효율적인 블록 스킵 기술들을 이용한 H.264에서의 고속 모드 결정 알고리즘

준회원 조 영 섭*, 정회원 정 제 창*

Fast Mode Decision Algorithm Using Efficient Block Skip Techniques for H.264 P Slices

Youngsub Jo* *Associate Member*, Jechang Jeong* *Regular Member*

요 약

본 논문에서 우리는 H.264의 인터모드 결정에 대한 복잡도를 줄일 수 있는 고속 알고리즘을 제안한다. 주된 아이디어는 두 가지 기술들로 구성된다. 첫 번째 기술은 모드 결정과정을 조기에 끝내는 기술이다. 우리는 여기서 스킵 모드와 16x16 모드에 초점을 맞춘다. 왜냐하면 이 두 모드가 대부분의 시퀀스에서 최적 모드가 될 확률이 가장 크기 때문이다. 두 번째 기술은 불필요한 8x8 모드를 스킵하는 기술이다. 8x8 모드를 계산하는데 걸리는 시간은 상당히 크다. 그러므로 만약 우리가 불필요한 8x8 모드의 계산을 제거할 수 있다면, 전체 인코딩 과정에서 많은 양의 시간이 절약 될 수 있다. 실험 결과는 제안한 알고리즘이 PSNR 손실이 거의 없이 43% 가량의 시간을 절약할 수 있음을 보여주었다. 또한 코딩된 전체 비트수의 증가도 별로 크지 않았다.

Key Words : H.264, Fast Algorithm, Mode Decision, RD Optimization, Video Coding

ABSTRACT

In this paper, we propose a fast algorithm that can reduce the complexity for inter mode decision of the H.264 encoder. The main idea consists of two techniques. The first one is the technique early terminating mode decision process. We focused on the skip and 16x16 mode because these modes occupies the largest portion in most of sequences. The second one is the technique skipping unnecessary 8x8 modes. The time consumption caused by the 8x8 mode is very considerable. Therefore if we can extract the unnecessary 8x8 mode calculation well, a large amount of time can be saved in total encoding process. The experimental results show that the proposed algorithm can achieve up to 43% speed up ratio with insignificant PSNR loss. The increase of total bits encoded is also not noticeable.

Ⅰ. 서 론

최근 디지털 비디오의 압축 전송 및 부호화 기술은 방송, 인터넷 등의 통신 기술의 발달에 따라 급속도로 진보하고 있다. 특히, 디지털 동영상의 압축 또는 부호화 분야는 다양한 스펙이 제안되고 있으며, 또한 성능을 개선하기 위한 다양한 노력이 진행되고 있다. 그러한 결과물로써 나온 최신의 비디오 압축 기술이 H.264이다. H.264는 코딩 효율면에서 이전의 비디오 압축 표준들에 비해 월등히 뛰어난 성능을 보여준다. 이전의 기술들이 일정한 크기로 매크로블록(Macroblock)을 코딩했다면, H.264는 4x4

※ 본 연구는 서울시 산학연 협력사업(10570)으로 구축된 서울 미래형 콘텐츠 컨버전스 클러스터 지원으로 수행되었습니다.
* 한양대학교 전자통신컴퓨터공학과 영상통신 및 신호처리 연구실(longglittz@naver.com)
논문번호 : KICS2009-10-451, 접수일자 : 2009년 10월 8일, 최종논문접수일자 : 2010년 1월 25일

부터 16x16까지 다양한 크기의 블록들을 이용하여 코딩함으로써 코딩효율면에서 상당한 이득을 얻게 되었다. 이와 더불어, 쿼터펠(quarter-pel)까지의 움직임 벡터 탐색(motion vector search), 다양한 예측방향을 가지는 인트라 모드 결정법(intra mode decision), 적응적인 디블록킹 필터(adaptive deblocking filter) 등과 같은 발전된 코딩 방법들도 H.264의 성능 향상에 기여한 중요한 기술들이라 할 수 있다 [1].

그러나 이러한 많은 뛰어난 기술들 때문에 H.264는 높은 계산상의 복잡도를 필요로 하게 된다. 특히, 모드 결정 과정(mode decision process)의 복잡도는 실제 구현하는데 있어서 매우 많은 노력을 필요로하게 한다. 모드 결정과정은 인트라 모드 결정과, 인터 모드(inter mode) 결정 과정으로 나뉜다. 인트라 모드 결정과정에서 우리는 16x16 루미넌스(luminance) 블록에서의 4가지 모드 그리고 4x4 루미넌스 블록에서의 9가지 모드들을 조사해서 최적의 모드를 찾게 된다. 인터 모드 결정에서는 최고의 성능을 보여주는 블록 크기를 찾기 위해서 하나의 매크로블록이 여러 가지 크기의 블록들로 나누어지게 된다. 이러한 여러 블록 크기들은 인터 모드 결정에서 크게 5가지 모드로써 정의 되어진다. 스킵 모드(skip mode), 16x16 모드, 16x8 모드, 8x16 모드, 그리고 8x8 모드가 그것이다. 여기서 8x8 모드는 다시 4가지 서브모드(sub-mode)로 나누어지는데 그것은 8x8, 8x4, 그리고 4x4 서브 모드이다. 그림 1이 여러 크기의 모드들에 대해서 자세히 묘사하고 있다.

