^2$$

$$D_L = \sum_{x=x_1-k}^{x_1-1} \sum_{y=y_1}^{y_N} (E(x, y, t-1) \Big|_{x,y \in SR} - E(x, y, t))^2 \quad (12)$$

이 방법에는 탐색영역의 크기뿐만 아니라 움직임 벡터 예측을 위해 사용되는 확장 영역의 폭의 결정이 중요한 변수이다. 확장 영역의 폭은 손실된 매크로블록의 공간적 상관도와 매우 밀접하다. 인접한 화소값들은 손실 매크로블록으로부터 멀어질수록 상관도가 떨어지고 가까울수록 상관도는 높다. 따라서 확장 영역의 폭을 정하는 방법에 따라서 오류는닉 기법의 성능이 달라질

수 있다. 또한 복이 두꺼울수록 계산량이 늘어나는 것이 특징이다.

위에서 설명한 손상된 매크로블록 주변의 확장 영역을 이용한 움직임 벡터 추정 방법들은 우수한 성능을 나타내지만, 복호기에서 블록 정합 알고리즘을 이용한 움직임 예측을 해야 하기 때문에 계산량이 부담이 된다. 또한 경계 정합 알고리즘과 복호기 움직임 예측 방법은 손상된 매크로블록의 왼쪽 화소값들도 이용하는데, 연속된 매크로블록이 손상되는 오류 상황에서는 오류 은닉된 데이터를 다음 손상된 매크로블록의 움직임 예측에 이용하는 것이기 때문에 정확하지 못한 결과를 유도할 수도 있다. 따라서 오류 은닉 성능은 우수하며 복호기의 계산량을 줄일 수 있는 옵티컬 플로우를 이용한 움직임 예측 방법을 제안한다.

### 3. 옵티컬 플로우를 이용한 움직임 벡터 복원 방법

화소의 밝기값은 조명이나 태양처럼 밝게 빛나는 개체에 의해서 변하기도 하지만 고정된 조명원이 있고 개체가 움직여도 변한다. 이런 조명원에 의한 밝기값의 변동을 옵티컬 플로우라고 한다. 본 논문에서는 밝기값의 변화량을 이용하여 손상된 매크로블록의 움직임 벡터를 추정하는 방법을 제안한다. 옵티컬 플로우를 얻는 많은 방법이 있지만, 본 논문에서는 비교적 간단하고 결과가 우수한 Horn과 Schunck가 제안한 방법을 이용한다.<sup>[11,12]</sup>

먼저 각 화소에서 밝기 변화의 움직임과 관련된 제약식을 유도한다.  $u(x, y)$ 와  $v(x, y)$ 는 각 화소 위치에서의 옵티컬 플로우의  $x$ 와  $y$ 방향 성분이다. 여기서 식 (13)처럼 아주 작은 시간 변화  $\delta t$ 와 공간상의 변화  $\delta x$ 와  $\delta y$ 에서의 밝기값은 같다고 가정한다. 여기서  $\delta x = u\delta t$ 이고  $\delta y = v\delta t$ 이다.

$$E(x, y, t) = E(x + \delta x, y + \delta y, t + \delta t) \quad (13)$$

식 (13)을 Taylor 전개와 Chain Rule을 이용하여 전개한 후, 고차항들을 무시하면 식 (14)과 같이 정리할 수 있다

$$\begin{aligned} E_x u + E_y v + E_t &= \\ E_x &= \frac{\partial E}{\partial x}, E_y = \frac{\partial E}{\partial y}, E_t = \frac{\partial E}{\partial t} \end{aligned} \quad (14)$$

$E_x, E_y, E_t$ 는 화소의 밝기값  $E$ 의  $x, y, t$ 에 대한 편미분이다. 식 (14)을 옵티컬 플로우 제약식 (Optical flow

constraint) 이라고 한다. 우리가 알고자 하는 변수는  $u$ 와  $v$ 로 두 개인데, 방정식은 하나이기 때문에 해를 구할 수 없다. Horn과 Schunck는 옵티컬 플로우는 화소 별로 독립적으로 계산될 수 없고, 식 (15)처럼 옵티컬 플로우의 기울기 값을 최소화 시키도록 부드럽게 변해야 한다는 부드러움 제약식 (Smoothness Constraint)을 제안했다.<sup>[11,12]</sup>

$$\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2, \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \quad (15)$$

