

H.264/AVC에서 OBMA 기반의 시방향 에러 은닉 기법

* 김 동형, ** 정제창
한양대학교 전자통신전파공학과
kimdh@ece.hanyang.ac.kr

A Temporal Error Concealment Algorithm for H.264/AVC based on OBMA

* Donghyung Kim, ** Jechang Jeong
Dept. of Electronics and Computer Engineering
Hanyang University, Korea

요약

H.264/AVC 동영상 부호화 표준은 이전의 여러 동영상 부호화 표준에는 없던 새로운 부호화 도구들이 추가되었으며 이를 통하여 보다 높은 압축 효율을 보인다. 추가된 부호화 도구들로 인하여 H.264내의 매크로블록은 이전의 부호화 표준에서 보다 많은 정보를 포함한다. 하나의 매크로블록에 대해서 최대 16개까지의 서로 다른 움직임벡터를 가질 수 있으며 최대 4개의 서로 다른 참조프레임을 가질 수 있다. 또한 다양한 블록크기로의 움직임 추정하며 이는 매크로블록의 모드로서 정의된다. 따라서 H.264내의 매크로블록은 기존보다 많은 움직임벡터를 가질 뿐만 아니라 기존에는 없던 참조프레임과 매크로블록모드의 정보를 새로이 포함하고 있다.

반면 현재의 H.264내의 시방향 에러은닉기법은 이전 부호화 표준에서 사용하던 방법과 유사한 방법으로 에러가 발생한 블록의 주변 매크로블록의 움직임 벡터만을 고려하여 에러를 은닉한다.

본 논문은 H.264 부호화 표준의 특성을 고려하여 매크로블록이 포함하고 있는 다양한 크기의 움직임 벡터 및 참조 프레임 뿐 아니라 주변 매크로블록들의 모드를 이용하여 서로 다른 블록 크기로 에러를 은닉함으로써 매크로블록이 포함하고 있는 정보를 최대한 활용하고 이를 통한 효율적인 에러은닉 알고리즘을 제안한다. 제안하는 알고리즘은 기존의 H.264에서의 시방향 에러은닉 기법과 비교하여 적은 연산량만을 가지면서도 최대 2.17 dB까지의 향상된 화질을 나타낸다.

1. 서론

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/AVC는 기존의 동영상 표준(MPEG-2, MPEG-4 Part2, H.263 등)에 없던 다양한 부호화 도구들을 사용함으로써 기존의 표준들에 비해 보다 높은 부호화 효율을 갖는다. 새롭게 추가된 부호화 도구에는 정수 변환 및 양자화, 엔트로피 부호화(CAVLC, CABAC), 루프 필터, 1/4 화소단위의 움직임 추정, 인트라 예측부호화, 다중 참조영상 사용 그리고 다양한 블록 크기의 움직임 추정 및 보상 등이 있다. 이중 다양한 블록 크기의 움직임 추정기법은 기존의 부호화 표준과 비교하여 보다 많은 계산량을 요구하지만 그에 따라 보다 세밀한 단위로 움직임을 추정함으로써 부호화 효율을 높이는 데 상당한 영향을 미치는 부호화 도구이다. 움직임 추정에 사용되는 블록의 크기는 16x16, 16x8, 8x16, 8x8의 크기를 가지며 8x8의 블록은 다시 8x4, 4x8, 4x4의 크기로 세분화 되어진다. 이렇게 다양한 블록 크기를 사용하여 움직임을 추정함으로써 나타나는 또 다른 특징은 기존의 동영상 표준에 비해 보다 많은 움직임 벡터를 생성한다는 것이다. 즉, 기존의 동영상 표준이 16x16크기를 갖는 하나의 매크로블록에 대해 하나의 움직임 벡터만을 갖는 반면 H.264는 16x16크기의 매크로블록 하나가 가질 수 있는 움직임 벡터의 개수는 최대 16개이다. 이는 H.264에서의 오류은닉시 사용될 수 있는 주변 매크로블록의 움직임벡터 정보의 양이 기존의 부호화 표준에 비해 많음을 나타낸다. 또한 H.264에서는 여러장의 참조 프레임을 허용한다. 참조프레임 선택의 최소단위는 8x8 블록 크기이며 이는 하나의 매크로블록이 최대 4개의

