

MME와 OBMA를 이용한 H.264를 위한 개선된 에러은닉 방법

*조상협 **명성현 ***김동형 ****정제창

한양대학교 전자통신컴퓨터공학과

*jimmycho@ece.hanyang.ac.kr

An improved error concealment method for H.264 using the MME and the OBMA

*Cho, Sang-Hyup **Myung, Sung-Hyun ***Kim, Dong-Hyung ***Jeong, Je-Chang

Dept. of Electronics and Computer Engineering, Hanyang University

요 약

H.264/AVC 동영상 부호화 표준에서는 새로운 부호화 도구들이 추가되었으며 이를 통해 이전의 동영상 표준과 비교하여 높은 압축 효율을 가진다. 특히, VBS(Variable Block Size)와 다중 참조 프레임의 사용으로 H.264내의 매크로블록은 이전의 부호화 표준들에 비하여 보다 많은 정보들을 포함한다. H.264 부호화 표준에서는 하나의 매크로블록이 최대 16개까지의 서로 다른 움직임 벡터를 가질 수 있으며 최대 4개의 서로 다른 참조프레임을 가질 수 있다. 또한 다양한 블록크기의 움직임 추정을 하며 이는 매크로블록 모드로서 정의된다. 따라서 H.264내의 매크로블록은 기존의 동영상 부호화 표준보다 많은 움직임 벡터를 가질 뿐만 아니라 기존에는 없던 참조프레임과 매크로블록 모드의 정보를 새로이 포함하고 있다.

본 논문은 H.264 부호화 표준의 특성을 고려하여 매크로블록이 포함하고 있는 다양한 크기의 움직임 벡터 및 참조 프레임 뿐만 아니라 주변 매크로블록들의 모드를 이용하여 손실된 매크로블록에 대해 에러를 은닉함으로써 매크로블록이 포함하고 있는 정보를 최대한 활용하는 효율적인 에러은닉 알고리즘을 제안한다.

1. 서론

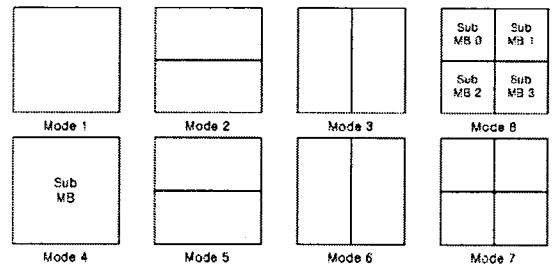
H.264/AVC 표준은 ITU-T의 VCEG(Video Coding Experts Group)과 ISO/IEC의 MPEG(Moving Picture Expert Group)의 공동 작업으로 제정되었다[1]. H.264/AVC 비디오 부호화 방식은 이전의 비디오 부호화 표준들(MPEG2, MPEG4 Part2, H.263)에는 없던 새로운 부호화 도구들, 정수 변환 및 양자화, 엔트로피 부호화, 루프 필터, 인트라 예측 부호화, 다중 참조 영상 및 1/4 화소 단위의 움직임 추정, VBS(Variable Block Size)의 움직임 추정, 적응적 프레임/필드 부호화 및 NAL 등이 추가되었으며, 이러한 도구들은 H.264 부호화 표준이 이전의 비디오 부호화 표준과 비교하여 보다 높은 압축효율을 갖게 하였다[1].

이중 VBS(Variable Block Size)의 움직임 추정기법은 기존의 부호화 표준에 비해 더 많은 계산양이 필요하지만 보다 세밀한 단위로 움직임을 추정하여 부호화 효율을 높일 수 있게 된다. 또한 기존의 동영상 표준이 16x16 크기를 갖는 하나의 매크로블록에 대해 하나의 움직임 벡터만을 갖는 반면 H.264는 16x16 크기의 매크로블록 하나가 가질 수 있는 움직임 벡터의 개수는 최대 16개이기 때문에 오류은닉 시 사용될 수 있는 주변 매크로블록의 움직임 벡터 정보의 양이 기존의 부호화 표준에 비해 많으므로 이를 이용하여 더 효율적인 움직임을 할 수 있게 된다.

본 논문은 하나의 매크로블록이 최대 16개의 서로 다른 움직임 벡터와 최대 4개의 서로 다른 참조프레임을 갖는 H.264의 부호화 특성을 고려하여 손실된 블록 주변의 매크로블록 모드에 따라 다양한 크기로 에러를 은닉하는 개선된 에러은닉 방법을 제안한다.

