

일반논문-08-13-3-03

H.264/AVC 표준에서 역트리 구조를 이용하여 고속으로 화면내 모드를 결정하는 방법

고 현 석^{a)}, 유 기 원^{a)}, 서 정 동^{a)}, 손 광 훈^{a)†}

Fast Intra-Mode Decision for H.264/AVC using Inverse Tree-Structure

Hyunsuk Ko^{a)}, Kiwon Yoo^{a)}, Jungdong Seo^{a)}, and Kwanghoon Sohn^{a)†}

요 약

H.264/AVC는 각 매크로블록에 대해서 최적의 부호화 모드와 참조 프레임을 결정해 주는 RDO (Rate-Distortion Optimization) 기법을 사용하여 기존의 비디오 압축 표준보다 더 좋은 부호화 효율을 얻고 있다. 하지만, RDO 기법은 하나의 매크로블록 모드를 결정할 때마다, 다양한 블록 타입의 화면내 (Intra) 예측을 수행하고 화면간 (Inter) 예측에 대해서도 1/4 화소까지 고려하는 움직임 추정(Motion Estimation)을 수행한 후 발생하는 비트까지 고려하여 최적의 모드를 결정하기 때문에 부호화기의 복잡도가 매우 큰 문제점이 있다. 따라서 영상의 객관적 화질은 유지하면서 부호화기의 복잡도를 낮추기 위한 많은 고속 알고리즘들이 제안되었고 연구 중에 있다. 본 논문에서는, 역 트리 구조의 경계 방향 예측 알고리즘을 이용한 고속 화면내 모드 결정 기법을 제안한다. 제안된 방법은 4x4 블록의 지역 경계 정보를 이용하여 해당 블록의 DE (Dominant Edge)를 찾아내고 DE에 상응하는 화면내 모드를 이용하여 RDO를 수행한다. 8x8 블록 (또는 16x16 블록)의 DE는 이전 단계 4개의 4x4 블록 (또는 8x8 블록) DE들로부터 계산되고, 이 단계에서의 RDO 또한 DE에 상응하는 화면내 모드를 이용한다. 실험결과 제안 방법은 화면내 부호화에 사용되는 후보 모드의 수를 줄임으로써 JM12.2와 비교하여 화면내 부호화 시간을 평균 64% 단축시킬 수 있었다.

Abstract

The H.264/AVC standard achieves higher coding efficiency than previous video coding standards with the rate-distortion optimization (RDO) technique which selects the best coding mode and reference frame for each macroblock. As a result, the complexity of the encoder have been significantly increased. In this paper, a fast intra-mode decision algorithm is proposed to reduce the computational load of intra-mode search, which is based on the inverse tree-structure edge prediction algorithm. First, we obtained the dominant edge for each 4x4 block from local edge information, then the RDO process is only performed by the mode which corresponds to dominant edge direction. Then, for the 8x8 (or 16x16) block stage, the dominant edge is calculated from its four 4x4 (or 8x8) blocks' dominant edges without additional calculation and the RDO process is also performed by the mode which is related to dominant edge direction. Experimental results show that proposed scheme can significantly improve the speed of the intra prediction with a negligible loss in the peak signal to noise ratio (PSNR) and a little increase of bits.

Keywords : H.264/AVC, Intra-prediction, Rate-distortion optimization (RDO), Intra-frame mode decision

a) 연세대학교 전기전자공학부

Dept. of Electrical and Electronic Engineering, Yonsei University

† 교신저자 : 손광훈(khsohn@yonsei.ac.kr)

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 IT신성장동력핵심기술개발사업의 일환으로 수행하였음. [2007-F-036-01, 인간 시각 특징 기반 주관적 화질 측정 방법 및 장치 기술]

1. 서 론

H.264/AVC는 기존의 비디오 압축 표준들과 비교하여 더 좋은 부호화 효율을 보여줄 뿐만 아니라, 복원된 영상의 화질과 PSNR 측면에서도 우수하다^[1]. 이러한 압축 성능의 향상은 주변 화소 정보를 이용한 화면내 부호화, 다양한 블록 사이즈의 움직임 추정 및 보상, 1/4 픽셀단위까지의 움직임 추정 기법, 다중 참조 프레임의 사용, 정수 기반의 DCT 변환 그리고 향상된 엔트로피 부호화 (Context-based Adaptive Binary Arithmetic Coding) 등의 이전에는 사용되지 않은 다양한 신기술로 인한 것이다. 이중에서도, RDO (Rate-Distortion Optimization) 기법은 부호화 효율을 극대화 하고 발생하는 비트량은 최소화 하면서 부호화기의 압축 성능을 향상시키는 가장 핵심적인 기법이다. 그림 1은 RDO 프로세스의 전체 개념도를 보여준다. 부호화기에서는 가능한 모든 모드의 조합에 따라 율-왜곡 관계를 조사하고, 그 중 최적 모드를 선택한다. 하지만, 하나의 매크로블록을 부호화하기 위해서는 모든 모드에 대해 RD_Cost를 구하는 과정이 필요하므로, 압축 효율은 높일 수 있으나 연산량이 증가하고 부호화 시간이 매우 길어지는 단점이 있다.

이러한 문제를 극복하고 부호화기의 복잡도를 줄이기 위한 연구가 활발히 진행되고 있고, 대표적인 것으로 고속 화면내 모드 결정 기법이 있다^{[2],[5]}. 화면내 예측 부호화는 블록 타입당 수행해야 할 후보 모드가 매우 많고, 화면내 예측 프레임 뿐만 아니라 화면간 예측 프레임에서도 수행되기 때문에 고속 화면내 모드 결정 기법을 통해서 전체 부호화기의 복잡도를 크게 줄일 수 있다. 이 기법들은 크게 두가지 방식으로 분류될 수 있다. 첫 번째는 모드 결정 과정에서 화면내 모드의 수행 유무를 결정하는 기법으로 인접 픽셀 값들과 차영상의 변환 계수를 이용한 선택적 화면 모드 기법^[2] 등이 제안되었다. 이러한 접근방식은 화면내 모드 자체를 RDO 과정에서 제외시킴으로써 부호화기의 복잡도를 보다 크게 줄일 수 있지만, 상대적으로 부호화 효율이 떨어지는 문제가 있다. 두 번째 접근방식은 화면내 모드 결정 단계에서 후보 모드의 개수를 줄이는 방법이다. 이러한 기법으로는 매크로블록의 경계 정보를 이용하는 방법^[3], 주변 블록간의 상관성을 이용한 3단계 결정 방법^[4], 주변 블록에

서 선택된 모드의 확률정보를 이용하는 방법^[5] 등이 있다.