H.264 인코더에서는 이러한 여러 블록 크기를

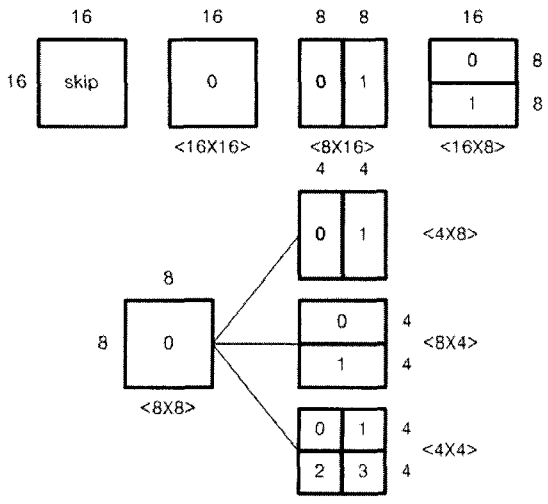


그림 1. 블록 크기에 따른 모드들

갖는 모드들 중에서 최적의 모드를 찾는 방법으로써 보통 Rate Distortion Optimization(RDO) 방법을 사용한다 [2]. RDO 방법은 비트율(bit rate)이나 화질면에서 최적의 결과를 가져오지만 엄청난 복잡도를 필요로 한다. RDO 방법을 이용하여 모드 결정을 할 때, 각각의 모드에 대한 Rate Distortion Cost(RDC)가 구해져야 한다. 그리고 최소의 RDC를 갖는 모드가 최적의 모드로 선택이 된다. 이것을 나타내는 것이 식 (1)로 표현된다.

$$J(M|QP,\lambda_M)=D(M|QP)+\lambda_M R(M|QP) \quad (1)$$

위 식에서 J는 RDC를 나타내고, M과 D는 모드와 왜곡(Distortion)을 나타낸다. R과 λ는 각각 비트율과 라그랑지 계수(Lagrange multiplier)를 나타낸다. 인코더의 모드 결정에 대한 높은 계산상 복잡도를 효과적으로 줄이기 위해서, 많은 알고리즘들이 제안되었다. [3]에서는 고속 인트라 모드 결정을 위해서 에지맵(edge map)을 이용한 알고리즘을 제안하였는데, [4]에서는 모드결정의 조기 종료(early termination)을 위해 이 에지맵 기술을 이용하여 동일성 검사(homogeneity detection)를 제안하였으며 더불어 SAD를 이용한 시간적 유사성 검사를 제안하였다. 비슷하게, [5]에서도 에지맵을 이용하였는데 이들은 동일성 검사를 위해 8x8 블록에 모드 히스토그램(mode histogram)을 적용시켰다. 또한 [5]의 조기 종료 방법은 시공간적으로 이웃해있는 매크로블록들의 RDC와 모드 히스토그램을 이용했다. [6]은 모드 결정 과정을 여러 레벨로 나누어서 조기 종료를 시도하는 계층적 구조를 제안했으며, [7]은 움직임 벡터와 현재블록과 예측블록 사이의 코릴레이션 비율(correlation ratio)을 사용하는 매크로블록 추적 기법을 이용하는 모드 결정 과정 조기 종료 방법을 제안하였다.

본 논문에서는, 첫 번째로 스킵 모드와 16x16 모드에 대해 최적모드 인지 아닌지를 조기에 판단하여 모드 결정을 빨리 종료하는 방법을 제안하고 두 번째로 다른 모드의 RDC를 이용하여 8x8 모드를 스킵하는 방법을 제안한다. 첫 번째 방법은 두가지 세부방법으로 나뉘게 되는데, 다이나믹 임계값(dynamic threshold)을 이용하는 방법과 모드들의 RDC를 비교하는 방법으로 구분된다. 다이나믹 임계값은 현재 매크로블록의 위치가 이동함에 따라 달라지는 주변 매크로블록들의 RDC 값에 의존하게 된다. 실험 결과는 제안된 알고리즘이 적용되면 PSNR 이나 비트

율에서의 성능저하가 거의 없이 평균 42%의 인코딩 시간을 절약할 수 있다는 것을 보여주고 있다.