* 본 논문은 정보통신부의 출연금으로 수행한 IT SoC 핵심설계인력양성사업의 수행결과입니다.

* 본 논문은 정보통신부의 출연금으로 수행한 IT SoC 핵심설계인력양성사업의 수행결과입니다.



<그림 1> 움직임 추정에 사용되는 다양한 크기의 매크로블록 모드

2. H.264/AVC표준의 개관

가. H.264의 매크로블록 모드

H.264는 <그림 1>에서 보는 바와 같이 다양한 크기의 매크로블록을 이용하여 움직임을 추정한다. Mode1, Mode2, Mode3, Mode8은 16x16 크기의 매크로블록을 말하며 Mode4, Mode5, Mode6, Mode7은 8x8의 부 매크로블록을 의미하는데 영상에서 움직임이 많을 경우에 Mode8이 선택이 되어 부 매크로블록에서의 움직임에 대한 세부적인 추정을 할 수 있게 되는 것이다.

나. H.264에서의 비용함수

H.264에서는 각 매크로블록의 모드를 선택하기 위해 움직임 벡터의 비용함수(MVcost), 참조 프레임 비용함수(REFcost), 그리고 울-왜곡 비용함수(RDcost)를 사용하게 되고 각각은 다음과 같다.

$$MVCost = WeightedCost(f, mwbts[(cx \ll s) - px] + mwbts[(cy \ll s) - py]) \quad (1)$$

where, f : lambda factor

$$REFcost = WeightedCost(f, refbits(ref)) \quad (2)$$

where, f : lambda factor

$$RDCost = Distortion + lambda \times Rate \quad (3)$$

식 (1)에서 cx 와 cy 는 현재 매크로블록의 움직임 벡터를, px 와 py 는 예측된 움직임 벡터 나타내며, s 는 움직임 벡터의 변화량이고, Weighted Cost함수는 f 값을 사용하여 움직임 벡터 예측 오차의 부호화에 사용되는 비트량을 인수로 받아 비용값을 반환한다.

이렇게 구해진 값들을 이용하여 울-왜곡 최적화 기법을 사용할 경우, 울-왜곡 비용값을 최소로 하는 매크로블록 모드를 결정하고 그 외에는 움직임 벡터 비용값과 참조 프레임 비용값의 합이 최소가 되는 매크로블록 모드를 선택하게 된다.

3. H.264/AVC에서의 에러 은닉 방법

가. BMA 및 OBMA 에러은닉 방법

BMA(Boundary Matching Algorithm)방법은 인접한 화소들 간에 큰 상관성을 가지는 자연영상의 특징을 이용하여 에러가 발생한 매크로블록의 인접화소들과 주변 매크로블록의 움직임 벡터로부터 추정된 블록의 가장자리 화소들 간의 차이값을 최소로 하는 움직임 벡터를 선정하여 에러를 은닉한다. BMA에서 사용되는 비용함수는 식 (4)와 같다.

$$Cost_{BM} = \sum_{x=x_0}^{x_0+N-1} [|P_{x,y_0-1} - P_{x+v_x,y_0+v_y}| + |P_{x,y_0+N} - P_{x+v_x,y_0+N+v_y}|] + \sum_{y=y_0}^{y_0+N-1} [|P_{x_0-1,y} - P_{x_0+v_x,y+v_y}| + |P_{x_0+N,y} - P_{x_0+N+v_x,y+v_y}|] \quad (4)$$

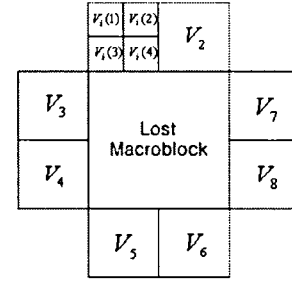
이 식에서 (x_0, y_0) 는 손실된 매크로블록의 좌상단의 픽셀 위치이며, (v_x, v_y) 는 후보 움직임 벡터들을 나타내고 P_x 와 P_x' 은 각각 현재 프레임과 참조 프레임의 화소값을 나타낸다.

