

움직임벡터의 거리와 방향성을 고려한 H.264 에러 은닉 방법

정희원 손남례*, 이귀상**

Error Concealment Method considering Distance and Direction of Motion Vectors in H.264

Nam-rye Son*, Guee-sang Lee** *Regular Members*

요약

본 논문은 무선망과 같이 패킷 손실이 많은 환경에 H.264 부호화 영상이 전송될 때 복호기 단말기에서 손실된 움직임 벡터를 효율적으로 에러은닉 방법에 대하여 2가지를 제안한다. 첫째, 손실된 블록(매크로블록)에 대하여 후보벡터집합(candidate vector set)을 선정하는 방법으로는 손실된 블록에 인접한 주변 블록의 움직임벡터들의 높은 상관성을 착안하여 후보벡터를 선정한다. 이때 제안한 알고리즘은 주변 블록의 움직임벡터들 간에 거리를 이용하여 클러스터한다. 클러스터된 움직임 벡터 집합(클러스터 집합)에서 최적의 후보벡터 선택방법은 중앙값을 선택한다. 둘째, 손실된 블록의 최종의 후보벡터를 선정하는 방법으로는 후보벡터집합에서 주변에 인접한 픽셀간의 방향성을 고려하여 왜곡 값이 최소인 벡터를 후보벡터로 결정한다. 패킷이 손실되는 환경에서 실험한 결과, 제안한 에러 은닉 방법은 기존 방법에 비하여 후보벡터 개수가 평균적으로 23%~61%까지 감소하였고, 디코딩 시간이 평균적으로 3~4(sec) 감소하였다. 또한 화질에 대한 객관적 평가 기준인 PSNR은 평균적으로 기존 방법들과 거의 비슷하였다.

Key Words : H.264, error concealment, motion vector, cluster, FMO, Interleave

ABSTRACT

When H.264 encoded video streams are transmitted over wireless network, packet loss is unavoidable. Responding on this environment, we propose methods to recover missed motion vector in the decoder: At first, A candidate vector set for missing macroblock is estimated from high correlation coefficient of neighboring motion vectors and missing block vectors the algorithm clusters candidate vectors through distances amongst motion vectors of neighboring blocks. Then the optimal candidate vector is determined by the median value of the clustered motion vector set. In next stage, from the candidate vector set, the final candidate vector of missing block is determined it has minimum distortion value considering directions of neighboring pixels' boundary. Test results showed that the proposed algorithm decreases the candidate motion vectors 23~61% and reduces 3~4sec on average processing(decoding) time comparing the existing H.264 codec. The PSNR, in terms of visual quality is similar to existing methods.

※ 본 연구는 2007년 정부의 재원(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2007-531-D00013)

* 호남대학교 인터넷소프트웨어학과(nrson@honam.ac.kr), ** 전남대학교 컴퓨터정보학부(gslee@chonnam.ac.kr)

논문번호: KICS2008-06-271, 접수일자: 2008년 6월 27일, 최종논문접수일자: 2008년 11월 10일

I. 서 론

최근 인터넷과 이동 통신의 급속한 성장과 영상 통신 서비스의 보편화에 따라 화상회의, 화상전화, 디지털 TV와 같은 다양한 멀티미디어 서비스가 대중화되어 가면서 신뢰성 있는 실시간 멀티미디어 서비스에 대한 관심이 증가되고 있다. 이에 따라 ITU-T와 국제 표준화 기구에서 유무선 통신망 환경에서 동영상 서비스를 위한 표준 규격인 H.264를 제정하였다^{[1][2]}. H.264는 기존 영상 압축 표준(MPEG-2, MPEG-4, H.263)보다 다양하고 새로운 압축 기법들이 도입되었지만, 기본적으로 공간적 중복성을 줄이기 위해 블록 기반 이산 여현 변환(DCT: discrete cosine transform)과 양자화 기법을 사용하며, 시간적 중복성을 줄이기 위해 움직임 보상(MC: motion compensation) 기법을 사용한다. 또한 부호화 효율을 높이기 위해 예측 부호화(prediction coding) 기법 등을 이용해 상호 연관성을 제거함으로써 비디오 데이터를 고밀도로 압축한다. 대부분의 영상 압축 표준들은 가변장 부호화 기법과 예측 부호화 기법을 사용하여 상호 연관성을 제거함으로써 높은 압축 효율을 얻고 있다. 이 두 기법은 이동 통신 환경과 같은 에러에 노출된 채널 환경에서 에러가 발생한 경우 영상의 화질에 큰 영향을 끼친다. 즉, 가변장 부호화 기법은 에러가 있는 채널 환경에서 매우 취약하여 에러가 발생할 경우 복호시 동기화된 정보를 잃어버리게 되는 원인을 제공하고, 예측 부호화 기법은 전체 영상을 통해 에러가 전파되도록 만들기 때문에 전송 채널 환경에서의 에러는 복원된 영상의 품질을 저하시키고 심지어 전체 영상 통신 시스템의 서비스 단절을 초래하기도 한다.

이런 문제점을 대응하기 위해서 무선망에서 데이터 전송시 발생하는 에러에 대한 관심이 고조되고 있다. 데이터 전송시 발생할 수 있는 무선망의 에러 유형으로는 무선망에서 채널 전송시 다경로 반사(multi-path reflection)나 페이딩(fading)과 같은 현상에 의한 단일 비트 에러나 군집 에러와 같은 비트 오류와 무선망에서의 대표적인 패킷 손실 에러가 있으며, 이러한 전송 중 발생하는 에러 상황에 대해 에러가 발생할 환경 하에서 압축된 동영상 데이터의 전송을 위해 다양한 에러 복원 기법들이 개발되어 왔다.

일반적으로 이들 에러 복원 기법들은 에러 검출(error detection), 재동기화(resynchronization), 데이

터 복원(data recovery), 에러 은닉(error concealment) 방법으로 분류된다. 에러 검출 기법은 예측할 수 없는 장애 요인에 의해서 발생하는 에러로 다른 내용이 송신될 수 있다. 이러한 에러를 줄이기 위해 에러가 발생하는 경우 정확한 에러 정보가 어떤 것인가를 수신한 비트열로부터 유추할 수 있는 기법으로는 전방향 에러 수정(FEC: forward error correction) 방식과 후진 에러 수정방식(BEC: backward error correction), 그리고 자동 재전송 요구(ARQ: automatic repeat on request) 방식이 있다.