이 논문의 구성은 다음과 같다. 2장에서는 다이나믹 임계값과 모드들 사이의 RDC의 관계를 이용한, 스킵 모드와 16x16 모드에 대한 조기 종료 방법을 설명하고, 3장에서는 8x8 모드의 스킵을 위해서 모드들간 RDC를 이용하는 방법을 보여준다. 4장과 5장에서는 각각 실험 결과와 결론을 기술한다.

Ⅱ. 스킵모드와 16x16 모드에 대한 조기종료 알고리즘

2.1 다이나믹 임계값을 이용하는 첫 번째 방법

이 단계에서, 우리는 현재 매크로블록(MB)의 이웃한 블록들의 최적모드가 무엇인지 조사하고 그들의 RDC 값을 이용하기로 한다. 우리는 오직 현재 MB으로부터 공간적으로 가장 가까운 4개의 MB(왼쪽, 왼쪽 위, 위쪽, 오른쪽 위)의 정보만 이용할 것이다. 이들 네 개의 MB들은 이미 인코딩 된 상태이며 우리는 그들의 최적모드와 RDC 값들을 알고 있다. 이 정보를 이용해서 우리는 스킵모드와 16x16 모드(모드01)가 최적 모드인지 아닌지를 조기에 검출할 수 있는 RDC 임계값을 예측한다. 이번 단계에서 사용되는 현재 MB으로부터의 이웃한 네 개의 MB의 위치는 그림 2에서 보여진다.

그림 2에서 볼 수 있는 것처럼, 현재 프레임의 경계부분에 위치한 MB들에는 이번 단계의 알고리즘을 적용시키지 않는다. 왜냐하면, 이 MB들은 현재 MB의 모드01을 검출하기에 충분한 정보를 가지고 있지 않기 때문이다. 만일 이들 MB들이 사용된다면, 인코딩 시간을 거의 줄이지 못할뿐더러 인코더의 성능까지 감소시킬 것이다. 그러므로 우리는 이 단계에서 성능의 감소를 피하기 위해 경계 MB들은 고려하지 않을 것이다. 모드01의 검출을 위한 조건은 식 (2)에서 보여진다.

$$\begin{aligned} &\textit{If } \{(RDC_L < RDC_H)\&(RDC_0 < RDC_L)\&(RDC_1 < RDC_L)\} \\ &\quad \textit{Then} \text{ best mode is MD01 and early terminated} \\ &\textit{Otherwise} \text{ jump to the next step} \end{aligned} \tag{2}$$

이웃한 블록들의 RDC 값들이 식 (2)에서 사용된다. 식(2)에서는 스킵모드 (모드0)나 16x16 모드(모드1)을 최적 모드로 가지는 이웃MB 들만이 이용될 수 있다. 첫 번째로, 우리는 RDC_L과 RDC_H를 찾는다. RDC_L은 최적 모드가 모드0 이거나 모드1인 MB들의 RDC의 평균값이다. RDC_H는 최적모드가 모드01이 아닌 MB들의 RDC의 평균값을 나타낸다. 두 번째로 RDC_L과 RDC_H를 비교한다. RDC_L과 RDC_H 사이의 비교후에, 만약 RDC_L이 RDC_H보다 작다면 RDC_L을 다시 RDC_0와 RDC_1에 대해 비교를 한다. 여기서 RDC_0은 모드0의 RDC값을 의미하고, RDC_1은 모드1의 RDC값을 의미한다. 이 마지막 비교에서 우리의 목적은 RDC_L보다 작은 RDC_0 혹은 RDC_1을 찾는 것이다.

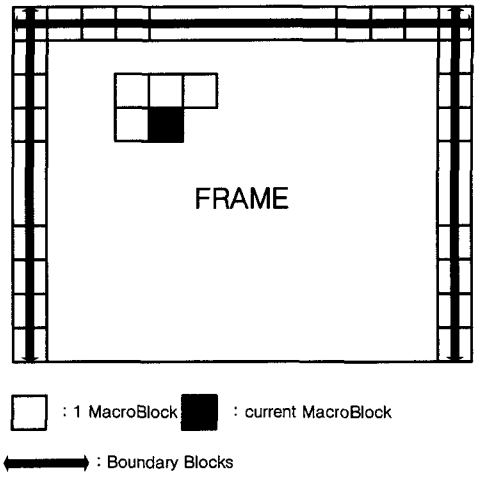


그림 2. 경계 MB과 현재 MB의 이웃한 4개의 MB

만일 식 (2)가 만족을 한다면, 우리는 모드 결정 과정을 조기 종료할 수 있다. 이 조기 종료의 경우, 만일 RDC_0이 RDC_1보다 작으면, 모드0이 최종적인 최적모드이고, 그 반대이면 모드1이 최적모드가 된다.