OBMA(Overlapping BMA)는 BMA와는 다르게 인접화소간의 차이값 대신에 참조프레임에서 움직임 벡터를 고려한 후 동일 위치간의 차이값을 비용값으로 사용하며 비용함수는 아래 식 (5)와 같다.

$$Cost_{OBM} = \sum_{x=x_0}^{x_0+N-1} [|P_{x,y_0-1} - P_{x+v_x,y_0-1+v_y}| + |P_{x,y_0+N} - P_{x+v_x,y_0+N+v_y}|] + \sum_{y=y_0}^{y_0+N-1} [|P_{x_0-1,y} - P_{x_0-1+v_x,y+v_y}| + |P_{x_0+N,y} - P_{x_0+N+v_x,y+v_y}|] \quad (5)$$

나. H.264/AVC에서의 시방향 에러은닉방법

H.264에서는 움직임 추정을 다양한 블록크기로 수행함으로써 하나의 매크로블록은 최대 16개의 움직임 벡터를 가질 수 있고 여러 장의 참조프레임을 사용함으로써 각 매크로블록은 참조프레임의 정보도 포함하고 있다. 프레임참조의 최소단위는 8x8 블록을 사용하기 때문에 하나의 매크로블록이 가질 수 있는 최대 참조프레임의 수는 4개가 된다.



<그림 2> 인접 매크로블록의 움직임 벡터정보 및 참조프레임 정보 $V_i = (M_x, M_y, Ref)$

H.264는 8x8 단위의 참조 프레임 정보 및 4x4 단위의 움직임 벡터 정보를 가지는 주변 매크로블록의 정보를 이용하여 에러를 은닉하는데 <그림 2>에서 보는 바와 같이 H.264에서 시방향 에러은닉 시 사용하는 움직임 벡터정보로 상하좌우 방향의 인접한 8x8 블록의 움직임 벡터를 사용한다.

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

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

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

식 (6)의 ZM은 바로 이전 프레임의 같은 위치의 블록을 복사하는 경우를 말한다. 이와 같이 H.264에서는 시방향 에러은닉 시 하나의 매크로블록에 대해서 최대 9개의 움직임 벡터를 사용한다.

주변 매크로블록으로부터 움직임 벡터 후보군이 결정되면 식 (7)을 통하여 손실된 매크로블록의 최종 움직임 벡터를 선정하며 이를 이용하여 에러를 은닉하게 된다.

$$(v_x, v_y, ref) = (M_x, M_y, Ref) \text{ Arg Min}(Cost_{BM}(V_{M,M_y,Ref})) \quad (7)$$

식 (7)의 v_x, v_y 및 ref 는 비용값을 최소로 하는 움직임 벡터 및 참조 영상을 말하며 최종적으로 식 (8)을 이용하여 에러를 은닉한다.

$$MB_{conceal}(x, y, t) = MB(x, v_x, y, v_y, t - 1 - ref) \quad (8)$$

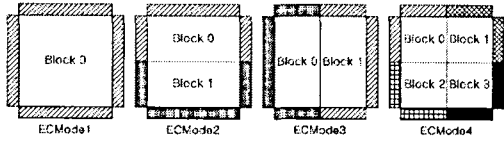
$i, j = 0, 1, \dots, 15 \quad ref = 0 \sim 4$

식 (8)은 손실된 매크로블록의 주변 매크로블록 정보로부터 최종 움직임 벡터 및 참조프레임이 결정된 후, 결정된 참조프레임에서 결정된 움직임 벡터에 위치한 블록을 가져옴을 의미한다. 다섯 장의 참조프레임을 사용하는 경우 ref 값은 0~4까지의 값 중 하나를 가질 수 있다.

3. 제안하는 알고리즘

제안하는 알고리즘은 크게 3 단계로 구성된다. 먼저 주변 매크로블록의 모드를 이용하여 손실된 매크로블록의 모드를 추정하고 추정된 모드에 따라 에러은닉 모드를 결정한다. 이 후 제안하는 알고리즘은 기존의 H.264의 에러은닉기법과는 다르게 OBMA 비용함수를 이용하여 에러를 은닉한다. 마지막으로 움직임 벡터의 미세 조정을 통하여 보다 정확한 에러은닉을 수행한다.

* 본 논문은 정보통신부의 출연금으로 수행한 IT SoC 핵심설계인력양성사업의 수행결과입니다.