본 논문에서는 역 트리 구조의 DE (Dominant Edge) 예측 기법을 이용하여 H.264/AVC의 연산 복잡도를 줄이는 새로운 고속 화면내 모드 결정 기법을 제안한다. 이때 DE는 블록 내 텍스처의 대표적인 방향성을 의미한다. 제안된 방식은 H.264/AVC High-Profile 에서 화면내 부호화를 위해 지원하는 3가지의 블록 크기, 즉 4x4/8x8/16x16 블록들의 DE간의 상관성을 이용한다.

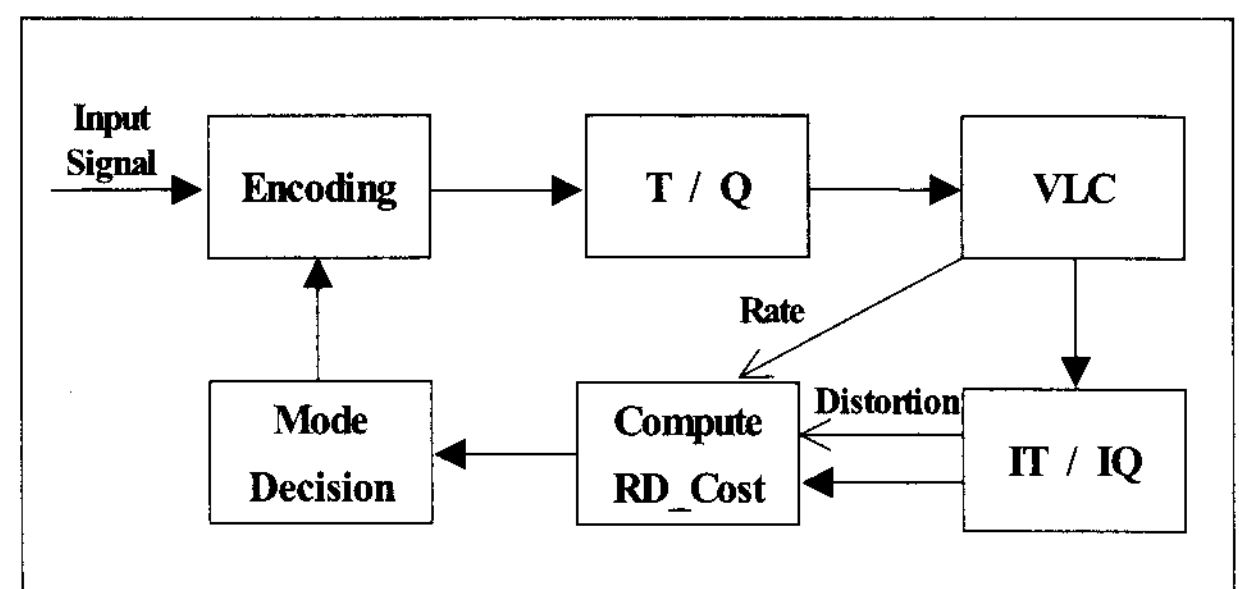


그림 1. H.264/AVC의 RDO (Rate-Distortion Optimization)
Fig. 1. Rate-Distortion Optimization for H.264

예를 들어, 8x8 블록의 DE는 그 블록을 구성하는 4개 4x4 블록의 DE들과 높은 상관성을 갖는다. 유사하게, 16x16 블록의 DE 또한 그 블록을 구성하는 4개 8x8 블록의 DE들과 높은 상관성을 갖는다. 따라서 4x4 블록들의 DE 정보들을 알고 있다면, 이 정보들로부터 상위 블록 (8x8/16x16)의 DE를 결정할 수 있고, 각 블록 모드에서의 RDO는 DE에 상응하는 방향의 모드를 이용하여 수행한다. 이 기법을 이용하여 RDO 과정에서의 후보 모드의 개수를 줄이고 화면내 부호화 시간을 단축시킬 수 있다. 실험 결과를 통해서 제안된 방법이 PSNR 감소 및 비트량의 큰 증가 없이 전체 부호화 시간을 평균 크게 감소시키는 것을 확인하였다.

본 논문은 다음과 같은 순서로 구성된다. 2장에서 H.264/AVC의 화면내 모드 결정 알고리즘에 대해서 설명하고 3장에서는 제안하는 고속 화면내 모드 결정 기법을 자세히 설명한다. 4장에서는 실험 결과를 통하여 H.264/AVC의 참조 모델 및 기존 방법과 제안한 방법의 부호화 성능을 비교하며 5장에서 결론을 맺는다.

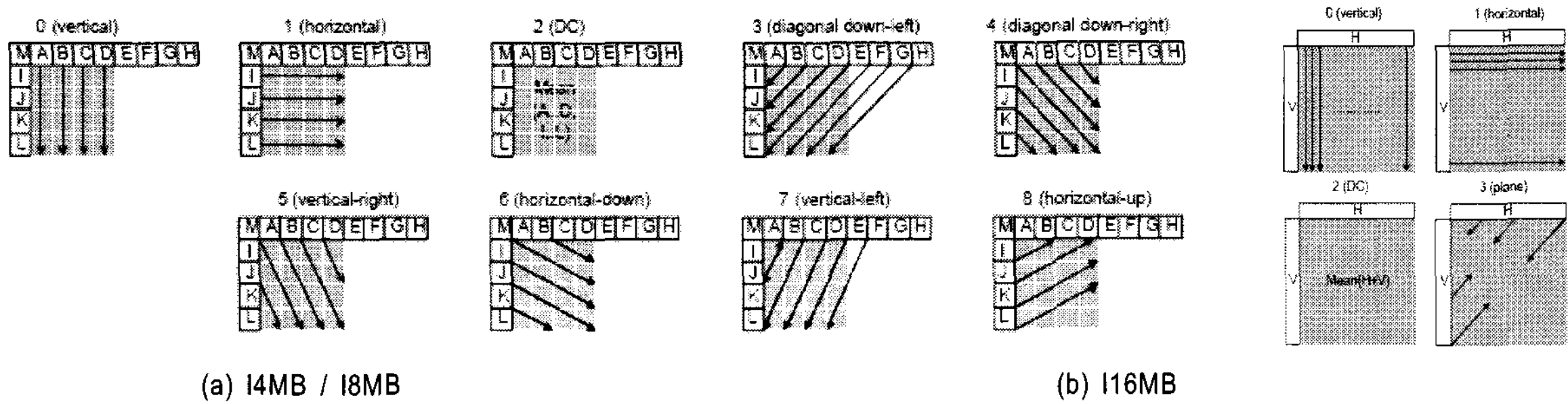


그림 2. H.264/AVC의 화면내 예측 모드

Fig. 2. Intra-prediction modes according to block types

II. H.264/AVC 의 화면내 모드 결정

H.264/AVC의 High-Profile 모드를 가정했을 때, 화면내 부호화를 위한 블록 사이즈는 16x16 (I16MB), 8x8 (I8MB), 4x4 (I4MB) 블록이 지원되고 화면간 부호화를 위한 블록 사이즈는 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4 블록이 지원된다. 최적의 매크로블록 모드를 결정하기 위해서, 부호화기는 상기된 모든 후보 모드들의 RD_Cost를 다음과 같이 계산한다^[6].