에러 은닉 방법은 디코더에서 인코더로부터의 추가적인 정보에 의존하지 않고 이전에 수신된 에러가 없는 영상 정보를 이용하여 손실되거나 에러가 있는 영상 데이터를 적절히 복구하는 순수 디코더 기반의 방법이다. 많은 에러 강화 비디오 코딩 중, 에러 은닉 방법은 독립적이면서 서로 다른 강인한(robust) 비디오 코딩 방법과 함께 사용되므로 에러 내성(error resiliency) 측면에서 가장 효율적인 방법 중 하나로 고려되고 있다.

본 논문의 구성은 다음과 같다. II장에서는 H.264 부호화특징에 대해 살펴보고, III장에서는 H.264 표준에 적용중인 에러 은닉 방법들에 대하여 설명한다. IV장에서는 본 논문에서 제안한 손실 블록에 인접한 주변블록의 움직임 벡터들을 이용하여 손실블록의 움직임벡터를 결정하는 방법에 대하여 설명한다. V장에서는 무선 환경과 같이 패킷이 손실된다는 가정 하에서 기존 에러 은닉 방법과 제안한 에러 은닉 방법에 대하여 비교, 분석한다. 마지막으로 VI장에서는 본 논문의 결론을 맺는다.

II. H.264 부호화 특징

2.1 H.264 에러 강인화 기법

에러 강인화(error resilience) 기법은 에러에 의한 영향이 확산될 원인을 미리 제거함으로써 에러에 의한 손실을 최소화하는 기법으로 패킷 교환망을 주 응용 분야로 삼고 있다. H.264 표준은 기존의 에러 강화 기법을 패킷망에 적용할 수 있도록 변형한 인트라 리프레쉬(Intra refresh), 파라미터 집합(parameter set), 데이터 분할(data partition), 여분의 슬라이스(redundant slice), FMO(flexible macroblock ordering)가 있다.

인트라 리프레쉬는 인트라 프레임 내부에 있는 몇 개의 매크로 블록을 강제적으로 인트라부호화 하는 방식으로 움직임이 많은 매크로 블록인 경우 선택

적으로 부호화하면 화질 열화 전파가 방지된다. 파라미터 집합은 프레임들이나 슬라이스를 디코딩하기 위해 필요한 중요한 정보를 신뢰적인 전송 프로토콜을 이용하여 보호할 수 있다. 데이터 분할은 정보의 중요도에 따라 A, B, C로 구분하여 전송하는 방식으로 매크로블록의 헤더, 슬라이스 헤더, 매크로블록 타입, 양자화 계수 등 디코딩하는데 가장 중요한 정보를 A구역, 인트라 프레임으로 코딩한 데이터는 B구역, 인터 프레임으로 코딩한 데이터를 C 구역으로 슬라이스 당 총 3구역으로 나누어 3개의 네트워크 친화 계층 구조로 코딩하여 각각 따로 전송하여 패킷 손실에 대응하는 방식이다. 여분의 슬라이스 기법은 패킷 손실이 많이 발생하는 환경에서의 전송을 위해 코딩되는 슬라이스의 매크로블록에 여분의 매크로블록을 인코딩하는 것으로 패킷 손실로 인해 원래의 슬라이스가 없어졌을 경우 여분의 슬라이스를 가지고 디코딩 한다. FMO는 슬라이스 그룹 개념을 도입해 매크로블록마다 슬라이스 그룹을 지정하여 슬라이스에 매크로블록을 유연하게 할당하는 방식이다.

2.2 H.264 슬라이스 그룹

H.264 부호기의 가장 큰 특징 중 하나는 하나의 프레임을 여러 개의 슬라이스로 나누어서 부호화할 수 있다. 하나의 슬라이스는 매크로블록의 집합으로 구성되며 슬라이스를 구성하는 형식에 따라 슬라이스 그룹이 형성된다. H.264 부호기에서 제공하는 슬라이스 그룹은 기본적으로 6가지(interleave, dispersed, foreground with left-over, box-out clockwise and counter-clockwise, raster scan and reverse scan, wipe right and wipe left) 형식으로 구성할 수 있다. 슬라이스 그룹 단위로 부호화함으로써 발생하는 장점은 에러가 발생하더라도 현재 에러 발생 슬라이스 그룹에만 영향을 미치며 서로 다른 슬라이스 그룹에 대해서는 영향을 미치지 않는다.

본 논문에서는 H.264 부호기에서 가장 많이 사용하는 인터리브 및 디스펠스 방법을 이용하여 부호화하여 실험한다. 인터리브 방법은 그림 1(a)과 같이 기존 H.263+에서 제공하는 방식과 마찬가지로 한 프레임 내에 존재하는 매크로블록들을 GOB 단위로 한 묶음 하여 부호화하는 방식이다. 디스펠스 방법은 그림 1(b)과 같이 그룹들을 흩어지게 할당하는 방식으로 슬라이스 그룹을 두 개로 해서 체크보드 타입으로 코딩이 이루어지면 어느 한 그룹에 속

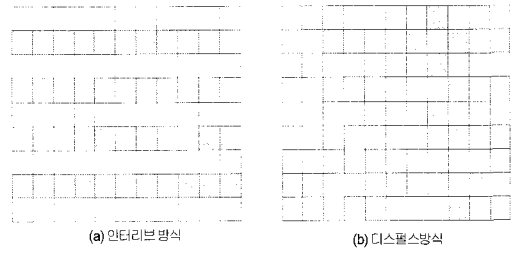


그림 1. 인터리브 및 디스펠스 방식

하는 패킷이 손실되었을 경우 다른 그룹에 속하는 패킷의 정보를 이용하여 효과적으로 에러 은닉을 적용시킬 수 있다⁴⁾.

III. 기존 에러은닉방법

3.1 H.264의 인트라 프레임을 위한 에러은닉 방법

인트라 프레임 경우에는 이전 프레임에 대한 정보가 없으므로 현재 프레임에 먼저 인코딩된 매크로블록들을 기반으로 복호화하기 때문에 2차원 공간적(spatially)으로 에러 은닉을 수행한다. 그림 2에서 보여주듯이 픽셀간의 거리에 반비례하게 가중치를 계산해 평균을 내는 가중 픽셀 평균 방식(weighted pixel averaging)을 기반으로 수행한다³⁾.

가중픽셀평균방식은 식(3)을 이용하여 손실된 블록을 은닉한다. 이때 F 는 현재프레임에 대한 위쪽, 왼쪽, 아래쪽, 오른쪽에 대한 픽셀의 좌표이다. $F_{\hat{s}}(m+i, n+j)$ 는 손실블록에 대한 픽셀의 좌표이다. 이때 손실블록의 수직좌표를 m , 수평좌표를 n , 그리고 손실블록을 구성하는 픽셀의 수직 좌표를 i ($0 \leq i \leq 15$), 수평좌표를 j ($0 \leq j \leq 15$)라고 한다.