서로 다른 참조 프레임 정보를 가질 수 있다는 것을 나타낸다. 이에 추가적으로 움직임 추정시 사용되는 블록의 크기를 결정짓는 매크로블록 모드 정보 또한 H.264내의 매크로블록에 포함된다. 따라서 H.264내의 하나의 매크로블록이 가지는 정보로는 최대 16개의 서로 다른 움직임 벡터, 최대 4개의 서로 다른 참조 프레임 그리고 매크로블록 모드이다.

MPEG-2가 제정된 이후부터 많은 시방향 에러은닉 기법들이 연구되어져 왔으며 최근 H.264에서의 에러은닉 기법 또한 다각도로 연구되어지고 있다. 하지만 보다 많은 움직임벡터를 고려하는 것을 제외한다면 이전의 부호화 표준에서의 적용방법과 대부분 유사성을 지니고 있다. 본 논문에서는 H.264 내의 매크로블록이 이전의 동영상 부호화 표준과 비교하여 보다 많은 정보를 포함한다는 특성을 이용하여 이러한 정보를 최대한 활용한 시방향 에러 은닉 기법을 제안한다. 즉, 오류가 발생한 블록에 인접한 매크로블록의 움직임 벡터 및 참조프레임뿐 아니라 매크로블록 모드정보도 함께 이용하여 에러은닉이 수행되는 블록의 크기 및 움직임벡터 후보군을 적용적으로 선택함으로써 현재 H.264에서의 시방향 에러은닉 기법에 비해 적은 연산량을 가지면서도 보다 높은 화질을 나타내는 알고리즘을 제안한다.

논문의 구성은 2장에서 현재 H.264에서의 시방향 에러은닉 기법을 기술하고 3장에서 본 논문에서 제안하는 적용적 에러은닉 알고리즘을 기술한다. 4장에서는 제안하는 알고리즘의 적용을 통하여 알고리즘의 타당성을 보이고 마지막 절에서 결론을 맺는다.

2. H.264/AVC에서의 시방향 에러은닉기법

2.1 BMA(Boundary Matching Algorithm) 및 OBMA(Overlapping BMA)

BMA(Boundary Matching Algorithm)기법은 동영상 부호화 표준에서 시방향 움직임추정을 통한 에러은닉에 가장 널리 사용되는 방법이다. 이는 인접한 화소들 간에 큰 상관성을 가지는 자연영상의 특징을 이용한 알고리즘으로서 에러가 발생한 매크로블록의 인접화소들과 주변 매크로블록의 움직임벡터로부터 추정된 블록의 가장자리 화소들 간의 차이값을 최소화 하는 움직임벡터를 선정함으로써 에러를 은닉한다. 아래 식(1-4)는 에러블록의 상하좌우에서의 인접화소들간의 차이를 비용값으로 산출한 수식이며 이를 이용하여 산출한 각 움직임 벡터값에서의 최종 비용값은 식 (5)와 같다.

$$BM_T = \sum_{x=x_0}^{x_0+15} |P_{x,y_0-1} - P_{x+M_x,y_0+M_y}^r| \quad (1)$$

$$BM_B = \sum_{x=x_0}^{x_0+15} |P_{x,y_0+16} - P_{x+M_x,y_0+15+M_y}^r| \quad (2)$$

$$BM_L = \sum_{y=y_0}^{y_0+15} |P_{x_0-1,y} - P_{x_0+M_x,y+M_y}^r| \quad (3)$$

$$BM_R = \sum_{y=y_0}^{y_0+15} |P_{x_0+16,y} - P_{x_0+15+M_x,y+M_y}^r| \quad (4)$$

$$Cost_{BM} = BM_T + BM_B + BM_L + BM_R \quad (5)$$

이때 $P_{x,y}$ 는 현재 프레임의 (x,y) 위치에서의 화소값을 의미하며, $P_{x,y}^r$ 는 참조프레임의 (x,y) 위치에서의 화소값을 나타낸다. M_x, M_y 는 에러가 발생한 블록에 인접한 주변 매크로블록으로부터 얻어낸 움직임벡터정보이다.