$$RD_Cost = SSD(S, R, Mode / QP) + \lambda_{mode} \cdot R(S, R, Mode / QP) \quad (1)$$

S는 원본 블록, R은 복원된 블록, QP는 양자화 계수를 뜻하며, λ_{mode} 는 Lagrange multiplier이고 $0.85 \times 2^{QP/3}$ 에 따라 계산된다. 두 블록간 Distortion은 SSD (Sum of squared error) 를 이용하여 원본 블록과 비교한 복원된 매크로블록의 왜곡 정도를 나타내고 다음과 같이 구한다.

$$SSD(S, R, Mode / QP) = \sum_{x=1}^{16} \sum_{y=1}^{16} [S_Y(x, y) - R_Y(x, y, Mode / QP)]^2 + \sum_{x=1}^8 \sum_{y=1}^8 [S_U(x, y) - R_U(x, y, Mode / QP)]^2 + \sum_{x=1}^8 \sum_{y=1}^8 [S_V(x, y) - R_V(x, y, Mode / QP)]^2 \quad (2)$$

S_Y, R_Y 는 각각 원본 신호와 복원된 신호의 휘도 성분을 의미하고, S_U, S_V 와 R_U, R_V 는 색차 성분을 의미한다.

$R(S, R, Mode / QP)$ 는 Mode 정보와 양자화 계수 값 그리고 차영상 계수들을 부호화하기 위해 필요한 비트량이다. 위와 같이 모든 모드에 대해서 RD_Cost를 계산한 후, 가장 작은 값을 갖는 모드를 해당 블록의 최적 모드로 결정한다.

특히 화면내 예측의 경우, 기존의 H.263 및 MPEG-2/4 비디오 압축 표준이 예측을 변환 도메인에서 수행했던 것에 반해 H.264/AVC에서는 공간 도메인에서 예측을 수행하고, 이때 이미 압축되고 복원된 현재 블록의 왼쪽 혹은 위쪽에 존재하는 이웃 샘플 값들을 이용한다. 앞서 언급한 대로, H.264/AVC High-Profile에서는 I4MB, I8MB, I16MB 그리고 chroma_8x8 4가지 종류의 화면내 예측 모드가 제공되고 각각의 모드는 다시 다양한 방향성을 지니는 예측 모드들로 세분화 된다. 예를 들어, I4MB와 I8MB는 각각 4x4 블록과 8x8 블록 휘도 성분에 대한 9개 방향의 예측 모드를 지원하고, I16MB는 16x16 블록 휘도 성분에 대한 4개 방향의 예측 모드를 지원한다. 그리고 chroma_8x8은 색차 신호에 대한 4개의 예측 모드를 지원한다. 그림 2는 화면내 예측 모드들의 방향성을 구체적으로 보여 준다. Chroma_8x8 예측 모드는 16x16 예측 모드와 유사하고 모드의 순서만 DC (mode 0), horizontal (mode 1), vertical (mode 2), plane (mode 3)로 다르다. H.264/AVC는 모든 예측 모드에 대해서 부호화를 수행한 후 가장 작은 RD_Cost를 갖는 모드를 최적 모드로 선택하기 때문에 부호화기의 복잡도가 증가하게 되고 따라서 부호화기의 연산량을 줄이고 전체 압축 시간을 줄이기 위한 고속 화면내 모드 결정 알고리즘이 필요하게 된다.

III. 제안 알고리즘

1. 역 트리 구조 기반의 고속 화면내 모드 결정 기법

RDO 과정에서, 휘도 성분 화면내 예측의 경우 I4MB, I8MB, I16MB에서 선택된 모드들 사이에는 큰 상관성이 존재한다. 예를 들어, I4MB 모드에서 선택된 예측 모드의 방향이 수평방향이라면 I8MB와 I16MB 모드에서 선택되는 예측 모드 또한 수평 방향이 될 확률이 크다. 본 논문은 이러한 경향성을 이용한 역 트리 구조의 화면내 고속 모드 결정 방식을 제안한다.

그림 3은 제안된 방식의 전체적인 개념도를 보여주는데, 4x4 블록의 모드 결정이 우선적으로 수행된다. 이 단계에서 블록의 DE를 결정하기 위해 해당 블록의 픽셀 단위의 경계 정보가 필요하다. 따라서 적절한 연산자를 이용하여 Edge Map을 생성하고, 이 Edge Map으로부터 4x4 블록 안의 모든 픽셀의 방향성을 고려하여 블록 내 텍스처의 대표적인 방향성을 나타내는 지표인 DE를 결정해준다. 최종적으로 선택되는 화면내 예측 모드는 DE의 방향성과 일치할 확률이 매우 높기 때문에 RDO는 DE에 상응하는 모드와 DC 모드에 대해서만 수행하고 나머지 모드는 모두 건너뛴다^[3]. 이때, DC 모드는 항상 후보 모드로 고려되는데 이는 DC 모드가 평활한 영역처럼 방향성이 약하게 나타나는 블록들에서 가장 빈번하게 선택 되는 모드이기 때문이다.