식(3)에서 상(WMB_T), 하(WMB_B), 좌(WMB_L), 우

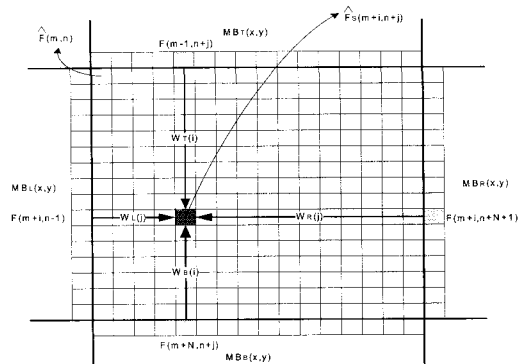


그림 2. 인트라 프레임을 위한 공간적 보간법

(WMB_R)는 식(1)과 식(2)에서 픽셀간의 거리에 반비례하게 가중치를 적용한 결과이다.

$$\begin{aligned} W_T(i) &= i \\ W_B(i) &= 15-i \\ W_L(j) &= j \\ W_R(j) &= 15-j \end{aligned} \quad (1)$$

$$\begin{aligned} WMB_T &= W_T(i) \times F(m+N, n+j) \\ WMB_B &= W_B(i) \times F(m-1, n+j) \\ WMB_L &= W_L(j) \times F(m+i, n+N+1) \\ WMB_R &= W_R(j) \times F(m+i, n-1) \end{aligned} \quad (2)$$

$$\hat{F}_s(m+i, n+j) = \frac{WMB_T + WMB_B + WMB_L + WMB_R}{W_T(i) + W_B(i) + W_L(j) + W_R(j)} \quad (3)$$

3.2 H.264의 인터 프레임에 위한 에러은닉 방법

인터 프레임은 이전의 프레임을 참조하여 부호화하므로 2차원 공간적으로 이웃하거나 시간 방향으로 이웃하는 매크로블록들의 움직임 정보를 보고 없어진 매크로블록들의 움직임 벡터를 추측하여 에러 은닉을 한다. 정확히 받은 슬라이스들의 평균 움직임 벡터가 어느 일정 값보다 작으면 기준 프레임의 해당하는 위치에 있는 값으로 대체되고, 크거나 같으면 경계정합알고리즘(BMA: boundary matching algorithm)을 기반으로 이루어진다.

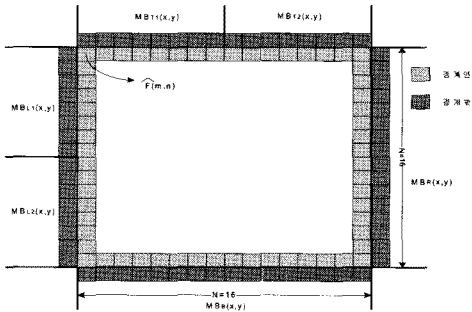


그림 3. 경계정합알고리즘(BMA)

기존 동영상 코덱인 H.263에서 사용하고 있는 경계정합알고리즘에서 사용하고 있는 후보벡터집합으로는 다음과 같다.

- a) 손실블록과 같은 위치에 있는 이전 프레임 블록의 움직임 벡터
- b) 손실블록에 인접한 블록의 움직임벡터에 대한 평균값(average MV)

- c) 손실블록에 인접한 블록의 움직임벡터에 대한 중앙값(median MV)
- d) 손실블록에 인접한 블록의 움직임 벡터 값
- e) (0,0) 움직임 벡터(zero vector)

그러나 H.264 참조 소프트웨어에서 제공하고 있는 인터프레임에 대한 에러은닉방법은 기존 동영상 코덱에 대한 부호화 방법의 차이로 인하여 후보벡터를 선정하는 방법이 다르다. 현 H.264 코덱은 가변블록을 사용하고 있으므로 다양한 움직임 벡터를 가지고 있다. 따라서 움직임 벡터의 최소 단위는 8x8단위로 인접한 매크로블록들이 4x8, 8x4, 4x4블록으로 나누어졌을 경우는 평균을 구해 8x8단위의 움직임 벡터를 이용한다. 경계정합알고리즘에서 사용하고 있는 후보벡터집합은 손실블록에 인접한 블록의 위, 왼쪽, 아래, 오른쪽 블록의 움직임 벡터를 후보벡터로 선정한다. 즉, 그림 (3)에서의 후보벡터 집합은 손실블록에 인접한 움직임 벡터(6개)와 zero 움직임벡터(0,0)로 총 7개로 구성되어있다.

이때 블록경계의 유사성 판단은 SMD(side match distortion)를 이용한다. 즉, 식(4)과 같이 후보벡터들을 이용하여 움직임이 보상된 블록의 경계 안 화소값과 이웃 블록의 경계 밖 화소값의 차이에 의해서 SMD를 구하여 최소의 SMD를 갖는 후보벡터를 최종적으로 손실블록의 움직임벡터로 하여 움직임 보상을 한다.

$$\begin{aligned} d_T &= \sum_{i=0}^{N-1} (\hat{F}(m, n+i) - F(m-1, n+i))^2 \\ d_L &= \sum_{i=0}^{N-1} (\hat{F}(m+i, n) - F(m+i, n-1))^2 \\ d_B &= \sum_{i=0}^{N-1} (\hat{F}(m+N-1, n+i) - F(m+N, n+i))^2 \\ d_R &= \sum_{i=0}^{N-1} (\hat{F}(m+i, n+N) - F(m+i, n+N+1))^2 \\ d_S &= d_T + d_L + d_B + d_R \end{aligned} \quad (4)$$

식(4)에서 d_T, d_L, d_B, d_R 는 손실블록 경계안과 밖에 대한 위쪽, 왼쪽, 아래쪽, 오른쪽에 대한 픽셀의 차이이고 N은 매크로블록사이즈이다. $\hat{F}(m,n)$ 는 후보벡터를 이용하여 이전 프레임에서 보상된 블록이고 F는 현재프레임이다. d_S 는 추정된 블록의 경계정합 왜곡으로 그 값이 최소가 되는 후보벡터를 손실블록의 움직임벡터로 복원하여 움직임 보상을 수행한다.