옵티컬 플로우를 결정하는 것은 옵티컬 플로우 제약식의 전개 과정에서 발생하는 오류를 최소화 시키면서 화면간에 부드럽게 변해야 한다는 부드러움 제약 조건을 만족하도록 하는 이중 적분의 해를 구하는 것이다. 최소값은 적절한 옵티컬 플로우 ( $u, v$ )를 구할 때 얻어진다.

$$e^2 = \iint [(E_x u + E_y v + E_t)^2 + \alpha^2 ((u_x^2 + u_y^2) + (v_x^2 + v_y^2))] dx dy \quad (16)$$

위 식을 변동 계산(Calculus of variation)을 적용하여 전개하면 한 쌍의 Euler 방정식을 푸는 문제로 해석된다. Laplacian 추정을 적용하고 몇 단계의 수학적 풀이를 진행하면 식 (16)은 식 (17)와 같이 정리된다.

$$\begin{aligned} (\alpha^2 + E_x^2 + E_y^2)(u - \bar{u}) &= -E_x [E_x \bar{u} + E_y \bar{v} + E_t] \\ (\alpha^2 + E_x^2 + E_y^2)(v - \bar{v}) &= -E_y [E_x \bar{u} + E_y \bar{v} + E_t] \end{aligned} \quad (17)$$

편미분 값을 아래 식 (18)를 이용하여 구할 수 있다. 편미분 값을 위한 공간과 시간의 관계가 <그림 4>에 있다.

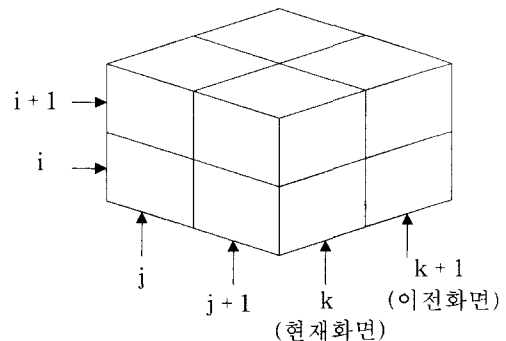


그림 4. 편미분값 예측을 위한 시공간 관계  
Fig. 4. Relationship of Time and Spatial for Estimation of Partial Derivatives.

$$\begin{aligned}
 E_x &= \frac{1}{4} \{ E_{i,j+1,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i+1,j,k} + E_{i,j+1,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i+1,j,k+1} \} \\
 E_y &= \frac{1}{4} \{ E_{i+1,j,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i,j+1,k} + E_{i+1,j,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i,j+1,k+1} \} \\
 E_z &= \frac{1}{4} \{ E_{i,j,k+1} - E_{i,j,k} + E_{i+1,j,k+1} - E_{i+1,j,k} + E_{i,j+1,k+1} - E_{i,j+1,k} + E_{i+1,j+1,k+1} - E_{i+1,j+1,k} \}
 \end{aligned}
 \tag{18}$$

또한 옵티컬 플로우의 국부적 평균값  $\bar{u}$ 와  $\bar{v}$ 는 식 (19)을 이용하여 계산된다.

$$\begin{aligned}
 \bar{u}_{i,j,k} &= \frac{1}{6} \{ u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k} + u_{i,j-1,k} \} + \frac{1}{12} \{ u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i+1,j-1,k} + u_{i+1,j+1,k} \} \\
 \bar{v}_{i,j,k} &= \frac{1}{6} \{ v_{i-1,j,k} + v_{i,j+1,k} + v_{i+1,j,k} + v_{i,j-1,k} \} + \frac{1}{12} \{ v_{i-1,j-1,k} + v_{i-1,j+1,k} + v_{i+1,j-1,k} + v_{i+1,j+1,k} \}
 \end{aligned}
 \tag{19}$$

이제 식 (18)와 식 (19)을 이용하여 식 (17)를 계산하면 된다. 이것을 풀기 위해 Gauss-Jordan 제거 방법과 같은 전형적인 수학적 분석 방법으로 해를 구하는 것은 많은 행과 열을 포함하기 때문에 어렵다. 대안으로 식 (20)처럼 반복적으로 일정 수식을 풀어서 좀더 쉽게 얻을 수 있다.