OBMA의 경우는 화면내의 에지성분을 어느 정도 고려할 수 있다는 장점을 가지며 이는 비용값 산출 시 인접화소간의 차이값 대신에 에러블록의 주변화소와 참조프레임에서의 동일 위치간의 차이값을 비용값으로 사용한다.

2.2 H.264/AVC에서의 시방향 에러은닉기법

H.264/AVC 이전의 대부분의 부호화 표준들은 16x16단위로 움직임을 추정함으로써 하나의 매크로블록이 가질 수 있는 움직임 벡터의 수는 하나로 한정되어 있었다. 반면 H.264에서는 움직임을 다양한 블록크기로 수행함으로써 하나의 매크로블록은 최대 16개의 움직임 벡터를 가질 수 있다. 또한 여러 장의 참조프레임을 사용함으로써 각 매크로블록은 참조프레임의 정보도 포함하고 있다. 프레임참조의 최소단위는 8x8블록으로 수행되기 때문에 하나의 매크로블록이 가질 수 있는 최대 참조프레임의 수는 4개이다.

H.264에서는 최소 8x8단위의 참조 프레임 정보와 최소 4x4단위의 움직임 벡터정보를 가지는 주변 매크로블록의 정보를 이용하여 에러를 은닉한다. 그림 1은 현재 H.264에서의 시방향 에러은닉기법에 사용되는 움직임 벡터 후보군을 나타낸다.

그림 1에서 보는 바와 같이 H.264에서 시방향 에러은닉 시 사용하는 움직임 벡터정보로서 상하좌우 방향으로 인접한 8x8블록의 움직임 벡터를 사용한다. 8x8블록의 크기의 단위로 움직임벡터를 선정하는 주된 이유는 8x8 단위로 참조영상을 선택하는 H.264의 특징 때문이다.

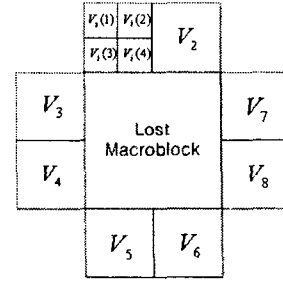


그림 1 인접 매크로블록의 움직임 벡터정보 및 참조 프레임 정보 $V_i = (M_x, M_y, Ref)$

H.264에서의 시방향 에러은닉 기법은 기본적으로 BMA에 기반하며 이때 사용되는 움직임 벡터 후보군은 아래와 같다.

$$V_{M_x, M_y, Ref} = [ZM, V_1, V_2, \dots, V_8] \quad (6)$$

where, $V_i = Mean(V_i(1), V_i(2), V_i(3), V_i(4))$

위의 수식에서 ZM은 후보 움직임벡터가 (0,0,0)인 경우를 나타내며 이는 바로 이전 프레임으로부터 같은 위치의 매크로블록을 복사하는 경우를 의미한다.

3. 제안하는 알고리즘

3.1 H.264 매크로블록의 블록모드

H.264에서는 이전의 부호화 표준과 다르게 다양한 블록 크기로 움직임을 추정한다. 아래 그림은 움직임 추정에 사용되는 다양한 블록 크기를 갖는 매크로블록의 모드를 나타낸다.