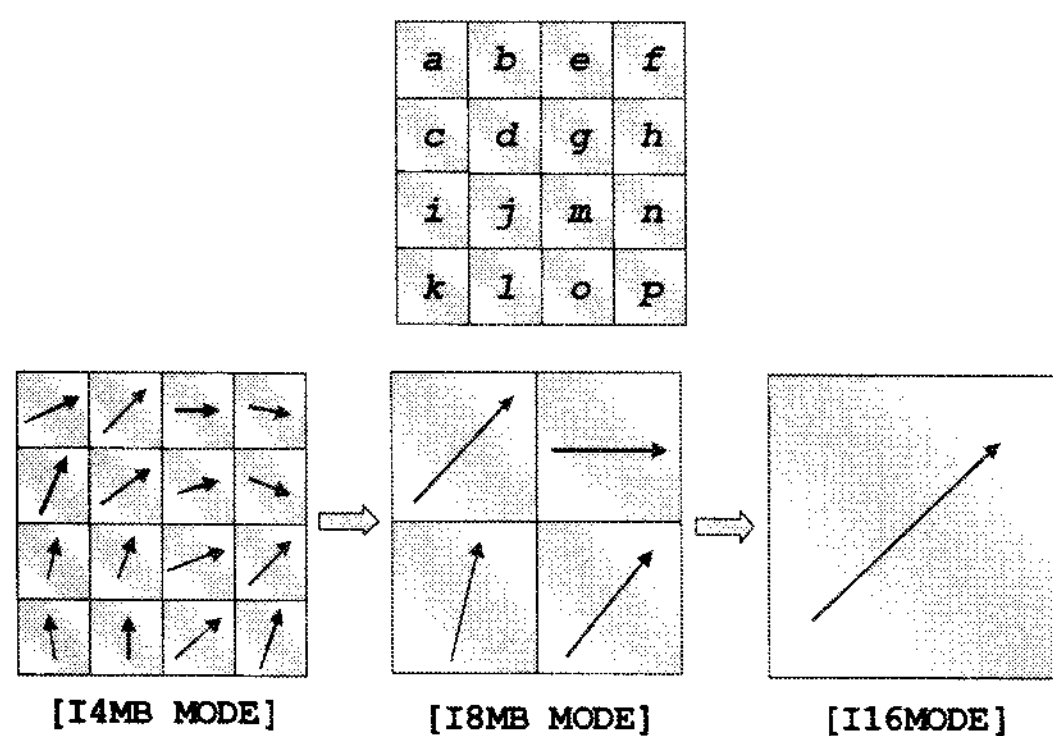


그림 3. 제안한 알고리즘의 전체 개념도
Fig. 3. Overall concept of proposed algorithm

8x8 블록의 RDO를 수행할 때는 4x4 블록에서 구한 DE

정보들을 이용한다. 예를 들어, 그림 3에서 첫번째 8x8 블록의 DE는 그 블록을 구성하고 있는 4개의 4x4 블록 'a, b, c, d'의 DE들로부터 예측할 수 있다. 동일한 방법으로, 두번째 8x8 블록은 'e, f, g, h', 세번째 8x8 블록은 'i, j, k, l', 마지막 8x8 블록은 'm, n, o, p'의 DE들로부터 예측된다. 마지막으로, 16x16 블록의 DE는 그 블록을 구성하는 4개의 8x8 블록의 DE들로부터 예측되고 8x8, 16x16 블록에서의 RDO 또한 DE에 상응하는 화면내 모드와 DC_Mode에 대해서만 수행한다. 제안된 기법은 8x8/16x16 블록의 DE를 구하기 위해 Edge Map을 다시 이용해야 하는 추가적인 연산이 필요 없다는 장점이 있다.

1.1 휘도 4x4 블록과 색차 8x8 블록의 모드 결정 기법

4x4 블록의 DE를 결정하기 위해 필요한 Edge Map은 소벨 연산자 (Sobel Operation)를 이용하여 프레임 단위로 구해진다. 구해진 Edge Map으로부터 Pan의 알고리즘^[3]을 이용하여 휘도 4x4 블록과 색차 8x8 블록의 DE의 방향을 결정한다. 색차 8x8 블록의 경우는 U와 V, 두 개의 성분을 고려해야 하고, U, V의 DE 각각이 화면내 예측 모드를 위한 후보 모드가 된다. 앞서 언급했듯이, RDO는 DE에 상응하는 화면내 모드와 DC_Mode에 대해서만 수행되고, 휘도 4x4 블록의 DE들은 이후 휘도 8x8 블록의 DE를 결정할 때 사용하기 위해 그림 4와 같이 각 모드에 상응하는 각도로 저장 된다.

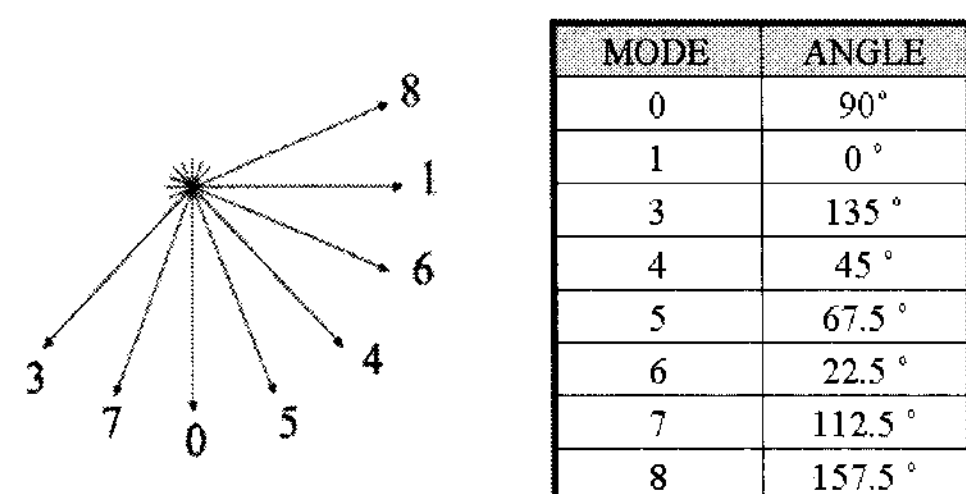


그림 4. 4x4 블록 모드들의 저장 각도
Fig. 4. Angle for each 4x4 block mode

1.2 휘도 8x8 블록 / 휘도 16x16 블록의 모드 결정 기법

8x8 휘도 블록의 화면내 예측에서는 후보 모드의 개수를 줄이기 위해, 이전 단계에서 저장된 4x4 블록들의 DE 정보를 이용한다. 8x8 블록의 DE는 다음과 같이 계산된다.

$$Dmn_Edge_{8 \times 8} = (First_DE_{4 \times 4} + Second_DE_{4 \times 4} + Third_DE_{4 \times 4} + Forth_DE_{4 \times 4}) / 4 \quad (3)$$