2.2 모드들 사이의 관계를 이용한 두 번째 방법

비록 모드01을 최적모드로 가지는 많은 MB들이 첫 번째 방법을 통해 조기종료 되었지만, 모드01을 최적 모드로 가지는 MB이 아직도 많이 남아있다. 이번 두 번째 방법에서 우리는 모드들 간의 관계를 사용하여 모드01을 검출할 수 있는 효과적인 비교 방법을 제안한다. 모드01과의 비교방법에 사용될 타겟 모드는 많은 실험들에 의해 결정 될 수 있다. 계산상의 복잡도를 피하기위해서 우리는 모드01을 제외하고 오직 한 개의 타겟모드 만을 추가할 것이다. 타겟 모드의 후보자로는 16x8 모드(모드2), 8x16 모드(모드3), 그리고 8x8 모드(모드8) 이렇게 3개를 고려한다. 구체적인 통계적 자료들이 표1에서 보여진다.

히트율 (Hit Ratio, HR)과 정확성 (AR)의 평균값이 타겟모드 결정의 파라미터로써 사용된다. 그들에

대한 방정식은 식 (3)과 (4)에서 보여진다.

$$HR = \frac{Num.\ of\ Hit\ MB}{Total\ num.\ of\ MB} \times 100(\%) \quad (3)$$

$$AR = \frac{Num.\ of\ Hit\ MB}{Num.\ of\ Hit\ MB + Miss\ MB} \times 100(\%) \quad (4)$$

Total number of MB 은 어떤 sequence에서 최적모드가 모드01인 MB의 전체개수를 나타내고, Hit MB은 제안된 알고리즘을 사용해서 최적모드가 모드01인 MB를 정확하게 찾은 MB을 뜻한다. 그리고 Miss MB는 실제로는 최적모드가 모드01이 아닌데, 우리가 잘못 예측을 한 MB을 의미한다.

표 1에서 볼 때 가장 알맞은 타겟 모드는 모드8로 고려되어진다. 하지만 모드8의 계산복잡도는 모드2나 모드3에 비해 월등히 높다. 그러므로, 만약 우리가 모드8을 타겟 모드로 사용한다면, 우리는 인코딩 시간 감소 측면에서 약간의 이득만을 얻을 수 있을 것이다. 표 1에서 볼 때 모드들간 HR과 AR의 평균값의 차이는 별로 크지 않기 때문에 어떤 모드를 가장 적합한 타겟모드로 할 것인지는 표 2에서 보여지는 좀 더 자세한 실험을 통해 결정될 수 있을 것이다.

HR, AR, PSNR, 그리고 비트율 에 대한 실험으로부터, 우리는 HR과 AR의 높은 값이 PSNR과 비트율 측면에서 더 좋은 성능을 의미한다는 것을 알고 있다. 제안된 알고리즘에서는 PSNR과 비트율에 대한 성능뿐만 아니라 인코딩 시간 감소율 또한 매우 중요한 파라미터이기 때문, 궁극적으로 우리가 가장 알맞다고 생각하는 타겟 모드는 더 좋은 성능 그리고 더 많은 시간감소율을 가지는 모드가 될 것

표 1. 모드들 사이의 관계(CIF, QP=28, 100프레임)

Sequences	모드2		모드3		모드8	
	HR	AR	HR	AR	HR	AR
Foreman	60	92	61	93	68	92
Akiyo	90	99	91	99	95	98
Coastguard	35	91	35	91	38	92
Stefan	54	93	53	93	57	94
Mobile	27	87	27	88	27	94
Container	85	97	85	97	90	96
Tempete	43	89	43	88	45	91
Bus	43	91	43	91	46	92
Average	55	92	55	92	58	93

표 2. 각 타겟 모드에 대한 성능변화 실험(CIF, QP=28, 100프레임)

Target	Sequences	ΔPSNR (dB)	ΔBR (%)	ΔTime(%)
모드2	Foreman	-0.07	-0.50	-21.34
	Akiyo	-0.05	-0.71	-43.50
	Coastguard	-0.02	-0.24	-10.08
	Stefan	-0.05	0.17	-16.74
	Mobile	-0.04	-0.24	-7.01
	Container	-0.03	-0.04	-43.36
	Tempete	-0.05	-0.43	-13.35
	Bus	-0.02	0.12	-14.70
	Average	-0.04	-0.23	-21.26
모드3	Foreman	-0.05	-0.59	-23.07
	Akiyo	-0.04	-0.72	-43.54
	Coastguard	-0.02	-0.06	-12.60
	Stefan	-0.06	0.07	-18.68
	Mobile	-0.05	-0.25	-10.02
	Container	-0.04	-0.63	-43.50
	Tempete	-0.04	-0.34	-16.03
	Bus	-0.03	0.06	-16.88
	Average	-0.04	-0.31	-23.04
모드8	Foreman	-0.04	-0.38	-17.99
	Akiyo	-0.04	-0.95	-32.50
	Coastguard	-0.03	-0.19	-10.02
	Stefan	-0.05	0.24	-14.51
	Mobile	-0.02	0.05	-7.51
	Container	-0.04	0.07	-32.43
	Tempete	-0.04	-0.16	-12.54
	Bus	-0.03	-0.01	-13.84
	Average	-0.04	-0.17	-17.67