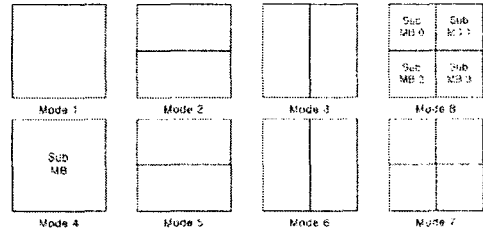


그림 2 움직임추정에 사용되는 매크로블록 모드

그림 2에서 Mode1, Mode2, Mode3, Mode8은 16x16의 매크로블록 크기를 나타내며, Mode4~Mode7은 8x8크기의 부 매크로블록 크기를 나타낸다.

3.2 적응적 시방향 에러은닉

제안하는 알고리즘은 에러가 발생한 블록의 인접화소들과 참조프레임에서 움직임을 고려하였을 때 같은 위치에서의 화소와의 차이값을 비용값으로 산정하는 OBM 기법을 기반으로 하며 주변 매크로블록의 움직임벡터 및 참조 영상뿐만 아니라 주변 매크로블록의 모드 정보를 이용하여 에러은닉에 사용되는 블록의 크기를 가변적으로 선택함으로써 보다 효율적으로 에러를 은닉한다. 에러은닉 모드는 에러은닉을 수행하는 블록의 크기에 따라 네 가지로 나뉜다. 에러은닉모드-1의 경우 기존의 H.264 에러은닉방법과 동일한 16x16의 크기로 에러은닉을 수행하며 에러은닉모드-2 및 에러은닉모드-3은 각각 16x8 및 8x16블록 단위로 에러은닉을 수행한다. 마지막으로 에러은닉모드-4의 경우는 8x8블록 단위로 에러를 은닉한다. 그림 3은

에러은닉모드에 따른 블록 모양 및 비용값 계산에 사용되는 주변 화소를 도시한다.

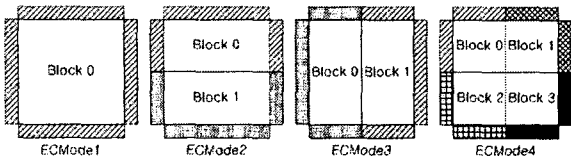


그림 3 서로 다른 블록 크기를 가지는 에러은닉모드

네 가지의 에러은닉 모드는 서로 다른 블록 단위로 에러를 은닉하며 그에 따라 비용값 산출에 사용되는 화소의 수도 서로 다르게 가진다.

3.2.1 에러은닉모드-1 (ECMode1)

ECMode1은 에러가 발생한 매크로블록의 상하 블록중 하나 이상이 Mode3 또는 Mode8 이외의 매크로블록 모드를 가지며, 좌우 블록 중 하나 이상이 Mode2 또는 Mode8 이외의 매크로블록 모드를 가질 때 선택되며 이 경우 기존의 H.264에서 채택하고 있는 방법과 동일하게 16x16단위로 에러를 은닉한다. 비용함수는 OBMA와 동일하게 사용하며 비용값 산출에 사용되는 화소의 개수는 64개이다. 이 경우 움직임벡터 후보군은 그림 1에서 보는바와 같이 주변 8x8블록의 평균 움직임벡터와 참조프레임을 모두 사용한다. 즉, 아래 식과 같이 9개의 후보움직임 벡터를 갖는다.

$$Block0 : \{ZM, V_1, V_2, \dots, V_8\} \quad (7)$$

3.2.2 에러은닉모드-2 (ECMode2)

ECMode2는 에러가 발생한 매크로블록의 좌우 블록 모두가 가로 방향으로의 나뉜 성분을 가지고 있는 Mode2 또는 Mode8의 매크로블록 모드를 가지며, 상하 블록중 하나 이상이 세로 방향으로 나뉜 성분을 가지고 있는 Mode3 또는 Mode8 이외의 매크로블록 모드를 가지는 경우 선택된다. 이 경우 에러은닉은 두개의 16x8의 블록 크기로 수행되며 각 세분화된 블록에 대한 움직임벡터 후보군은 다음과 같다.