$$\begin{aligned}
 u^{n+1} &= \bar{u} - E_x [E_x \bar{u}^n + E_y \bar{v}^n + E_z] / (\alpha^2 + E_x^2 + E_y^2) \\
 v^{n+1} &= \bar{v} - E_y [E_x \bar{u}^n + E_y \bar{v}^n + E_z] / (\alpha^2 + E_x^2 + E_y^2)
 \end{aligned}
 \tag{20}$$

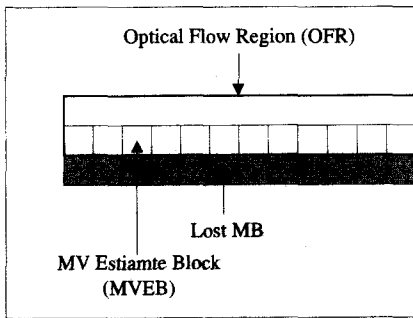


그림 5. 옵티컬 플로우를 이용한 움직임 벡터 예측  
Fig. 5. Motion Vector Estimation Based on Optical Flow.

본 논문에서는 옵티컬 플로우를 이용하여 손상된 매크로블록의 움직임 벡터를 추정하고 움직임 보상함으로써 전송 오류 영향을 최소화시킨다.<sup>[10]</sup> <그림 5>에서 볼 수 있듯이, 먼저 손상된 데이터와 맞닿은 옵티컬 플로우 영역(Optical Flow Region)을 정의한다. 손상된 부분의 위쪽 데이터가 정상적으로 수신되었다면 위쪽 부분을 옵티컬 플로우 영역으로 정의한다. 만약 두개의 연속된 슬라이스 오류가 발생하면 아래쪽 슬라이스의 복원을 위해서는 아래쪽에 정상적으로 수신된 데이터

를 옵티컬 플로우 영역으로 정의한다. 식 (20)을 이용하여 정의된 영역내의 옵티컬 플로우를 계산한다. 그리고 손상된 매크로블록과 맞닿은 매크로블록크기 내부의 옵티컬 플로우의 평균값을 손상된 매크로블록의 움직임 벡터로 결정한다. 옵티컬 플로우는 주변 화소값들과 연동하여 계산되기 때문에 옵티컬 플로우 영역의 크기를 실제 평균값으로 사용되는 크기보다 크게 정의하였다. 움직임 벡터의 예측을 위해 정의된 블록은 손상된 매크로블록과 최대한 근접해야 상관성이 높기 때문에 맞닿은 블록으로 정의하였다.

V. 실험 결과

제안한 오류은닉 방법들의 성능을 비교하기 위해 각 알고리즘을 S/W로 구현하여 여러 실험 영상을 대상으로 컴퓨터 실험을 수행하였다. 이때 사용된 실험 영상은 FOOTBALL, BICYCLE, BALLET, FLOWER GARDEN, TRAIN으로 720×480 크기를 가진다. MPEG-2 비디오 표준은 슬라이스 구조를 매우 유연하게 정의하여 매 화면마다 그 구조를 달리할 수도 있지만, 본 논문의 실험에서는 제한된 구조의 슬라이스 형태를 이용하였기 때문에 한 화면내의 모든 매크로블록이 부호화되었다. 목표 비트율을 5 Mbits/sec로 하였으며, 오류가 발생하여 패킷 손실이 발생하는 화면은 첫 번째 P-화면이다.

표 1. 여러 움직임 벡터 복원 방법의 PSNR 비교

Table 1. PSNR Values of Various MV Recovery Algorithms.

	FOOTBALL	BICYCLE	BALLET	FLOWER GARDEN	TRAIN
Original	32.59	26.57	29.12	26.36	24.88
AVG	30.62	24.55	28.41	24.90	22.95
BMA	31.25	23.37	28.53	25.11	23.39
DMVE	31.23	23.30	28.52	25.12	23.38
OFA	31.75	25.21	28.87	25.36	23.14

본 실험에서는 수직으로 인접한 매크로블록의 움직임 벡터의 평균 (AVG),<sup>[7]</sup> 경계값을 이용한 경계 정합 알고리즘 (BMA),<sup>[8]</sup> 복호기 움직임 벡터 예측 방법 (DMVE),<sup>[9]</sup> 옵티컬 플로우를 이용한 움직임 벡터 추정 알고리즘 (OFA)<sup>[10]</sup> 등의 성능을 비교하였다. <표 1>에