이다. 따라서, HR, AR 그리고 ΔTime 의 세 개의 파라미터를 이용하여 세 개의 후보 모드중 최적 모드가 무엇인지 결정할 수 있다. 표 1에서 모드2와 모드3의 HR과 AR은 거의 비슷하지만 모드8에 대한 값은 상대적으로 조금 더 뛰어난 값을 보여준다. 표 2에서는 세 개의 타겟 모드의 PSNR과 비트율의 변화량이 매우 비슷하다. 하지만 시간 감소율에서 두드러진 차이점을 보여주고 있다. 결과적으로 식 (5)에 기반하여 모드3이 최적의 타겟 모드로 선택이 되었다.

$$m_k = \underset{m=\{m_2,m_3,m_8\}}{\text{argmax}}\ \text{MP}(HR, AR, T)$$
$$\text{MP} = \alpha(HR + AR) + (1-\alpha)T \quad (5)$$

식 (5)에서, MP는 최대 성능을 나타낸다. T는 인코딩 시간 감소율을 나타내고, α는 실험상의 경험적 계수이다. 최종적으로 우리는 최적의 모드01에 대한 초기 검출로 모드결정 조기종료를 할수 있는 두 번째 방법으로써, 모드01과 모드3의 관계를 이용한 방정식을 식 (6)과 같이 도출 해 낼 수 있다.

$$If\ \{(RDC_0 < RDC_3)\&(RDC_1 < RDC_3)\}$$
$$Then\ \text{best mode is MD01 and early terminated} \quad (6)$$
$$Otherwise\ \text{jump to the next step}$$

Ⅲ. 모드8 조기 스킵(early skip) 알고리즘

모드 결정에 있어서 모드8이 최적모드가 될 확률은 그리 크지 않다. 심지어 어떤 시퀀스에서는 그 확률이 매우 낮다. 표 3은 7개의 시퀀스들에서 최적 모드로써 모드8에 대한 비율의 통계적 수치를 보여준다. 표 3에서 Ratio는 전체 MB의 개수에 대한 최적모드로써의 모드8의 개수의 비율을 나타낸다. Mobile, Tempete, 그리고 Stefan처럼 많은 움직임 정보를 포함하고 있는 시퀀스들은 상대적으로 높은 Ratio를 가진다. 이것은 모드결정에 있어서 모드8의 역할이 상당히 중요함을 나타낸다. Akiyo 나 Container 같은 움직임 정보가 별로 없는 시퀀스들을 보면 모드8의 Ratio가 매우 낮다는 것을 알 수 있다. 그 두 시퀀스의 평균 Ratio는 5%이하이다. 이것으로 볼 때 우리는 움직임 정보가 별로 없는 시퀀스에서 모드8은 그리 중요한 역할을 하지 않는다는 것을 알 수 있다.

이로써 우리는 움직임 정보가 별로 없는 시퀀스들에 있어서 모드8 조기 스킵 알고리즘의 적용은 특히 적절할 것이라는 것을 예측할 수 있다. 표 4에서 우리는 모드8의 조기 스킵 알고리즘이 모든 시퀀스에서 매우 고려할만하다 라는 것을 관찰할 수 있다. 적절한 데이터 수집을 위해서 우리는 모드결정과정에서 모드8을 빼고 인코딩을 해보았다. 그리고 원래의 모드결정방법에 비해서 성능이 얼마나 변했는지 인코딩 시간은 얼마나 감소했는지에 대한 데이터를 추출했다. 그 결과들은 표 4에서 확인할 수 있다.

표 3. 최적모드인 모드8의 비율

Sequences	최적모드8의 개수	Ratio(%)
Foreman	2471	12.7
Akiyo	425	2.2
Coastguard	3588	18.5
Stefan	4340	22.4
Mobile	6650	34.3
Container	584	3.0
Tempete	4898	25.2

표 4. 모드 결정과정에서 모드8을 뺐을경우의 변화(CIF, QP=28, 50프레임)