$$Block0 : \{ZM, V_1, V_2, V_3, V_7\} \quad (8)$$

$$Block1 : \{ZM, V_4, V_5, V_6, V_8\} \quad (9)$$

3.2.3 에러 은닉 모드-3 (ECMode3)

ECMode3은 에러가 발생한 매크로블록의 상하 블록 모두가 세로 방향으로의 나뉜 성분을 가지고 있는 Mode3 또는 Mode8의 매크로블록 모드를 가지며, 좌우 블록중 하나 이상이 가로 방향으로 나뉜 성분을 가지고 있는 Mode3 또는 Mode8 이외의 매크로블록 모드를 가지는 경우 선택된다. 이 경우 에러은닉은 두개의 8x16 블록 크기로 수행되며 각 세분화된 블록에 대한 움직임벡터 후보군은 다음과 같다.

$$Block0 : \{ZeroMotion, V_1, V_3, V_4, V_5\} \quad (10)$$

$$Block1 : \{ZeroMotion, V_2, V_6, V_7, V_8\} \quad (11)$$

3.2.4 에러은닉모드-4 (ECMode4)

ECMode4는 에러가 발생한 매크로블록의 상하 블록 모두가 세로 방향으로의 나뉜 성분을 가지고 있는 Mode3 또는 Mode8의 매크로블록 모드를 가지며, 좌우 블록 모두가 가로방향으로의 나뉜 성분을 가지고 있는 Mode2 또는

Mode8을 가지는 경우 선택된다. 이 경우 에러은닉은 네개의 8x8 블록 크기로 수행되며 각 세분화된 블록에 대한 움직임벡터 후보군은 다음과 같다.

$$Block0 : \{ZM, V_1, V_3\} \quad (12)$$

$$Block1 : \{ZM, V_2, V_7\} \quad (13)$$

$$Block2 : \{ZM, V_4, V_5\} \quad (14)$$

$$Block3 : \{ZM, V_6, V_8\} \quad (15)$$

3.3 연산량 비교

에러은닉 기법에서는 객관적 화질만큼이나 복잡도 또한 중요한 성능으로 고려되어야 한다. 이는 복호화시에 에러블록을 포함하고 있는 프레임은 에러은닉과정이 끝난 후 다음 프레임을 복호화 하는 과정에서 참조프레임으로서 사용되기 때문이다. 따라서 높은 복잡도를 가지는 에러은닉 알고리즘은 실시간 복호화에 상당한 영향을 미칠 수 있다.

제안하는 알고리즘은 주변 매크로블록의 모드에 따라 네 가지의 블록 크기로 에러은닉을 수행한다. 각 에러은닉 모드에 따라 서로 다른 연산량을 가지며 상하좌우의 모든 매크로블록의 정보가 사용가능한 경우 하나의 매크로블록에 대해서 각 에러은닉모드에 따른 계산량은 아래 표와 같다.

표 1 에러은닉모드에 따른 연산량 비교

ECModes	Addition	Subtraction	Comparison
ECMode1	9	576	9
ECMode2	10	320	10
ECMode3	10	320	10
ECMode4	12	192	12

만일 모든 에러블록이 ECMode1으로만 선택되는 경우 모든 에러블록은 16x16 단위로 에러은닉을 수행하며 이 경우 기존의 H.264에서의 시방향 에러은닉 기법과 같은 연산량을 가진다. ECMode2-4의 경우에는 고려하는 움직임 벡터군을 에러은닉 모드에 따라 제한함으로써 ECMode1 보다 적은 계산량을 갖는다.

위의 표에서 뺄셈개수의 차이는 단순한 감산회수 이상의 의미를 갖는다. 이는 H.264에서의 움직임 추정이 1/4화소 단위로 수행되며 후보움직임벡터가 1/2 또는 1/4단위를 가질 때 참조프레임을 보간하여 뺄셈을 수행하기 때문에 보간에 사용되는 복잡도를 함께 고려하면 실제 각 모드간의 복잡도는 많은 차이를 보이게 된다.