<그림 3> 다양한 크기를 갖는 네 가지의 에러은닉모드

가. 적응적 시방향 에러은닉

앞서 기술한 H.264내의 비용함수 중 울-왜곡 비용값과 움직임 벡터 비용값은 어느 정도 비례하는 성질을 가지는데 이는 움직임 추정을 통한 예측오차의 크기가 비트율에 직접적으로 영향을 주기 때문이다. 가령 매크로블록이 Mode2로 선택되었다면 두 개의 16x8 블록 크기로 움직임 추정을 수행하는 것이 최소의 예측오차를 발생한다고 기대할 수 있다.

따라서 제안하는 알고리즘은 오류가 발생한 매크로블록의 모드를 인접한 매크로블록의 모드로서 예측하고 이를 이용하여 에러은닉에 사용되는 블록의 크기를 가변적으로 선택함으로써 보다 효율적으로 에러를 은닉한다. 에러은닉 모드는 <그림 3>과 같다.

네 가지의 에러은닉 모드는 서로 다른 블록 단위로 에러를 은닉하고 OBMA를 이용하여 비용값을 산출하며 각 에러은닉모드에 따라 서로 다른 개수의 참조 화소수를 갖는다.

에러은닉모드-1은 에러가 발생한 매크로블록의 상하 블록중 하나 이상이 Mode1, 또는 Mode2이며, 좌우 블록 중 하나 이상이 Mode1, 또는 Mode3인 경우 선택되며 이 때 기존의 H.264에서 채택하고 있는 방법과 동일하게 16x16 단위로 에러를 은닉하며 이때 사용되는 비용값은 식 (5)와 동일하다.

에러은닉모드-2일 경우 손실된 매크로블록의 좌우가 가로 방향으로의 나뉜 성분을 가지고 있는 Mode2 또는 Mode8의 매크로블록 모드를 가지며, 상하 블록 중 하나 이상이 Mode1, 또는 Mode2인 경우가 된다. 이 경우 에러은닉은 두개의 16x8의 블록 크기로 수행되며 각 세분화된 블록에 대한 비용값은 아래의 수식을 사용한다.

$$Cost_{Block0} = \sum_{x=x_0}^{x_0+15} (|P_{x,y_0-1} - P_{x+M_x,y_0-1}^{r_1} + M_x|) + \sum_{y=y_0}^{y_0+7} (|P_{x_0-1,y} - P_{x_0-1+M_x,y}^{r_2} + M_x|) + |P_{x_0+16,y} - P_{x_0+16+M_x,y}^{r_3} + M_x| \quad (9)$$

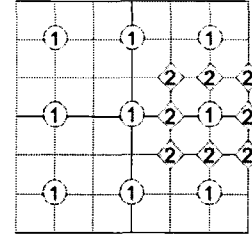
$$Cost_{Block1} = \sum_{x=x_0}^{x_0+15} (|P_{x,y_0+16} - P_{x+M_x,y_0+16}^{r_1} + M_x|) + \sum_{y=y_0+8}^{y_0+15} (|P_{x_0-1,y} - P_{x_0-1+M_x,y}^{r_2} + M_x|) + |P_{x_0+16,y} - P_{x_0+16+M_x,y}^{r_3} + M_x| \quad (10)$$

이때 각 블록에 대해 고려하는 움직임 벡터 후보군은 아래와 같이 각 블록 (Block0, Block1)에 인접한 주변 매크로블록의 움직임 벡터가 사용된다.

$$Block 0 : \{ZM, V_1, V_2, V_3, V_7\} \quad (11)$$

$$Block 1 : \{ZM, V_4, V_5, V_6, V_8\} \quad (12)$$

에러은닉모드-3 및 4도 에러은닉모드-2와 비슷한 처리 과정을 가지고 서로 다른 블록 크기로 에러를 은닉한다.



<그림 4> 2단계 움직임 벡터 정밀 검색 방법

나. 움직임 벡터 정밀 검색(MV-Refinement)

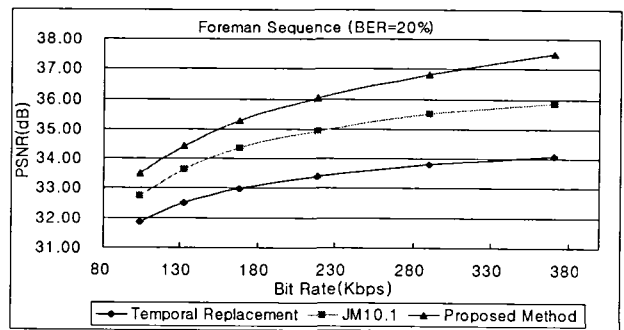
앞서 MME와 OBMA의 사용을 통해 선택된 손실된 매크로블록의 움직임 벡터는 한정된 후보군으로부터 선택하였기 때문에 추가적인 미세조정을 통해서 보다 정밀한 손실된 매크로블록의 움직임 벡터를 찾을 필요가 있다.