3.3 픽셀의 방향성을 고려한 에러은닉방법

기존 경계정합알고리즘은 손실블록 경계선에 영

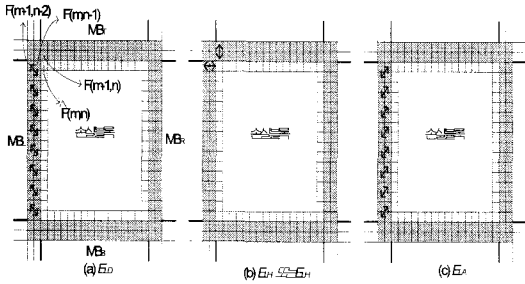


그림 4. MBMA(Modified Boundary Matching Algorithm)

상의 수평 또는 수직 에지가 놓여져 있을 때는 효과적인 방법이지만 주변에 인접한 픽셀들의 방향이 사선인 경우 경계정합방식으로 왜곡 값을 구할 때 오차가 발생하여 최적의 후보 벡터를 선정하지 못한 단점을 가지고 있다. 이런 문제점을 해결하기 위하여 수정된 경계정합알고리즘(MBMA: Modified Boundary Matching Algorithm)이 제안되었다^[13]. 참고문헌 [13] 방법은 손실블록에 인접한 블록 즉 위쪽, 왼쪽, 아래쪽, 오른쪽에 대하여 각각 3가지 방향(사선, 수평 또는 수직, 역사선)에 대한 왜곡 값을 구하여 최소인 값을 구한다. 이때 그림 4와 같이 손실주변에 인접한 두 픽셀간의 사선(E_{LD}), 수평(E_{LH}) 또는 수직(E_{LV}), 역사선(E_{LA})에 대하여 왜곡 값은 식 (5)와 같다. 즉, E_{LD} 는 손실블록에 인접한 왼쪽 블록 경계면에 대한 사선방향 왜곡 식이고, E_{LH} 는 수평방향에 대한 왜곡 식이고, E_{LA} 는 역사선에 대한 왜곡 식이다. 마지막으로 손실블록과 주변블럭 사이에 에지의 방향이 결정되면 에지의 방향에 맞게 다시 경계정합알고리즘을 사용하여 최적의 후보 벡터를 선정한다.

$$E_{LD} = \frac{1}{N-1} \sum_{i=0}^{N-2} |F(m-1+i, n-2) - F(m+i, n-1)|^2 \quad (5)$$

$$E_{LH} = \frac{1}{N-1} \sum_{i=0}^{N-2} |F(m+i, n-2) - F(m+i, n-1)|^2$$

$$E_{LA} = \frac{1}{N-1} \sum_{i=0}^{N-2} |F(m+i, n-2) - R(m-1+i, n-1)|^2$$

3.4 움직임 벡터의 방향성을 고려한 에러 은닉 방법

3.2절에서 소개한 경계정합알고리즘은 손실블록이 만약 영상 내에서 수직, 수평으로 움직였을 경우 성능이 좋지만, 영상의 에지부분이 손실블록과 주변블록에 걸쳐있는 경우 경계정합오차가 크게 나타나기 때문에 최적의 움직임 벡터를 찾는 데 어려움이 있다. 따라서 손실 블록에 인접한 움직임 벡터들의

방향을 고려하여 손실블록의 움직임 벡터를 복원하는 방법들이 있다^[5,6]. 참고문헌 [5]는 그림 4와 같이 8개의 방향을 구분한다. 즉 360°을 45°로 나누어 8가지 방향을 설정하여 손실 블록에 인접한 각각의 블록에 대하여 움직임 벡터의 방향에 대한 기울기 값을 고려하여 그룹들을 결정한다. 각 그룹 안에 존재하는 움직임 벡터들 중에서 중간 값을 후보 벡터로 한다.

참고문헌 [6]은 그림 5와 같이 주변에 인접한 움직임 벡터를 그룹 9개로 구분하여 클러스터 한다. 즉, 그림 5(a)는 4개의 그룹을 형성하고 움직임이 거의 없는 배경을 고려하여 GROUP0(0,0)을 추가하여 5개의 그룹을 형성한다. 그리고 그림 5(b)와 같이 중심선 주위에 움직임 벡터가 있는 경우를 고려하여 45°를 기준선으로 하여 그룹 4개를 추가해 9개의 그룹을 형성한다. 9개의 그룹을 기준으로 검색 트리를 이용하여 손실 블록에 인접한 움직임 벡터의 방향을 검색하여 그룹을 결정한다. 그리고 후보 벡터 선정방법은 참고문헌 [6]과 같은 방법으로 그룹 안에 존재하는 움직임 벡터들 중에서 중간 값을 후보 벡터로 한다.

참고문헌 [5]와 [6]과 달리 [14]는 그림7과 같이 디코더 측에서 움직임 벡터를 추정하는 DMVE (Decoder Motion Vector Estimation)방식을 제안하였다. 이 방법은 손실블록에 인접한 위쪽, 왼쪽, 아

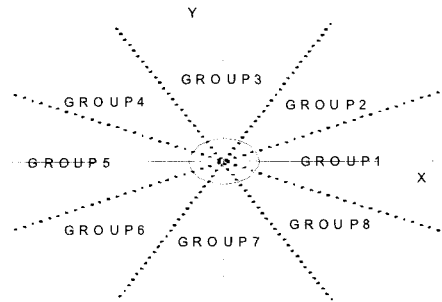


그림 5. 블록의 기울기에 따라 8가지 그룹 생성

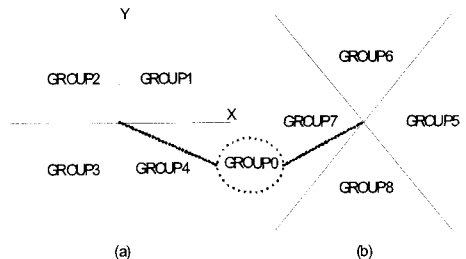


그림 6. 인접한 블록에 대한 9가지 그룹 생성 방법^[6]

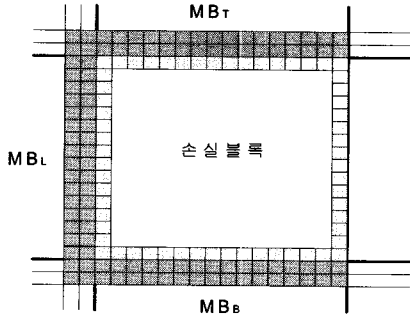


그림 7. DMVE(Decoder Motion Vector Estimation)

래의 두 라인 픽셀에 대하여 이전 프레임의 같은 위치에 있는 검색 영역(32×32) 안에서 움직임 추정을 한다. 이 방법은 기존 경계정합알고리즘 방법보다 PSNR 성능은 우수하지만 디코더에서 움직임을 추정하므로 계산 시간이 많이 걸리는 단점을 가지고 있다.