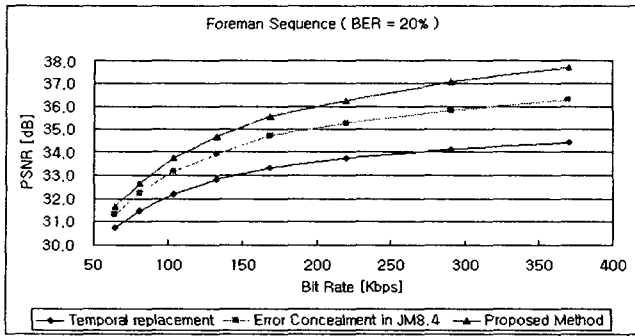
따라서 제안하는 알고리즘은 주변 매크로블록의 블록모드를 이용하여 적용적으로 에러은닉 블록 크기를 적용적으로 선택하고 각 에러은닉블록모드에 따라 움직임 벡터 후보군을 제한함으로써 기존의 H.264에서 사용되는 에러은닉기법보다 상대적으로 낮은 복잡도를 갖는다.

4. 실험 결과 및 분석

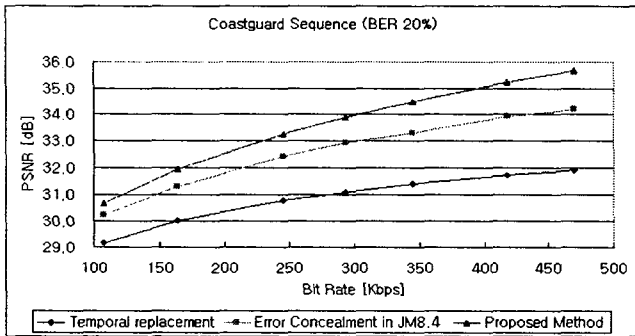
제안한 알고리즘은 H.264 참조 소프트웨어(JM 8.4)에 적용하였으며, QCIF 크기의 Foreman, Coastguard, Table 시퀀스 100 프레임에 대하여 실험하였다.

부호화는 시방향의 에러은닉 성능을 비교하기 위해서 IPPP 구조로 부호화 하였으며, 매크로블록 오류율이 5%, 10%, 15%, 20%에 대해서 각 비트율에 따른 PSNR을 비교하였다. 그림 4는 오류율이 20%인 경우 비트율에 따라 시방향 매크로블록 복사 방법과 현재 H.264에서의 에러은

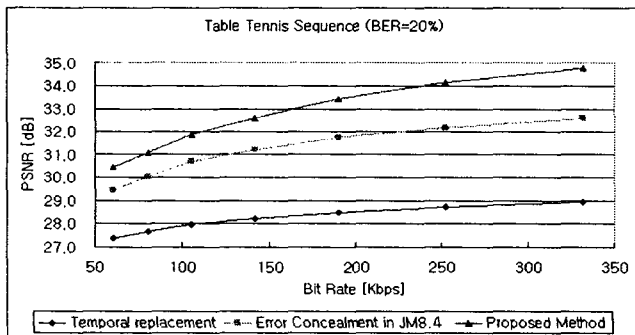
닉방법 그리고 제안한 알고리즘의 PSNR을 나타낸다.



(a) Foreman Sequence 100 프레임 (BER=20%)



(b) Coastguard Sequence 100 프레임 (BER=20%)



(c) Table tennis Sequence 100 프레임 (BER=20%)

그림 4 매크로블록 에러율이 20%일때의 실험결과

그림 4의 결과에서 보는바와 같이 제안한 알고리즘은 현재의 H.264에서의 시방향 에러은닉방법 보다 비트율에 상관없이 높은 성능을 보이는 것으로 나타난다.

아래의 표 2-4는 서로 다른 비율의 매크로블록 오류가 발생했을때 BER에 따른 PSNR을 나타낸다.