손상된 P 화면을 각 알고리즘에 의해서 오류가 은닉되어 복원된 화면의 PSNR (Peak Signal To Noise Ratio) 값을 나타내었다. BMA, DMVE 방법에서 손상된 매크로블록의 움직임 벡터를 예측하기 위해 [-25, 24] 크기의 탐색 영역을 이용하였다. DMVE는 손상된 매크로블록의 가장 자리의 두 화소 두께의 경계값들을 이용하여 움직임 예측을 하였다. 본 논문에서 제안한 OFA는 유틸리티 플로우를 얻기 위해서 손상된 매크로블록과 맞닿은 위쪽 32 화소 영역을 이용하였고  $\alpha$ 는 1로 하였다. FOOTBALL과 같은 빠른 영상에서는 적지만 PSNR 값의 향상을 볼 수 있고 BICYCLE 같은 동일 방향으로 천천히 움직이는 영상에서는 꽤 높은 향상을 볼 수 있다.

표 2. 복잡도 비교

Table 2. Comparison of Computational Complexity.

	(-) 연산 개수	(+) 연산 개수	(×) 연산 개수
BMA	5050163 =120000	5050153 =112500	5050163 =120000
OFA	321634+324 =6208	32163+32164 +3262=23296	321633+3216312 +3242=3968

<표 2>에 각 알고리즘의 복잡도를 계산하여 비교하였다. BMA 경우는 최소값을 결정하는 정렬에 필요한 계산을 제외하더라도 일정한 탐색 영역 내에서 세 방향의 경계부분에서 화소들의 차이값의 제곱의 합을 구하는데 소요되는 연산 회수를 계산한 결과이다. DMVE는 경계부분에서 차이값을 계산하기 위해 두 화소 두께를 이용하였기 때문에 BMA의 두 배 정도가 필요하다. OFA의 경우는 시공간 편미분 값을 한번 구하고 32 회의 반복과정에서 국부적인 평균값과 임시의 유틸리티 플로우를 계산하는데 소요된 연산 회수이다. 물론 BMA나 DMVE의 경우는 탐색영역의 크기를 줄이면 계산량을 줄일 수 있지만 너무 적게 정의하면 정밀한 움직임 벡터를 추정할 수 없기 때문에 적당한 탐색영역의 크기를 정하는 것이 중요하다.

오류은닉은 발생한 오류가 시각적으로 보이지 않게 숨기는 것이다. 따라서 오류은닉 방법의 성능은 복원된 화면으로 판별하는 것도 중요하다. 따라서 <그림 6>에 BICYCLE을 이용한 각 방법으로 복원된 영상을 나타내었다. 본 논문에서 제안한 OFA을 통해 복원된 결과가 매크로블록의 경계부분에서 가장 부드럽게 복원된



(a) 원래 영상



(b) 손실이 발생한 영상



(c) AVG에 의해 복원된 영상



(d) BMA에 의해 복원된 영상



(e) DMVE에 의해 복원된 영상



(f) 제안된 OFA에 의해 복원된 영상

그림 6. BICYCLE을 이용한 복원된 영상의 화질 비교  
Fig. 6. Subjective Quality Test of Reconstructed Frame with BICYCLE.

것을 알 수 있다. 다른 영상들에서도 제안한 방법으로 우수한 결과를 얻을 수 있었다. 하지만 FOOTBALL과 같이 움직임이 많고 큰 영상에서는 부분적으로 매크로블록의 경계에서 어긋나 보이는 현상이 있다. BALLET 영상은 매크로블록의 크기에 비해서 춤을 추고 있는 사람의 형체가 작기 때문에 옷과 다리 부분에서 시각적으로 거슬리는 부분이 있다. FLOWER GARDEN과 TRAIN은 움직임이 다른 영상에 비해 상대적으로 적은데 각 알고리즘의 성능이 비슷하게 나타났다.

## VI. 결 론

본 논문에서는 MPEG-2 동영상 전송 시스템에서 전송 오류의 영향을 최소화하기 위한 여러 가지 오류는닉 알고리즘을 기술하고 그 성능을 비교하였다. 또한 매크로블록의 오류 모델을 제시함으로써 손상된 매크로블록의 움직임 벡터를 추정하고 보상하는 전송 오류를 은닉하는 방법이 합리적이라는 것을 증명하였다. 기존의 손상된 매크로블록의 경계와 맞는 화소값들을 이용한 움직임 벡터 추정 방법들은 비교적 우수한 성능을 나타내지만 기본적으로 복호기에서 움직임을 예측하기 때문에 계산량이 부담이 된다. 본 논문에서는 계산량을 줄이면서도 성능이 우수한 옵티컬 플로우를 이용한 움직임 벡터를 예측하는 방법을 제안하였다. 이 방법은 손상된 데이터 주변의 정상적으로 수신된 데이터를 이용하여 정해진 회수의 반복만으로 옵티컬 플로우를 얻고, 얻어진 옵티컬 플로우의 평균 값을 손상된 매크로블록의 움직임 벡터로 이용한 결과 상당히 우수한 결과를 얻을 수 있다. 여러가지 실험 영상을 이용한 시각적 비교와 PSNR 값 비교로부터, 본 논문에서 제