$N_{th_DE_{4 \times 4}}$ 는 $N(=1, 2, 3, 4)$ 번째 4×4 블록의 DE의 저장된 각도를 의미한다. 즉, 그림 3에서 좌측 상단 8×8 블록의 DE는 'a,b,c,d' 4개의 4×4 블록 DE들의 평균 각도가 된다. 8×8 블록의 DE가 결정되면, 후보 모드로 DE와 가장 가까운 각도에 상응하는 모드가 선택되고 8×8 블록들의 DE 또한 이후 16×16 블록의 DE 계산을 위해서 저장된다. 16×16 블록 모드 결정 단계에서 DE는 8×8 블록에서 사용했던 방식과 유사하게 다음 식과 같이 구하고 이에 상응하는 모드를 결정하여 RDO를 수행한다.

$$Dmn_Edge_{16 \times 16} = (First_DE_{8 \times 8} + Second_DE_{8 \times 8} + Third_DE_{8 \times 8} + Forth_DE_{8 \times 8}) / 4 \quad (4)$$

제안된 방법을 사용하면 RDO 과정에서의 후보 모드의 개수를 크게 줄일 수 있다. 표 1에 정리된 것과 같이, I4MB와 I8MB의 경우 기존의 9개의 후보 모드가 2개의 후보 모드로, I16MB의 경우는 4개에서 2개로 모드 수가 줄어든다. Chroma_8x8의 경우는 U, V 두 개의 요소가 존재하기 때문에 두 요소의 DE가 같은 경우는 기존 4개의 후보 모드에서 2개로, 서로 다른 경우에는 3개로 각각 줄어들게 된다. 표 2는 제안된 알고리즘의 부호화 단계를 구체적으로 보여준다.

표 1. 제안 알고리즘의 후보 모드 감소 효과

Table 1. Reduction of candidate mode for proposed algorithm

	Block Type	Total # of modes	# of used modes
Luma (Y)	4x4 (I4MB)	9	2
Luma (Y)	8x8 (I8MB)	9	2
Luma (Y)	16x16 (I16MB)	4	2
Chroma (U,V)	8x8 (chroma_8x8)	4	2 or 3

2. 적응 알고리즘

제안된 알고리즘은 Edge Map으로부터 얻은 4×4 블록의 DE 정보만을 기반으로 $4 \times 4/8 \times 8/16 \times 16$ 블록의 화면내 모드

결정을 하기 때문에 4×4 블록 단계에서 정확한 모드를 선택하는 것이 매우 중요하다. 특별히, $8 \times 8/16 \times 16$ 블록의 경우는 계산 복잡도를 줄이기 위해서, Edge Map을 이용하여 해당 블록의 DE를 직접 찾아내지 않고, 4×4 블록의 DE로부터 예측하는 방식을 이용한다. 따라서, 4×4 블록의 DE로부터 선택된 모드가 최적의 모드가 아닐 경우는, 그림 5에서와 같이 이후의 $8 \times 8/16 \times 16$ 블록에서 선택되는 모드도 부정확해지고 이로 인해 비트량이 증가하는 문제가 있다. 특히, 화면내 예측의 경우 매크로블록을 부호화하기 위해서 이미 부호화 된 후 복원된 인접 블록의 픽셀 정보를 사용하기 때문에 임의의 매크로블록의 모드 선택이 잘못되었을 경우 그로 인해 발생하는 오차가 다음 매크로블록으로 전파되고 부호화가 진행될수록 누적되는 심각한 문제가 발생한다. 그러므로, 제안된 방식으로 4×4 블록에서 선택된 모드가 부정확하다고 판단되는 경우에는 모든 모드에 대해서 RDO를 수행하여 4×4 블록의 모드 선택 정확도를 높이는 적응

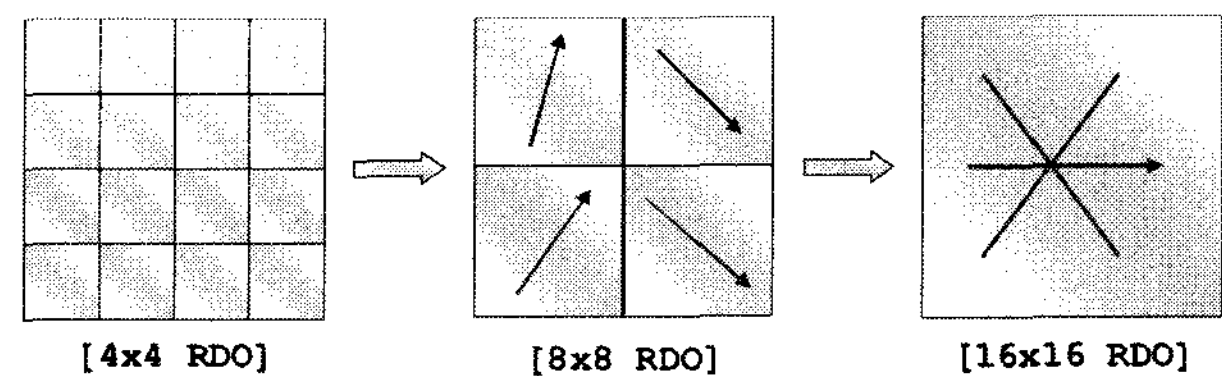


그림 5. 잘못된 모드 선택에 의한 에러 전파

Fig. 5. Error propagation caused by uncorrect mode decision

표 2. 제안 알고리즘의 순서도

Table 2. Detailed steps of the proposed algorithm

Step 1	소벨 연산자를 이용하여 프레임 단위의 Edge Map을 생성하고, 휘도 4×4 블록과 색차 8×8 블록의 DE를 구한다.
Step 2	I4MB의 경우 DE와 상응되는 화면내 모드와 DC_Mode에 대해서, Chroma_8x8의 경우 U와 V 두 요소의 DE에 상응하는 화면내 모드와 DC_Mode에 대해서 각각 RDO를 수행한다.
Step 3	8×8 블록의 DE를 그 블록을 구성하는 4개의 4×4 블록 DE들로부터 예측하고, DE에 상응하는 화면내 모드와 DC_Mode에 대해서 RDO를 수행한다.
Step 4	STEP 3와 유사하게, 16×16 블록의 DE는 그 블록을 구성하는 4개의 8×8 블록 DE들로부터 예측되고, DE에 상응하는 화면내 모드와 DC_Mode에 대해서 RDO를 수행한다.