움직임 벡터의 미세조정은 부 화소(sub-pixel) 위치인 1/2 및 1/4 화소 위치에 대해서만 실행된다. 아래 그림은 움직임 벡터의 미세조정을 위해 제안한 고속 탐색 기법이다.

<그림 4>와 같이 제안하는 방법은 최초 8개의 1/2 화소 위치에서의 비용값을 계산하고 최소의 비용값을 갖는 1/2화소 위치 주변의 8개의 1/4 화소 위치의 비용값을 계산하여 최소의 비용값을 갖는 위치를 손실된 매크로블록의 움직임 벡터로 선정한다. 따라서 이 과정에서는 16개의 추가적인 비용계산이 요구된다.

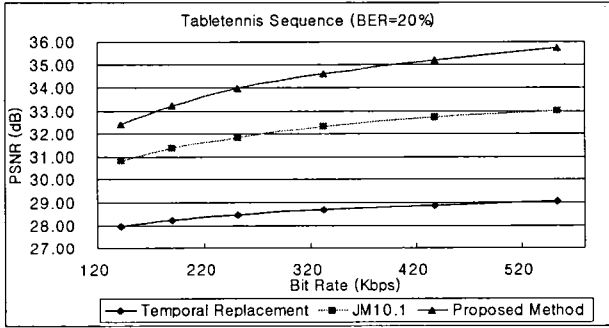
4. 실험 및 결과 분석

본 논문에서 제안한 알고리즘은 H.264 참조 소프트웨어(JM10.1)에 적용하였으며 실험을 위한 시퀀스로는 "foreman", "stefan", "table tennis"를 사용하였으며, 제안한 알고리즘이 중전의 방법인 JM10.1의 에러 은닉 방법과 시방향 블록 복사 방법에 비교하여 우수하다는 것을 보인다. 사용한 시퀀스는 QCIF(352X288)이고 각 시퀀스의 최초 100프레임을 IPPP 구조로 부호화한 결과를 복호화기에 입력하여 사용하였다. <그림 5>는 각 Sequence별로 매크로블록의 오류율이 20%일 때, 비트율에 대한 오류 은닉 후의 PSNR을 제안한 방법과 JM10.1의 에러 은닉 방법, 그리고 단순한 시방향 블록 복사 방법으로 에러를 은닉한 경우를 함께 나타내었다.

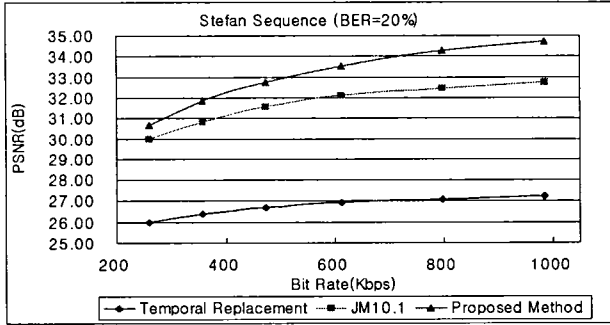


(a) Foreman Sequence 100 프레임 (BER=20%)

* 본 논문은 정보통신부의 출연금으로 수행한 IT SoC 핵심설계인력양성사업의 수행결과입니다.



(b) Table Tennis Sequence 100 프레임 (BER=20%)



(c) Stefan Sequence 100 프레임 (BER=20%)

<그림 5> 각 Sequence에 대한 실험 결과

<그림 5>의 실험 결과 도표에서 직관적으로 알 수 있듯이 제안한 MME와 OBMA를 이용한 에러 은닉 방법이 시방향 블록 복사 방법이나 JM10.1에 있는 에러 은닉 방법에 비하여 더 높은 PSNR을 나타내므로 우수한 성능을 보이고 있음을 확인 하였다.