IV. 제안한 에러 은닉 방법

그림 8과 같이 손실블록에 인접한 움직임벡터들의 방향을 살펴보면 3개의 움직임 객체가 있다. 이때 객체1의 움직임은 북동쪽 방향, 객체2는

동쪽 방향, 객체3은 서남쪽 방향으로 움직인다. 손실 블록의 움직임 벡터는 움직임 객체 또는 배경이 될 수 있으므로 손실 블록은 인접블록의 움직임 벡터들과 높은 상관성이 있다고 볼 수 있다.

그러나 기존 경계정합알고리즘은 손실 블록에 인접한 움직임 벡터들의 방향성을 고려하지 않고 후보벡터 집합에서 왜곡 값이 최적인 후보벡터를 선

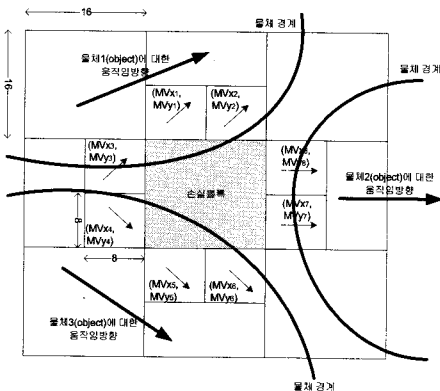


그림 8. 손실블록에 인접한 움직임벡터들의 방향

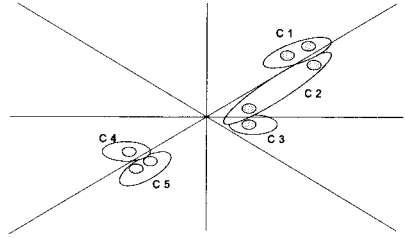


그림 9. 움직임벡터의 방향성을 고려한 클러스터 집합 생성

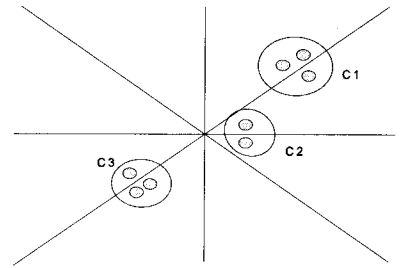


그림 10. 움직임벡터간의 거리를 고려한 클러스터 집합 생성

정하는 단점이 있고 [5][6]과 같이 손실 블록에 인접한 움직임 벡터들의 방향성을 고려하여 후보벡터 집합을 생성하는 방법도 있다. 하지만 그림 9의 C2를 보면 알 수 있듯이 움직임 벡터의 방향은 같지만 거리가 멀어 같은 그룹으로 보기가 힘들다. 또한 C1과 C2 결과에서 정해진 기울기 각도에 따라 그룹을 결정하므로 바로 인접한 움직임 벡터가 존재하여도 다른 그룹으로 결정한다는 단점이 있다. 따라서 본 논문에서는 그림 10과 같이 움직임 벡터들을 일정한 기울기의 각도를 그룹화 하지 않고 인접한 움직임 벡터들 간의 거리에 따라 클러스터 방법을 제안한다. 또한 손실블록을 은닉할 최적의 후보벡터를 결정할 때는 후보벡터집합 중에서 주변에 인접한 픽셀의 방향성을 고려하여 최적의 후보벡터를 결정한다.

본 논문에서 제안한 에러은닉 알고리즘은 다음 그림 9와 같이 절차로 진행된다. 첫째, 손실블록에 인접한 움직임 벡터를 검출한다. 둘째, 검출된 움직임 벡터들의 거리를 구하기 위하여 클러스터링의 대표적인 평균 연결 알고리즘(ALA: average linkage algorithm)을 수정하여 클러스터집합(cluster set)을 생성한다. 셋째, 클러스터 집합에서 손실블록을 은닉할 후보벡터집합(candidate vector set)을 선정한다. 넷째, 후보벡터집합 중에서 주변에 인접한 픽셀들의 방향성을 고려하여 왜곡 값이 최소인 후

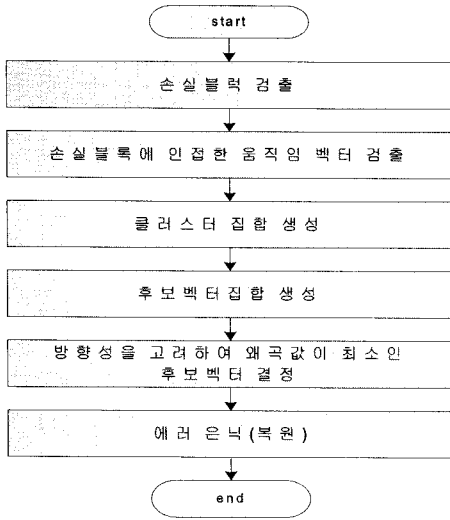


그림 11. 제안한 알고리즘 순서도

보벡터를 결정한다. 마지막으로 손실된 블록은 결정된 최적의 후보벡터로 에러를 은닉한다.

4.1 움직임 벡터 검출

H.264 부호기에서 생성된 패킷은 무선망으로 전송 중 패킷이 손실되므로 수신된 단말기에서 정확히 전송된 패킷을 검출한다. 이때 손실블록에 인접한 움직임 벡터만을 검출한다. 만약 그림 6에서 제시하듯 손실블록에 인접한 움직임 벡터들이 8개 정확히 전송되었다면 다음 식(6)과 같이 움직임벡터집합(motion vector set)을 구성할 것이다.

$$MVset = \{(MV_{x1}, MV_{y1}), (MV_{x2}, \dots, (MV_{xn}, MV_{yn})\}, n \leq 8 \quad (6)$$

4.2 클러스터 집합(cluster set) 생성

손실 블록에 인접한 블록의 움직임 벡터들 간의 거리(D: distance)는 식(7)과 같이 평균 연결 알고리즘을 이용한다. 이때 손실 블록에 인접한 움직임 벡터는 (x_n, y_n) 와 (x_{n-1}, y_{n-1}) 이고, D는 움직임 벡터 간의 거리이다.

$$D = \sqrt{|x_{n-1} - x_n|^2 + |y_{n-1} - y_n|^2}, (n \geq 2) \quad (7)$$

그림 6의 클러스터 집합은 다음 식(8)과 같이 3개의 클러스터 집합(C_1, C_2, C_3)으로 구성된다.