알고리즘이 필요하다. 본 논문에서는 4x4 블록들의 평균 RD_Cost를 이용한 적응 알고리즘을 제안한다. 즉, 4x4 블록의 평균 RD_Cost를 Avr_4x4_Rdcost 로 정의하고 부호화를 진행하면서 다음과 같이 갱신해 나간다.

$$Avr_4 \times 4_Rdcost = \frac{Cur_4 \times 4_Rdcost + n \times Avr_4 \times 4_Rdcost}{n + 1} \quad (5)$$

Cur_4x4_Rdcost 는 현재 4x4 블록의 RD_cost이고 n 은 현재까지 부호화된 4x4블록의 총 개수를 의미한다. 위와 같이 Avr_4x4_Rdcost 를 계산하고 이 값을 이용하여 임계 값을 정의한 후, 임계 값을 넘지 않는 4x4 블록에 대해서는 제안한 알고리즘을 적용하고, 임계 값을 넘는 4x4 블록에 대해서는 고속 결정 방식에 의해 선택된 모드가 최적의 모드가 아니라고 판단하여 모든 모드에 대해서 RDO를 수행한다. 이때, 임계 값을 다양한 실험을 통해서 다음과 같이 결정하였다.

```

Thd = Avr_4x4_Rdcost + 600
If (Cur_4x4_Rdcost < Thd)
    Perform Proposed algorithm
Else
    Perform Conventional RDO
    
```

3. 매크로블록의 Homogeneous 특성에 의한 Skip 알고리즘

일반적으로, 평활한 (Homogeneous) 영역에 위치한 매크로블록들은 큰 블록 크기의 화면내 예측 모드에 의해서 부호화 될 확률이 크다^[7]. 따라서 매크로블록의 이러한 특성 정보를 알고 있다면, 기존 D. Wu의 방법을 이용하여^[8] 평활한 블록들은 I16MB 모드로만 RDO를 수행하고, I8MB와 I4MB는 skip함으로써 부호화기의 복잡도를 줄일 수 있다. 앞서 4x4 블록의 DE를 결정해 주기 위해 프레임 단위로 Edge map을 구해주었기 때문에, 추가적인 연산 없이 이 정보를 이용하여 매크로블록의 Homogeneous 여부를 다음과 같이 결정한다.

$$Flag_H = \begin{cases} 1, & \sum_{i=0}^M \sum_{j=0}^M Mag(p_{i,j}) \leq Th_H \\ 0, & \sum_{i=0}^M \sum_{j=0}^M Mag(p_{i,j}) > Th_H \end{cases} \quad (6)$$

M 은 매크로블록의 크기(=16), $Mag(p_{i,j})$ 는 픽셀 $p_{i,j}$ 위치에서의 소벨 연산자의 Magnitude 값을 의미한다. 위 식에 의하여, 만약 매크로블록 모든 픽셀에서의 소벨 연산자 Magnitude의 합이 임계값보다 작으면 해당 매크로블록을 Homogeneous 하다고 판단한다. 임계값은 다양한 환경에서의 실험으로부터 20,000으로 정해주었다^[8].

IV. 실험 결과

본 논문에서 제안한 알고리즘은 JVT 참조 소프트웨어 (Reference Software)인 JM 12.2^[9]를 기반으로 구현되었고, 실험은 Intel Pentium Dual Processor 3.2GHz / 2GB DDR2 RAM 사양을 갖는 PC 환경에서 진행하였다. 실험 환경으로는 RDO를 활성화하였고 엔트로피 부호화는 CABAC을 이용하였으며, 제안한 4x4 블록의 적응 알고리즘을 적용하였다. 실험에 쓰인 영상은 QCIF (176x144) 해상도의 container, news, coastguard와 CIF (352x288) 해상도의 flower, waterfall, tempete 영상을 이용하였으며, 각 영상당 부호화된 프레임은 100장으로 모두 화면내 부호화 (I-frame) 로 압축하였다. QP는 총 4가지 (20, 24, 28, 32)에 대해서 실험

표 3. 실험 환경

Table 3. Simulation Conditions.

Reference Software		JM 12.2
Test Sequences	QCIF	container, news, coastguard
	CIF	flower, waterfall, tempete
Number of Frames (ALL I-Frames)		100
Quantization Parameters		20, 24, 28, 32
Variable Length Coding		CABAC
Proposed "Inverse Tree-Structure Fast Intra Algorithm"	Luma_4x4 loop	Dominant Edge+DC_Mode
	Luma_8x8 loop	Dominant Edge+DC_Mode
	Luma_16x16 loop	Dominant Edge+DC_Mode
	Chroma_8x8 loop	Dominant_U_Edge+Dominant_V_Edge+DC_Mode

하였고, 제안한 알고리즘의 결과를 Pan^[3]의 방법과 비교 분석하였다. 구체적인 실험 환경은 표 3에 명시되어 있다. 성능 비교를 위하여 부호화 시간 감소량(Time Saving)과 비트 증가량(Bits Increasing)을 다음과 같이 정의하였다.

$$Time\ Saving(\%) = \frac{Process\ Time(Proposed) - Process\ Time(JM)}{Process\ Time(JM)} \quad (7)$$

$$Bits\ Increasing(\%) = \frac{Total\ Bits(Proposed) - Total\ Bits(JM)}{Total\ Bits(JM)} \quad (8)$$

제안한 역 트리 구조 알고리즘과 JM12.2의 압축 성능을 비교한 실험 결과가 표 4에 나타나 있다. 표 4를 분석하면,

표 4. 역 트리 구조의 화면내 모드 결정 기법
Table 4. Results for Inverse Tree-structure Algorithm

Sequence	QP	ΔPSNR(dB)	ΔBits(%)	ΔTime(%)	
Q C I F	container	24	-0.21	1.65	-62.9
		28	-0.10	1.65	-57.4
		32	-0.05	1.61	-46.1
		36	-0.01	1.95	-44.8
		Avr.	-0.09	1.71	-52.8
	news	24	-0.19	3.26	-62.3
		28	0.11	3.24	-53.7
		32	-0.07	3.45	-45.4
		36	-0.03	3.10	-44.5
		Avr.	-0.10	3.26	-51.5
	coastguard	24	-0.20	1.61	-62.2
		28	-0.09	1.91	-56.6
32		-0.09	2.39	-52.2	
36		-0.03	3.25	-47.3	
Avr.		-0.10	2.29	-54.6	
C I F	flower	24	-0.36	0.84	-61.7
		28	-0.28	0.58	-55.3
		32	-0.08	0.74	-34.4
		36	-0.03	1.43	-29.4
		Avr.	-0.19	0.90	-45.3
	waterfall	24	-0.22	1.09	-62.3
		28	-0.21	1.09	-60.5
		32	-0.21	0.99	-46.3
		36	-0.19	0.76	-40.1
		Avr.	-0.21	0.99	-56.6
	tempete	24	-0.30	0.73	-65.3
		28	-0.24	1.19	-63.5
32		-0.13	1.11	-49.3	
36		-0.05	1.87	-42.2	
Avr.		-0.18	1.22	-55.1	

역 트리 구조 기법을 이용한 방법이 참조 소프트웨어에 비해 평균 0.14dB의 PSNR 감소와 1.7%의 비트량 증가를 가져오지만 부호화 시간은 JM12.2에 비해 평균 53% 줄어드는 것을 알 수 있다.