<표 1> Foreman Sequence에 대한 BER에 따른 PSNR 비교

QP	Original	PSNR	BER			
			5%	10%	15%	20%
24	38.67	PSNR _{TR}	36.42	35.44	34.35	33.39
		PSNR _{JM}	37.08	36.40	35.81	34.91
		PSNR _{PRO}	37.74	37.27	36.61	36.02
22	40.23	PSNR _{TR}	37.28	36.11	34.87	33.79
		PSNR _{JM}	38.18	37.29	36.51	35.52
		PSNR _{PRO}	38.92	38.32	37.55	36.80
20	41.66	PSNR _{TR}	37.96	36.62	35.24	34.07
		PSNR _{JM}	38.96	37.89	37.01	35.86
		PSNR _{PRO}	40.01	39.29	38.36	37.50

PSNR_{TR} : Temporal Replacement

PSNR_{JM} : Error Concealment in JM10.1

PSNR_{PRO} : Proposed Method

<표 2> Tabletennis Sequence에 대한 BER에 따른 PSNR 비교

QP	Original	PSNR	BER			
			5%	10%	15%	20%
28	35.21	PSNR _{TR}	31.79	30.21	29.35	28.21
		PSNR _{JM}	33.71	32.60	32.14	31.35
		PSNR _{PRO}	34.49	33.92	33.73	33.21
26	36.47	PSNR _{TR}	32.34	30.58	29.67	28.44
		PSNR _{JM}	34.53	33.28	32.75	31.83
		PSNR _{PRO}	35.47	34.78	34.62	33.97
24	37.89	PSNR _{TR}	32.87	30.95	29.96	28.65
		PSNR _{JM}	35.48	33.36	33.34	32.32
		PSNR _{PRO}	36.51	35.60	35.42	34.60

<표 3> Stefan Sequence에 대한 BER에 따른 PSNR 비교

QP	Original	PSNR	BER			
			5%	10%	15%	20%
26	35.68	PSNR _{TR}	31.28	29.44	27.54	26.69
		PSNR _{JM}	34.73	33.35	31.99	31.55
		PSNR _{PRO}	35.08	34.08	33.07	32.73
24	37.36	PSNR _{TR}	31.94	29.87	27.79	26.91
		PSNR _{JM}	36.08	34.25	32.60	32.12
		PSNR _{PRO}	36.53	35.15	33.96	33.54
22	40.64	PSNR _{TR}	32.56	30.26	28.01	27.09
		PSNR _{JM}	37.30	35.02	33.05	32.45
		PSNR _{PRO}	38.05	36.30	34.77	34.27

<표1>, <표2>, <표3>은 서로 다른 매크로블록 오류율에 따른 PSNR을 나타내며, 제안한 알고리즘이 비트율 및 매크로블록 오류율에 관계없이 최대 2.73dB까지의 객관적 화질 향상을 가져옴을 알 수 있었다.

5. 결론

H.264 부호화 표준은 다중참조영상, 1/4-화소 단위의 움직임 추정, VBS(Variable Block Size)와 같은 새로운 부호화 도구를 사용함으로써 이전의 여러 부호화 표준들에 비해 높은 부호화 효율을 갖는다.

H.264에 새로이 채택된 여러 부호화 도구들로 인하여 매크로블록은 움직임 벡터뿐만 아니라 참조프레임 정보와 매크로블록 모드 정보도 포함한다. 매크로블록 내에 포함된 이러한 추가된 정보를 이용하여 제안한 알고리즘은 손실된 매크로블록의 모드를 추정하고 OBMA기반의 비용값을 이용하여 손실된 매크로블록의 움직임벡터를 추정한다. 또한 부화소 위치까지의 움직임벡터 미세조정을 사용하여 손실된 매크로블록의 움직임벡터를 복원해 냄으로써 이전의 H.264에서의 방법과 비교하여 최대 2.73dB까지의 높은 객관적 화질을 향상시켰다.

<참고 문헌>

- [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] Lain E.G. Richardson, "H.264 and MPEG-4 video compression", John Wiley & Sons, 2003.
- [3] J. Suh and Y. Ho, "Recovery of motion vectors for error concealment," *IEEE TENCON*, pp. 750-753, June 1999.
- [4] D, Kwon and P. Driessen, "Error concealment techniques for H.263 video transmission," *IEEE*, pp. 276-279, February 1999.
- [5] 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.
- [6] 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, Dec. 2003.

* 본 논문은 정보통신부의 출연금으로 수행한 IT SoC 핵심설계인력양성사업의 수행결과입니다.