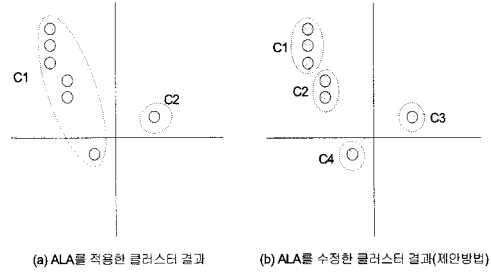


그림 12. 제안한 방법

$$\begin{aligned} C_1 &= \{(MV_{x1}, MV_{y1}), (MV_{x2}, MV_{y2}), (MV_{x3}, MV_{y3})\} \quad (8) \\ C_2 &= \{(MV_{x4}, MV_{y4}), (MV_{x5}, MV_{y5}), (MV_{x6}, MV_{y6})\} \\ C_3 &= \{(MV_{x7}, MV_{y7}), (MV_{x8}, MV_{y8})\} \end{aligned}$$

식(7)을 이용하여 움직임 벡터간의 거리를 구하고 최소가 되는 두 블록간의 평균값을 계산하는 과정을 반복함으로써 평균 연결 알고리즘을 수행하면 그림 12(a)와 같이 최종적으로 C1과 C2로 두 개의 그룹이 생성된다. 그러나 그림 12(a)와 같이 평균 연결 알고리즘은 최종적으로 두 개의 그룹을 생성하므로 생성된 그룹은 움직임 벡터들이 손실블록에 영향을 주지 않을 경우가 있다. 따라서 움직임 벡터간의 거리가 임계값(=10) 이상일 경우, 그림 12(b)와 같이 C2와 C4가 추가되어 총 그룹이 4개로 생성한다.

4.3 후보벡터집합(candidate motion vector set) 생성

이 단계에서는 앞 단계에서 생성한 각각의 클러스터 집합에서 최적의 후보벡터집합을 결정하는 단계이다. 식(8)에서 클러스터 집합은 3개로 구성되어 있고 이때 후보벡터집합은 각각 클러스터 집합에서 중앙값(median value)을 후보벡터로 선정한다. 따라서 후보벡터집합은 3개로 구성된다. 이때 중앙값을 선택한 이유는 기존 경계정합알고리즘의 후보벡터집합 중에서 중앙값으로 손실 블록을 은닉할 경우 화질성능이 우수하므로 중앙값을 선택한다^{7,81}.

4.4 손실블록에 대한 최종의 후보벡터 결정

기존 경계정합알고리즘의 왜곡 계산 방법은 손실 블록에 인접한 픽셀들의 방향성을 고려하지 않았기 때문에 후보벡터집합에 성능이 우수한 후보벡터가 있어도 선정되지 않은 경우가 많았다¹²¹¹³¹. 따라서 이 단계에서는 손실블록에 인접한 픽셀들의 방향성을 고려하여 최종 후보벡터를 선정한다. 최종 후보 벡터를 선정하는 방법은 그림 13과 같이 경계정합

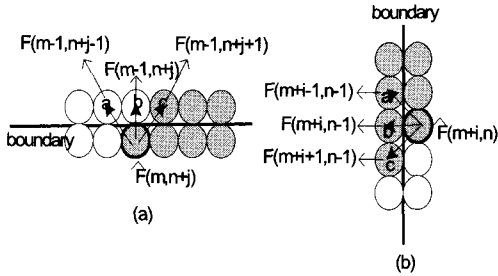


그림 13. 방향성을 고려한 왜곡 측정방법

알고리즘방식과 같은 방법으로 왜곡을 측정하다가 경계에 위치한 픽셀간의 차(b)가 일정한 임계치 (thr=140) 이상이면 영상의 경계 간에 변화가 있는 걸로 간주한다. 이렇게 경계부분에 영상의 변화가 있으므로 식(9)과 같이 경계간의 대각 방향성(a, c)을 고려하여 픽셀간의 차를 계산한다. 그리고 방향성을 고려한 픽셀간의 차(a_{thr}, b_{thr})가 가장 작은 픽셀 값을 선택하여 왜곡 값으로 계산한다.

$$b_{thr} = |\hat{F}(m, n+i) - F(m-1, n+j)| \quad (9)$$

$$\begin{aligned} & \text{if}(b_{thr} \geq thr) \\ & \begin{cases} a_{thr} = |\hat{F}(m, n+i) - F(m-1, n+j-1)| \\ c_{thr} = |\hat{F}(m, n+i) - F(m-1, m+j+1)| \end{cases} \\ & \text{else} \\ & d_T = b_{thr}; \\ & \text{if}(a_{thr} > c_{thr}) \\ & d_T = c_{thr}; \\ & \text{else} \\ & d_T = a_{thr}; \end{aligned}$$

식(9)에서 $\hat{F}(m, n)$ 는 후보벡터를 이용하여 이전 프레임에서 보강된 블록이고, $F(m, n)$ 는 현재프레임이다. $a_{thr}, b_{thr}, c_{thr}$ 는 각각 a, b, c 방향성을 고려한 경계간의 픽셀의 차이이다. 식(4)과 같이 손실블록에 인접한 블록의 경계안과 밖에 대한 위쪽(d_T), 왼쪽(d_L), 아래쪽(d_B), 오른쪽(d_R)에 대하여 각각 왜곡을 측정한다. 그리고 d_s 와 같이 추정된 블록의 경계 정합왜곡 값이 최소가 되는 후보벡터를 손실블록의 움직임벡터로 복원하여 움직임보상을 수행한다.

V. 실험 및 결과

본 논문에서는 H.264부호기에서 부호화된 패킷이 무선으로 전송될 때 손실된 패킷에 대하여 H.264 복호기에서 효율적으로 에러 은닉 방법을 제안한다. 제안한 알고리즘을 검증하기 위하여 H.264 참조 소

표 1. 실험환경

부호화 파라미터값	MV Resolution	1/4pel
	Hadamard Transform	ON
	RDO	ON
	Search Range	±16
	Reference Frames	1
	Symbol Mode	CAVLC
	GOP Size	10
시스템 환경	FMO	Disperse, Interleave
	CPU	인텔 펜티엄 3.20GHz
	메모리	512 MHz
	운영체제	window XP Pro.