다음으로 역 트리 구조 고속 모드결정 알고리즘과 매크로 블록의 homogeneous 특성을 이용하는 skip 알고리즘을 같이 적용하여 실험하였고, 기존 고속모드 기법과의 성능 비교를 위하여 Pan의 방법^[3]과 비교 분석하였다. 실험 결과

표 5. 역 트리 구조의 화면내 모드 결정 기법 + 매크로 블록의 Homogeneous 특성을 이용한 Skip 알고리즘

Table 5. Results for Inverse Tree-Structure Algorithm +Skip mode Algorithm by MB's Homogeneity

Sequence	QP	ΔPSNR(dB)		ΔBits(%)		ΔTime(%)		
		Pan	Proposed	Pan	Proposed	Pan	Proposed	
Q C I F	container	20	-0.24	-0.04	3.21	4.56	-53.7	-69.7
		24	-0.27	-0.03	3.44	3.97	-53.9	-65.4
		28	-0.25	-0.02	3.84	3.21	-55.0	-60.8
		32	-0.20	-0.02	4.22	2.53	-54.0	-55.8
		Avr.	-0.24	-0.03	3.71	3.56	-54.2	-62.9
	news	20	-0.29	-0.12	3.84	4.84	-53.4	-69.8
		24	-0.28	-0.07	3.88	4.61	-52.9	-65.3
		28	-0.31	-0.03	3.94	5.30	-55.5	-59.0
		32	-0.29	-0.01	4.24	4.77	-55.3	-55.1
		Avr.	-0.29	-0.06	3.98	4.88	-54.2	-62.3
	coastguard	20	-0.15	-0.01	3.10	3.47	-55.7	-69.6
		24	-0.16	-0.03	3.27	3.72	-54.1	-66.9
28		-0.18	-0.04	3.33	4.33	-54.0	-62.5	
32		-0.21	-0.01	3.54	3.84	-54.2	-58.7	
Avr.		-0.18	-0.02	3.31	3.84	-54.5	-64.4	
C I F	flower	20	-0.41	-0.32	1.58	2.05	-57.7	-68.6
		24	-0.40	-0.33	1.64	1.94	-58.1	-68.0
		28	-0.37	-0.22	1.68	1.83	-57.3	-62.8
		32	-0.35	-0.14	2.11	1.43	-57.3	-52.2
		Avr.	-0.38	-0.25	1.75	1.81	-57.7	-62.9
	waterfall	20	-0.38	-0.29	2.66	2.86	-58.0	-68.9
		24	-0.36	-0.38	2.04	1.69	-58.1	-69.1
		28	-0.31	-0.28	2.36	2.08	-58.6	-66.6
		32	-0.37	-0.21	2.81	2.53	-57.1	-60.3
		Avr.	-0.34	-0.29	2.46	2.29	-57.1	-66.2
	tempete	20	-0.34	-0.29	2.42	2.41	-58.4	-69.2
		24	-0.33	-0.25	2.74	2.90	-57.1	-67.9
28		-0.30	-0.16	2.24	3.29	-57.8	-61.4	
32		-0.29	-0.08	3.07	4.05	-57.5	-60.5	
Avr.		-0.32	-0.19	2.61	3.16	-58.0	-64.8	

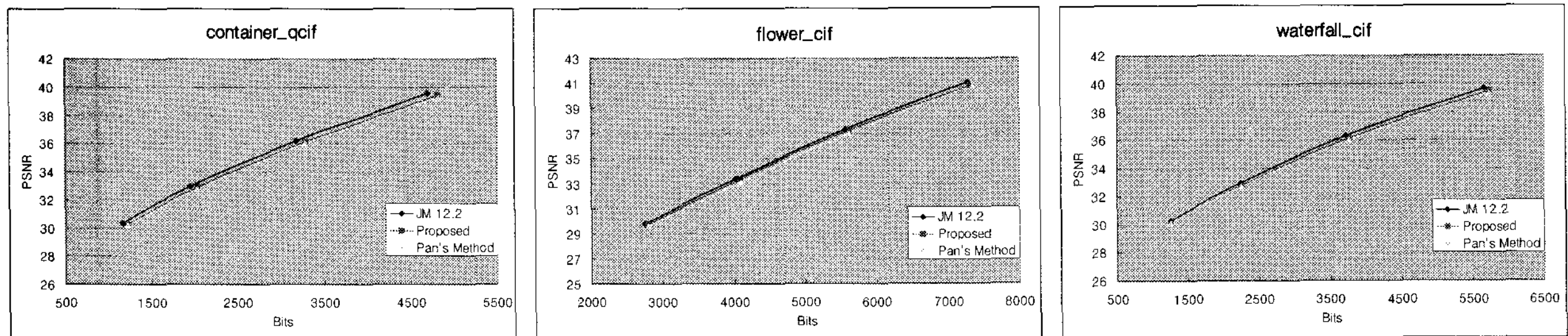


그림 6. 실험 영상들의 R-D 곡선
Fig. 6. R-D Curves of Test Sequences

는 표 5에 정리되어 있다. 표를 통해서 제안된 두 개의 기법을 동시에 사용했을 경우의 부호화 시간 감소량은, JM 12.2와 비교했을 때 평균 64% 줄어들고, 역 트리 구조 기법 하나만 이용했을 때와 비교하면 평균 11% 더 감소했음을 알 수 있다. 반면에 비트 증가량은 평균 3.3%로 앞의 실험보다 다소 증가했는데, 이것은 I4MB와 I8MB를 skip할 경우 최적의 모드가 선택되지 않는 경우가 발생되기 때문이고, 따라서 skip 알고리즘 효과는 각 영상의 homogeneous 특성에 영향을 받는 것을 알 수 있다. 제안한 방법의 결과를 기존 Pan 방법^[3]의 결과와 비교하면 비트 증가량은 거의 비슷하지만, PSNR 감소량을 비교하면 Pan의 방법이 평균 0.3dB, 제안한 방법이 평균 0.14dB로 제안한 방법으로 복원된 영상의 화질 열화가 더 적은 것을 알 수 있다. 부호화 시간 감소 측면에서는 제안한 방법이 Pan의 방법보다 5% (flower)에서 10% (coastguard)까지 더 우수한 결과를 보인다. 3개 실험 영상에 대한 R-D 곡선들을 그림 6에 나타내었다.