표 2 Foreman 영상에 대한 BER에 따른 PSNR

QP	Original PSNR	BER			
		5%	10%	15%	20%
24	$PSNR_{ZM}$	36.79	35.80	34.71	33.74
	$PSNR_{JM}$	37.51	36.77	36.12	35.26
	$PSNR_{PRO}$	38.06	37.55	36.89	36.25
22	$PSNR_{ZM}$	37.67	36.48	35.22	34.14
	$PSNR_{JM}$	38.58	37.67	36.80	35.82
	$PSNR_{PRO}$	39.33	38.68	37.80	37.06
20	$PSNR_{ZM}$	38.33	36.98	35.59	34.41
	$PSNR_{JM}$	39.46	38.36	37.41	36.29
	$PSNR_{PRO}$	40.38	39.64	38.58	37.73

$PSNR_{ZM}$: Temporal Replacement;

$PSNR_{JM}$: Error Concealment in JM8.4

$PSNR_{PRO}$: Proposed Method

표 3 Coastguard 영상에 대한 BER에 따른 PSNR

QP	Original PSNR	BER			
		5%	10%	15%	20%
26	$PSNR_{ZM}$	33.81	32.61	32.18	31.39
	$PSNR_{JM}$	34.86	34.06	33.79	33.33
	$PSNR_{PRO}$	35.27	34.82	34.70	34.49
25	$PSNR_{ZM}$	34.46	33.10	32.62	31.73
	$PSNR_{JM}$	35.74	34.81	34.46	33.94
	$PSNR_{PRO}$	36.22	35.65	35.48	35.23
24	$PSNR_{ZM}$	34.85	33.40	32.87	31.93
	$PSNR_{JM}$	36.19	35.15	34.81	34.21
	$PSNR_{PRO}$	36.80	36.14	35.95	35.68

표 4 Table 영상에 대한 BER에 따른 PSNR

QP	Original PSNR	BER			
		5%	10%	15%	20%
28	$PSNR_{ZM}$	32.11	30.51	29.64	28.49
	$PSNR_{JM}$	34.05	33.04	32.57	31.74
	$PSNR_{PRO}$	34.80	34.18	33.97	33.44
26	$PSNR_{ZM}$	32.65	30.88	29.96	28.72
	$PSNR_{JM}$	34.85	33.57	33.13	32.18
	$PSNR_{PRO}$	35.79	35.01	34.83	34.15
24	$PSNR_{ZM}$	33.19	31.25	30.27	28.94
	$PSNR_{JM}$	35.71	34.27	33.73	32.62
	$PSNR_{PRO}$	36.89	35.87	35.57	34.79

위의 결과에서 보는바와 같이 제안하는 알고리즘은 모든 비트율 및 BER에 대해서 기존의 방법에 비교하여 최대 2.17 dB까지의 높은 객관적 화질을 향상을 가져온다.

5. 결론

H.264내의 매크로블록은 4x4 블록 단위까지의 움직임 벡터 및 8x8단위의 참조프레임, 그리고 움직임추정 및 보상에 사용하는 블록의 크기를 정의하는 매크로블록 모드와 같이 이전의 다른 동영상 부호화 표준들에 비하여 보다 많은 정보를 포함한다. 제안한 알고리즘은 매크로블록 모드간의 상관성에 근거하여 에러블록의 주변 매크로블록 모드에 따라 서로 다른 네 가지의 크기를 갖는 에러은닉 모드를 적용적으로 선택하고 그에 따라 에러은닉을 수행함으로써 기존보다 적은 연산량을 가지면서도 비트율에 상관없이 보다 높은 객관적 화질은 나타낸다.

Reference

- [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] Jinghong Zheng and Lap-Pui Chau, "A temporal Error concealment algorithm for H.264 based on plane estimation," *ICICS-PCM 2003*, pp. 253-257, 2003.
- [4] JVT Model downloaded from <http://bs.hhi.de/~suehring/tml/download/jm84.zip>, accessed 2004