Sequences	DP(dB)	DB(%)	TR(%)
Akiyo	-0.02	1.26	-38.6
Bus	-0.02	3.31	-51.5
Children	-0.05	7.37	-44.6
Coastguard	-0.04	0.96	-51.5
Container	-0.01	1.67	-41.4
Flowerr	-0.01	3.87	-48.8
Football	-0.01	4.76	-51.1
Foreman	0.01	7.29	-45.7
Hall-mon.	-0.03	2.05	-39.4
Mobile	-0.07	2.58	-52.1
Mother	0	1.06	-40.8
Stefan	-0.03	3.63	-49.3
Table	-0.02	5.01	-47.1
Tempete	-0.03	2.27	-51.0
Average	-0.02	3.36	-46.6

DP는 원래의 인코더와 모드8을 고려하지 않은 인코더 사이의 PSNR의 차이를 나타낸다. DB는 비트율의 증가량을 나타낸다. 그리고 TR은 시간 감소율을 나타낸다. PSNR의 평균 감소치를 보면 거의 무시할만하다. 그러나 비트율의 증가는 꽤 크다고 할 수 있다. 시간감소율은 이 실험에서 가장 주목할 만한 결과라고 할 수 있는데, Mobile 시퀀스에서는 단지 모드8만을 인코딩시 제외했을 뿐인데 무려 52% 가량의 시간 감소율을 보였다. 평균 시간 감소율도 대략 47%에 달한다. 이로써 우리는 모드 결정과정에서 모드8을 고려하는 것이 얼마나 중요한 것인가 하는 것을 알 수 있다.

표 3에서 Akiyo와 Container에서의 모드8이 차지하는 비율은 2~3%정도 밖에 되지 않았다. 하지만 이 작은 비율을 고려하지 않고 인코딩을 했을때는 무려 40%가까이 시간감소를 시킬수가 있다. 하지만 모드8의 제거가 가져다주는 이 시간적인 이점에도 불구하고 우리는 전체 인코딩에 있어서 성능을 고려하지 않을 수 없다. 제안된 알고리즘의 주된 특징들 중의 하나가 성능 감소의 최소화이다. 비록 표 4에서는 모드8을 고려하지 않음으로써 비트율에 있

어서 큰 증가를 보여주고 있지만, 우리는 눈에띄는 성능저하 없이 많은 시간 감소를 이끌어낼 수 있는 효율적인 모드8의 조기 스킵 알고리즘을 제안하고자 한다.

조기에 스킵될 수 있는 모드8을 적절히 검출하기 위해서, 우리는 이미 앞절에서 사용된 모드들을 이용하고자 한다. 모드2, 모드3, 그리고 모드0의 관계를 적절히 이용하면 효율적으로 모드8을 스킵할 수 있을거라는 것이 우리의 가정이다. 이들 세 모드의 RDC 값들이 모드8 스킵을 위한 중요한 요소들이다. 이러한 관계가 식 (7)에서 보여진다.

$$If \left\{ \left[\frac{((RDC_2 + RDC_3) \div 2)}{RDC_0} \times 100 \right] > TH8 \right\} \quad (7)$$
$$Then \text{ MD8 is early skipped}$$
$$Otherwise \text{ MD8 is considered}$$

이 방정식에서 $TH8$은 모드8 스킵을 위한 임계값을 나타내며 매우 중요한 파라미터이다. 그러므로 $TH8$은 철저한 실험을 통해 미리 고정시키고자 한다. 우리는 모드2와 모드3의 RDC평균값과 모드0의 RDC 값을 비교함으로써 모드8을 조기 스킵할 수 있는 기준으로 삼고자 한다.

모드0과 모드1에 대한 조기종료 알고리즘을 충족시키지 못하고 남은 MB들의 다양한 통계적 특성들이 표 5에서 보여진다.

rMBs는 모드01 조기종료 알고리즘에 의해 검출되지 않은 MB을 뜻한다. bMBs는 rMBs중에서 최적 모드가 모드8인 MB의 수를 나타낸다. dMB8s는 식(7)을 사용해서 최적모드가 모드8로 검출된 수를 나타낸다. nMB8은 식(7)을 사용해서 최적모드가 모드8로 잘못 검출된 경우의 수를 나타낸다. 이때의 MB의 최적모드는 실제로 모드8이 아닌 것이다. tSKIP은 식 (7)을 이용해서 모드8이 조기에 스킵된 MB을 나타낸다. tSKIP의 수는 식 (8)을 통해서 계산이된다. D8 ratio와 tSKIP ratio는 각각 식 (9)와 (10)을 통해서 계산된다.

$$\#\text{ of } tSKIP = \#\text{ of } rMBs - \#\text{ of } dMB8s - \#\text{ of } nMB8s \quad (8)$$

$$D8 \text{ ratio} = \frac{\#\text{ of } dMB8s}{\#\text{ of } bMBs} * 100 \quad (9)$$

$$tMBskip \text{ ratio} = \frac{\#\text{ of } tSKIP}{\#\text{ of } rMBs} * 100 \quad (10)$$