V. 결론

본 논문은 H.264/AVC가 화면내 예측에서 지원하는, 세 가지 다른 크기 블록들 간에 존재하는 계층적인 상관성을 이용하는 고속 모드 결정 기법을 제안하였다. 화면내 예측은 인접 블록의 픽셀 값을 이용하여 방향성을 갖는 예측을 하는데 하위블록과 상위블록에서 결정되는 모드들의 방향성은 서로 큰 상관도를 갖기 때문에 이러한 경향성을 이용하여 화면내 모드 결정을 위한 부호화 시간을 단축시킬 수

있었다. 특히, 제안된 역 트리 구조 예측 기법의 안정성을 높이기 위한 적응 알고리즘을 추가하였고, 부호화기 복잡도를 조금 더 줄이기 위해서 매크로블록의 Homogeneous 특성을 이용한 Skip 알고리즘이 함께 사용되었다. 실험 결과 제안한 방법은 JM 12.2와 비교하여 큰 PSNR 감소 및 비트량 증가 없이, 평균 64%의 부호화 시간 감소율을 나타내었다. 또한 기존 Pan의 방법^[3]과 비교해 보았을 때도 5~10%의 성능 개선을 보였다. 또한, 본 논문에서는 화면내 예측에서의 고속 모드 기법만 고려했지만 제안된 알고리즘에서 사용한 DE 정보를 이용하면 매크로블록의 수평/수직 방향성을 판단할 수 있고, 이를 통하여 16x8 또는 8x16 블록 모드를 제외시킴으로써 화면간 예측에서도 후보 모드의 수를 줄임으로써 부호화 시간을 더욱 단축시킬 수 있을 것으로 예상된다.

참고 문헌

- [1] Report of The Formal Verification Tests on AVC, Dec. 2003. ISO/IEC 14 496-10 ITU-T Rec. H.264 MPEG2003/N6231.
- [2] Jeyun Lee, and Byeungwoo Jean, "Fast mode decision for H.264", IEEE International Conference on Multimedia and Expo, 2005
- [3] Feng Pan, Xiao Lin, Susanto Rahardja, Keng Pang Lim, Z. G. Li, Dajun Wu, and Si Wu, "Fast Mode Decision Algorithm for Intraprediction in H.264/AVC Video Coding", IEEE transactions on Circuits and Systems for Video Ttechnology, vol. 15, no. 7, July 2005.
- [4] C.-C. Cheng and T.-S. Chang, "Fast three step intra prediction algorithm for 4 4 blocks in H.264," Proc. IEEE ISCAS, vol. 2, pp. 1509 - -1512, 2005
- [5] D.-G. Sim and Y. Kim, ""Context-adaptive mode decision for intrablock coding in H.264/MPEG-4 part 10,"" Real-Time Imaging,

vol. 11, pp. 1 - -6, 2005.

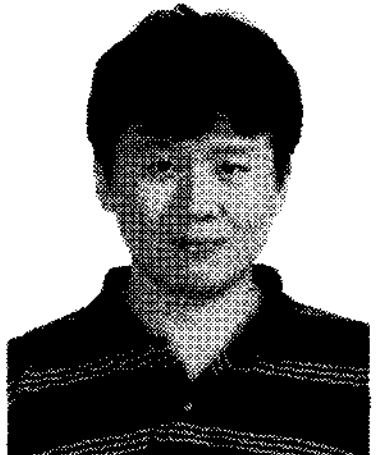
- [6] ITU-T Rec. & ISO/IEC 14496-10 AVC, ""Advanced Video Coding for Generic Audiovisual Services,"" version 3, 2005.
- [7] Andy C. Yu a, Ngan King Ngi b, and Graham R. Martin, "Efficient intra- and inter-mode selection algorithms for H.264/AVC", Journal of visual communication & image representation, 2006.
- [8] D. Wu, F. Pan, K. P. Lim, S. Wu, Z. G. Li, X. Lin, S. Rahardja, and C. C. Ko, "Fast Intermode Decision in H.264/AVC Video Coding", IEEE transactions on circuits and systems for video technology, vol.15, no. 6, July 2005.
- [9] JVT Reference Software unofficial version JM12.2, <http://iphome.hhi.de/suehring/tml/download/>

저 자 소 개



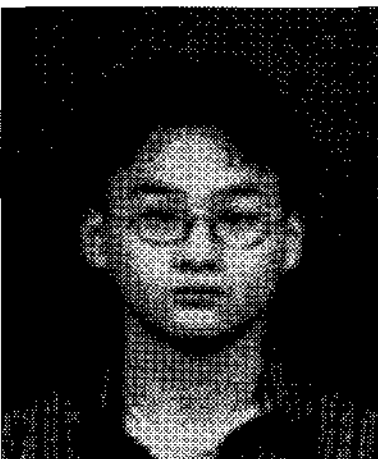
고 현 석

- 2006년 8월 : 연세대학교 전기전자공학부 학사
- 2006년 9월 ~ 현재 : 연세대학교 전기전자공학과 석사과정
- 주관심분야 : 영상 신호처리, H.264/AVC



유 기 원

- 1998년 2월 : 고려대학교 전자공학과 학사
- 2000년 2월 : 한국정보통신대학원 공학 석사
- 2000년 2월 ~ 현재 : 삼성전자 DM연구소
- 2007년 3월 ~ 현재 : 연세대학교 전기전자공학부 박사과정
- 주관심분야 : 신호처리, 동영상처리



서 정 동

- 2004년 2월 : 연세대학교 전기전자공학부 학사
- 2006년 2월 : 연세대학교 전기전자공학과 석사
- 2006년 3월 ~ 현재 : 연세대학교 전기전자공학부 박사과정
- 주관심분야 : 3차원 동영상 부호화/복호화, MPEG-2/4



손 광 훈

- 1983년 : 연세대학교 전자공학과 공학사
- 1985년 : University of Minnesota, MSSE
- 1991년 : North Carolina State University, Ph.D
- 1993년 : 한국전자통신연구원 선임연구원
- 1994년 : Georgetown University, Post-doctor fellow
- 1995년 ~ 현재 : 연세대학교 전기전자공학부 교수
- 2002년 9월 ~ 2003년 8월 : Nanyang Technological University, Visiting Professor
- 주관심분야 : 3차원 영상처리, 영상 통신