프트웨어인 JM9.3을 사용하였다⁹⁾. 부호화된 비트 스트림의 출력 파일 형태를 실시간 트랜스포트 프로토콜(RTP: realtime transport protocol) 패킷 모드로 시뮬레이션을 하였다. 본 논문을 실험하기 위한 환경은 [표 1]과 같다. 그리고 실험에 사용된 영상은 기존 영상 부호기에서 성능 평가 테스트로 광범위하게 사용되고 있는 CIF(352×288)급의 FOREMAN, MOBILE, STEFAN, TABLE TENNIS 영상과 QCIF(176×144)급의 SUZIE 영상이다. 또한 이 영상들은 각각 10fps로 부호화 하였다. I 프레임과 P 프레임에 대한 초기 양자화 계수는 '28'로 설정하여 부호화하고, 이 실험에서는 B 프레임을 사용하지 않았다. 또한 H.264의 에러 복구 기술인 FMO 방식 중 디스펠스 방식과 인터리브 방식을 적용하였다. 인터넷 상의 전송 중 패킷 손실 환경을 구현하기 위하여 RTP 패킷으로 구성된 비트 스트림에서 임의의 패킷을 손실시키는 C 프로그램을 수정하여 RTP 패킷 손실을 시뮬레이션 하였다¹⁰⁾¹¹⁾.

기존의 손실 블록 은닉 방법인 경계정합알고리즘과 움직임 방향성을 고려한 참고문헌 [6] 및 [13] 방법, 그리고 본 논문에서 제안한 손실 블록 알고리즘은 영상압축 부호화에 의하여 저하된 복원 영상의 품질을 평가하기 위하여 일반적으로 사용하고 있는 PSNR(Peak Signal-to-Noise Ratio)을 사용한다.

[표 2]와 [표 3]은 FMO 방법 중 디스펠스 방식과 인터리브 방식으로 부호화된 영상들을 경계정합 알고리즘과 움직임의 방향성을 고려한 방법[6] 및 [13], 제안한 방법에 대한 후보벡터 개수, 디코딩 시간(sec), PSNR를 비교한 값이다. 부호화 방식의 차이점 때문에 디스펠스방식으로 은닉한 영상들은

대부분 인터리브 방식보다 화질은 좋지만 후보벡터 개수가 많으므로 전체 디코딩 시간은 인터리브방식보다 더 소요되었다.

[표 2]의 디스펠스방식으로 부호화 영상에 대한 각각의 은닉방법들을 비교해보면, 제안한 알고리즘의 후보벡터 개수는 기존 경계정합알고리즘(참고문헌 [13]) 및 참고문헌 [6]에 비해 평균적으로 각각 61%, 48% 가 감소되었다. 또한 [표 3]의 인터리브 방식으로 부호화 영상에 대한 각각의 은닉방법들을 비교해 보면, 제안한 알고리즘의 후보벡터 개수는 기존 경계정합알고리즘(참고문헌[13]) 및 [6]에 비해 평균적으로 각각 52%, 23% 가 감소하여 전체 디코

딩 시간을 줄였다. 그러나 디스펠스 및 인터리브방식으로 은닉된 영상들은 객관적인 화질을 측정하는 PSNR는 기존 방법 및 제안한 알고리즘은 거의 비슷하였다. 또한 [표2]와 [표3]의 BMA 및 MBMA결과를 보면 BMA이 MBMA보다 PSNR 및 디코딩시간이 우수하는 걸로 나타났다. 그 이유는 H.264의 후보벡터개수는 주변블록의 8x8블록 및 16x16블록의 움직임 벡터를 포함하여 최소 4개~최대 8개이다. 즉, 8x8블록의 움직임벡터 값이 16x16순실블록에 영향을 끼치므로 정확한 후보벡터를 선정하지 않는 걸로 판단된다.

표 2. 기존 여러은닉방법과 제안한 방법에 대한 후보벡터개수, 디코딩 시간, PSNR에 대한 성능 비교(disperse)

Sequences	성능비교	BMA (A)	[6](B)	[13](C)	제안한 방법(D)	A-D	B-D	C-D
FOREMAN (CIF)	후보벡터개수	12,294	7,814	12,294	4,014	-8,280(67%)	-3,800(49%)	-8,280(67%)
	디코딩 시간	24.42	18.92	27.26	18.62	-5.58	-0.30	-8.64
	PSNR	33.25	33.25	33.20	33.29	0.04	0.04	0.09
SUZIE (QCIF)	후보벡터개수	512	362	512	151	-361(71%)	-211(58%)	-361(71%)
	디코딩 시간	13.43	11.65	17.01	11.37	-2.06	-0.28	-5.64
	PSNR	36.07	36.11	36.08	36.12	0.05	0.01	0.04
MOBILE (CIF)	후보벡터개수	4,320	2,412	4,320	826	-3,494(81%)	-4,586(66%)	-3,494(81%)
	디코딩 시간	26.01	20.68	30.76	20.4	-5.61	-0.28	-10.36
	PSNR	29.81	29.77	29.73	29.78	-0.03	0.01	0.05
TABLE TENNIS (CIF)	후보벡터개수	4,139	2,982	4,139	1,031	-3,108(75%)	-1,951(65%)	-3,108(75%)
	디코딩 시간	13.32	11.1	16.62	10.96	-2.36	-0.14	-5.66
	PSNR	30.3	30.05	30.00	30.04	-0.26	0.01	0.04
STEFAN (CIF)	후보벡터개수	21,570	12,930	21,570	5,949	-15,621(72%)	-6,981(54%)	-15,621(72%)
	디코딩 시간	25.65	18.5	29.40	18.26	-7.39	-0.24	-11.14
	PSNR	30.43	30.3	30.38	30.48	0.05	0.18	0.10

표 3. 기존 여러은닉방법과 제안한 방법에 대한 후보벡터개수, 디코딩 시간, PSNR에 대한 성능 비교(interleave)

Sequences	성능비교	BMA (A)	[6](B)	[13](C)	제안한 방법(D)	A-D	B-D	C-D
FOREMAN (CIF)	후보벡터개수	2,504	1,542	2,504	964	-1,540(62%)	-578(37.5%)	-1,540(62%)
	디코딩 시간	23.2	18.75	28.15	17.96	-5.24	-0.79	-10.19
	PSNR	33.08	33.09	33.14	33.1	0.02	0.01	-0.04
SUZIE (QCIF)	후보벡터개수	347	176	347	135	-212(49%)	-41(30%)	-212(49%)
	디코딩 시간	13.28	13.92	18.04	12.7	-0.58	-1.22	-5.34
	PSNR	35.94	35.91	35.97	36.09	0.15	0.18	0.12
MOBILE (CIF)	후보벡터개수	1,437	769	1,437	402	-1,035(72%)	-367(47.7%)	-1,035(72%)
	디코딩 시간	25.29	26.46	30.49	20.59	-4.70	-5.87	-9.90
	PSNR	29.5	29.51	29.39	29.49	-0.01	-0.02	0.10
TABLE TENNIS (CIF)	후보벡터개수	1,706	744	1,706	721	-985(56%)	-23(3.2%)	-985(56%)
	디코딩 시간	12.95	13.26	27.46	12.96	0.01	-0.30	-14.50
	PSNR	29.03	28.98	29.01	29.04	0.01	0.06	0.03
STEFAN (CIF)	후보벡터개수	11,840	6,667	11,840	4,673	-7,167(61%)	-1,994(29.9%)	-7,167(61%)
	디코딩 시간	23.7	23.93	29.20	18.18	-5.52	-5.75	-11.02
	PSNR	29.66	29.51	29.38	29.54	-0.12	0.03	0.16