식 (9)은 모드8이 최적모드로써 바르게 검출되는 비율을 나타낸다. 식 (10)은 불필요한 모드8들이 식 (7)을 이용해 스킵되는 비율을 나타낸다. 평균적인 D8 ratio는 73%이다. Akiyo, Container, 그리고 Hall-monitor 같은 모션정보가 적은 시퀀스들의 D8 ratio는 특히 낮다. 이것은 모션정보가 적은 시퀀스들에 대해서 우리 알고리즘의 디텍션 정확도가 낮다는 것을 의미한다. 하지만 우리는 실험결과로부터 이것이 상당한 성능 감소를 야기시키지는 않는다는 것을 알았다. tMBskip의 값은 상당히 좋은 결과를

표 5. 2절의 알고리즘을 통해 조기종료 되지 않은 MB들의 통계적 특성(CIF, QP=28, 50프레임)

Sequences	# of rMBs	# of bMBs	# of dMB8s	# of nMB8s	# of tSKIP	D8 tatio (%)	tMBskip ratio (%)
Akiyo	2763	398	213	710	1840	53.5	66.6
Bus	14362	3984	3231	7260	3871	81.1	26.9
Children	7992	3558	3298	3501	1193	92.7	14.9
Coastguard	14635	3469	2521	6753	5361	72.7	36.6
Container	3630	482	179	756	2695	37.1	74.2
Flowerr	11988	4995	4280	4896	2812	85.7	23.5
Football	13952	5224	5008	6256	2688	95.9	19.3
Foreman	10725	2410	1918	4645	4162	79.6	38.8
Hall-mon.	4950	932	568	1305	3077	60.9	62.2
Mobile	16030	6401	4545	6543	4942	71.0	30.8
Mother	4216	346	239	1054	2923	69.1	69.3
Stefan	12468	4108	3197	5600	3671	77.8	29.4
Table	11520	2731	2035	4104	5381	74.5	46.7
Tempete	13704	4630	3208	5000	5496	69.3	40.1

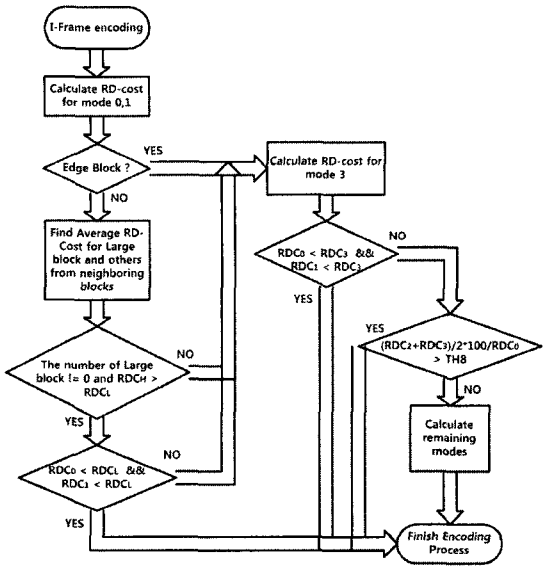


그림 3. 전체 알고리즘에 대한 블록도

보여준다. 평균값이 대략 41%이다. 이것은 우리가 MD01 조기종료 알고리즘이 적용되고 나서 남아있는 MB들중 41%에 대해 모드8을 조기 스킵 시킬 수 있음을 나타낸다. 모드8 조기 스킵 알고리즘의 적용에 대한 결과로써, 우리는 상당한 양의 시간 감소를 이끌어낼 수 있었다. 상세한 결과는 실험 결과를 설명하는 절에서 보여진다.

지금까지 설명한 전체 알고리즘에 대한 블록 다이어그램을 그림 3에서 보여주고 있다.

Ⅳ. 실험 결과

제안된 고속 모드 결정 알고리즘은 H.264 참조 소프트웨어인 JM12.4 [8]를 기반으로 개발되었다. 실험 조건은 표 6에서 보여지고 있다.

자세한 비교 실험 결과는 표 7에서 보여지고 있으며, 모든 값들은 세가지 파라미터에 기초해 계산되어졌다. 첫 번째 파라미터는 인코딩 시간 감소율(ΔTime)이고, 두 번째 파라미터는 PSNR차이(ΔPSNR)이다. 그리고 마지막 파라미터는 비트 변화율(ΔBR)이다. 이들에 대한 계산식은 각각 식 (11), (12), 그리고 (13)에서 보여진다.