안한 옵티컬 플로우를 이용하여 움직임 벡터를 추정하고 움직임 보상을 하는 방법이 우수한 성능을 보임을 확인하였다.

## 참 고 문 헌

- [1] ISO/IEC IS 13818-2 (MPEG-2 Video): Information Technology- Generic Coding of Moving Pictures and Associated Audio Information, April 1996.
- [2] Y. Wang, S. Wenger, J. Wen, and A.K. Katsaggelos, "Error Resilient Video Coding Techniques," *IEEE Signal Processing Magazine*, Vol. 17, No. 4, pp. 61~82, July 2000.
- [3] M. Ghanbari and V. Seferidis, "Cell-loss Concealment in ATM Video Codecs," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 3, No. 3, pp. 238~247, June 1993.
- [4] S. Aign and K. Fazel, "Temporal and Spatial Error Concealment Techniques for Hierarchical MPEG-2 Video Codec," *Proc. ICC*, Vol. 3, pp. 1778~1783, June 1995.
- [5] J.W. Suh and Y.S. Ho, "Error Concealment Based on Directional Interpolation," *IEEE Transactions on Consumer Electronics*, Vol. 43, No. 3, pp. 295~302, Aug. 1997.
- [6] Q. Zhu, Y. Wang and L. Shaw, "Coding and Cell-Loss Recovery in DCT-Based Packet Video," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 3, No. 3, pp. 248~258, June 1993.
- [7] H. Sun, K. Challapali and J. Zdepski, "Error Concealment in Digital Simulcast AD-HDTV Decoder," *IEEE Transactions on Consumer Electronics*, Vol. 38, No. 3, pp. 108~116, Aug. 1992.
- [8] W.M. Lam, A.R. Reilbman, and B. Liu, "Recovery of Lost or Erroneously Received Motion Vectors," *Proc. ICASSP*, pp. V417~V420, Apr. 1993.

- [9] J. Zhang, J.F. Arnold, and M.R. Frater, "A Cell-Loss Concealment Technique for MPEG-2 Coded Video," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, No. 4, pp. 659~665, June 2000.
- [10] J.W. Suh and Y.S. Ho, "Motion Vector Recovery Using Optical Flow," *Proc. ICCE*, pp. 234~235, 2000.
- [11] B. Horn, *Robot Vision*, The MIT Press, 1986.
- [12] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, Vol. 17, pp. 185~203, 1981.

---

저 자 소 개

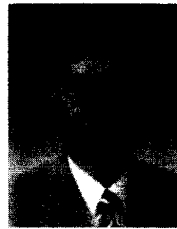
---



徐在源(正會員)

1995년 2월 충북대학교 전자공학과 졸업 (학사), 1997년 2월 광주과학기술원 정보통신공학과 졸업 (석사), 1997년 3월~현재 : 광주과학기술원 정보통신공학과 전공 (박사과정) <주관심분야 : 영상 부호화, 멀티미디어 영상통신, 디지털 비디오 오류 제어>

어 영상통신, 디지털 비디오 오류 제어>



扈堯盛(正會員)

1981년 2월 서울대학교 전자공학과 졸업 (학사), 1983년 2월 서울대학교 전자공학과 졸업 (석사), 1983년 3월~1995년 9월 : 한국전자통신연구소 선임연구원, 1989년 12월 미국 University of California, Santa

Barbara Department of Electrical and Computer Engineering (박사), 1990년 1월~1993년 5월 : 미국 Philips 연구소 Senior Research Member, 1995년 9월~현재 : 광주과학기술원 정보통신공학과, 부교수 <주관심분야 : 디지털 신호처리, 영상신호처리 및 압축, 초저속 영상통신, 디지털TV와 고선명 TV 방식, 멀티미디어 통신>