그림 15. 영상비교(interleave)

VI. 결 론

이동 통신 환경과 같이 무선채널 환경은 에러에 노출된 환경으로서 만족스러운 동영상 서비스를 제공하지 못한다. 이런 점을 착안하여 본 논문에서는 H.264 부호기에서 부호화된 패킷을 이동 통신 환경을 통하여 전송되어지는 패킷 중에서 손실된 패킷에 대하여 수신측의 단말기에서 에러를 은닉할 수 있는 방법을 연구하였다. 본 논문에서는 손실 블록에 인접한 움직임 벡터들을 이용하여 거리가 유사한 움직임 벡터들을 클러스터한다. 이때 클러스터된 움직임 벡터들은 후보벡터 집합으로 선정하여 후보벡터집합에서 각각 대표 후보벡터를 결정한다. 이렇게 결정된 후보벡터들은 주변에 인접한 픽셀들 간에 방향성을 고려하여 왜곡값이 최소인 후보벡터를 손실블록으로 은닉한다. 패킷이 손실되는 환경에서 실험한 결과, 화질에 대한 객관적 평가 기준인 PSNR 측면에서 제안한 알고리즘은 평균적으로 기존 방법들과 거의 비슷하였다. 하지만 후보벡터 개수가 평균적으로 23%~61%까지 감소하여 후보벡터에 대한 왜곡 측정 계산량이 평균적으로 3~4(sec) 감소하였다. 따라서 이동 통신 환경과 같이 저 계산 시간을 요구하는 모바일 단말기에 적합한 알고리즘을 개발하였다.

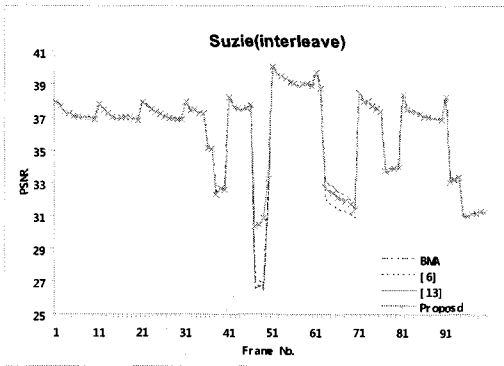


그림 14. Suzie 영상에 대한 PSNR 비교(Interleave)

그림 15(a)는 Suzie 영상의 46번째 원영상이고 (b), (c), (d), (e)는 각각 경계정합알고리즘, 참고문헌[6] 방법, 참고문헌[13] 방법, 제안한 방법으로 은닉한 영상이다. 결과에서 제시하듯 제안한 방법이 기존 방법들에 비하여 화질면에서 우수하게 은닉하였다. 특히, 그림 14에서 Suzie 영상의 46번째 프레임에서 PSNR 차이가 뚜렷하게 기존 방법보다 우수하게 나타났다.

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," 2003(5).

[2] L. E. G. Richardson, "H.264 and MPEG-4 Video Compression: video coding for next-generation multimedia," John Wiley & Sons Ltd. 2003.

[3] Y. K. Wang, M. M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The error concealment feature in the H.26L test model," *IEEE Int. Conf. Image Processing*, pp.729-733, 2002.

[4] T. Sockhammer, M. M. Hannuksela, T. Wiegand, "H.264/AVC in wireless environment," *IEEE Trans. Circuits Syst. Video Technol.*, Vol.13, No.7, pp.657-673, 2003(7).

[5] S. C. Park, N. R. Son, J. H. Kim, and G. S. Lee, "Recovery of motion vectors by detecting homogeneous movements for H.263 video communications," *IEEE ICME*, pp.33-36, 2001.

[6] S. Ghanbari and M. Z. Bober, "A cluster based method for the recovery of the lost motion vectors in video coding," *IEEE Mobile and wireless communications network*, 2002.

[7] K. David and D. Peter, "Error concealment techniques for H.263 video transmission," *IEEE communication, computers and signal processing*, 1999.

[8] A. Edurado and J. D. Edward, "Real time error concealment in compressed digital video streams," *IEEE Picture coding Symposium*, 1999.

[9] <http://iphome.hhi.de/suehring/tml/>

[10] S. Wenger, "Common conditions for wire-line low delay IP/UDP/RTP packet loss resilient testing," ITU-T SG16 Doc. VCEG-N79r1, 2001.

[11] <http://www.stewe.org/>

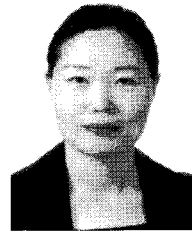
[12] W.M.Lam, A.R.Reilbman and B. Liu, "Recovery of lost or erroneously received motion vectors", *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.417-420, Apr. 1993.

[13] Jain Feng, K. T. Lo and Hassan Mehrpour, "Error Concealment for MPEG video transmission", *IEEE Tran. on Consumer Electronics*, pp.183-187, May. 1997.

[14] Jian Zhang, John F. Arnold, and Michael R. Frater, "A Cell-Loss Concealment Technique for MPEG-2 Coded Video", *IEEE Trans. on Circuits and Systems for Video Technology*, Vol.10, No.4, 2000

손 남 레 (Nam-rye Son)

정회원



1995년 2월 호남대학교 컴퓨터 공학과 학사 졸업
 2000년 2월 전남대학교 대학원 전산학과 석사
 2005년 2월 전남대학교 대학원 전산학과 박사
 2007년~현재 호남대학교 인터넷 소프트웨어학과 교수

<관심분야> 비디오 코딩, e-learning, 방송통신융합 기술, 영상통신

이 귀 상 (Guee-sang Lee)

정회원

1980년 2월 서울대학교 전기공학과 학사
 1982년 2월 서울대학교 전자계산기공학과 석사
 1983년~1984년 금성통신연구소 연구원
 1984년~현재 전남대학교 컴퓨터정보학부 교수

<관심분야> 멀티미디어통신, 영상처리 및 컴퓨터 비전, 임베디드 시스템