$$\Delta Time = \frac{T_{proposed} - T_{JM12.4}}{T_{JM12.4}} \times 100(\%) \quad (11)$$

$$\Delta PSNR = PSNR_{proposed} - PSNR_{JM12.4}\ (dB) \quad (12)$$

표 6. 실험 조건들

PC environments	Windows XP professional 2GB memory Intel Pentium D3.2 GHz
MV search	±16
RD Optimization	Enabled
The number of reference frames	1
Profile	Baseline
MV resolution	1/4
GOP structure	IPPPPP...
The number of frames in a sequence	100
Motion estimation scheme	Full search
QP	26, 28, 30, 32, 34
Fast chroma intra mode decision(option)	Enabled
TH8	75 (Pre-fixed)

$$\Delta BR = \frac{BR_{proposed} - BR_{JM12.4}}{BR_{JM12.4}} \quad (13)$$

표 7에서의 값들중 음수는 감소를 나타내고 양수는 증가를 나타낸다. 기본적으로 JM12.4를 이용한 결과는 high complexity mode를 기반으로 하고 있다. 객관적인 실험결과의 비교를 위해서 2가지 비교 대상에 대한 결과를 이용하였다. 하나는 Zhan's algorithm [9]이고 나머지 하나는 기본적인 JM12.4 소프트웨어에서 두가지 옵션 "EarlySkipEnable", "SelectiveIntraEnable"을 추가한 버전인데 이것을 JM12.4에서는 Fast high complexity mode 라고 한다. [9]는 시공간적으로 현재 MB의 근처에 위치한 MB들을 고속 알고리즘을 위한 정보로써 이용하고 있는데 본 논문과 방법상 유사성을 가지고 있어서 성능을 비교하기에 적절하다고 본다.

JM12.4의 fast 버전은 기존 JM의 결과와 PSNR, bit rate 측면에서 매우 유사한 결과를 보여주었다. 그러나 시간 감소율은 그다지 높지 않음을 알 수 있었다. 제안된 알고리즘은 fast JM버전과 성능 측면에서는 거의 동일한 값을 보여주는 반면 월등히 빠른 시간감소율을 보여주었다. 특히나 Akiyo, Container, Mother 그리고 Hall-monitor 같이 움직임이 별로 없는 시퀀스에서는 평균 61%에 달하는 시간감소율을 보여주었다. Tempete, Mobile, Stefan 그리고 Coastguard와 같이 움직임이 많은 시퀀스에

표 7. 다른 알고리즘들과의 최종 비교 결과

Sequences	Zhan's			Fast JM			Proposed		
	ΔPSNR (dB)	ΔBR (%)	ΔTime (%)	ΔPSNR (dB)	ΔBR (%)	ΔTime (%)	ΔPSNR (dB)	ΔBR (%)	ΔTime (%)
Akiyo	-0.032	-0.10	-49.2	-0.020	-0.88	-50.3	-0.042	-0.70	-62.6
Bus	-0.006	0.54	-18.1	-0.013	0.06	-10.3	-0.044	0.22	-31.1
Children	-0.02	0.84	-30.9	-0.075	-0.01	-38.8	-0.035	1.07	-44.9
Coastguard	0.004	0.48	-18.2	-0.008	-0.06	-6.7	-0.028	-0.27	-31.8
Container	-0.039	0.83	-50.0	-0.019	-0.11	-40.7	-0.037	1.11	-64.8
Flowerr	0.015	0.52	-24.1	-0.030	0.14	-11.0	-0.044	0.30	-23.0
Football	-0.007	1.49	-24.7	-0.005	0.10	-18.8	-0.037	0.56	-40.4
Foreman	-0.007	0.43	-40.6	-0.026	-1.34	-32.2	-0.048	-1.85	-58.8
Hall-mon.	0.008	0.52	-15.3	-0.007	0.23	-5.3	-0.062	0.23	-25.6
Mobile	-0.024	1.12	-42.7	-0.042	-0.09	-37.6	-0.043	-0.24	-56.8
Mother	-0.023	0.40	-28.7	-0.045	-0.96	-30.2	-0.054	0.58	-49.8
Stefan	0.015	0.80	-15.5	-0.037	-0.01	-11.2	-0.058	0.15	-32.7
Table	-0.008	0.59	-19.1	-0.027	-0.11	-12.6	-0.050	0.39	-35.2
Tempete	0.013	0.96	-15.7	-0.009	0.20	-6.8	-0.051	0.26	-34.9
Average	-0.008	0.67	-28.1	-0.026	-0.20	-22.3	-0.045	0.13	-42.3

서도 평균 31%의 시간감소율을 보여주었다. 이것은 꽤 주목할 만한 수치이다.

그림 4-7에서는 성능의 시각적 비교를 위한 RD curve를 보여주고 있다. 5개의 QP값에 대한 결과를 비교하고 있으며, 제안된 알고리즘이 참조 소프트웨어의 성능과 거의 같음을 보여준다. 이는 다시 말해, 제안된 알고리즘을 이용하면, QP값이 변하여도 성능에는 거의 변화없이 고속으로 인코딩을 할 수 있음을 뜻한다.

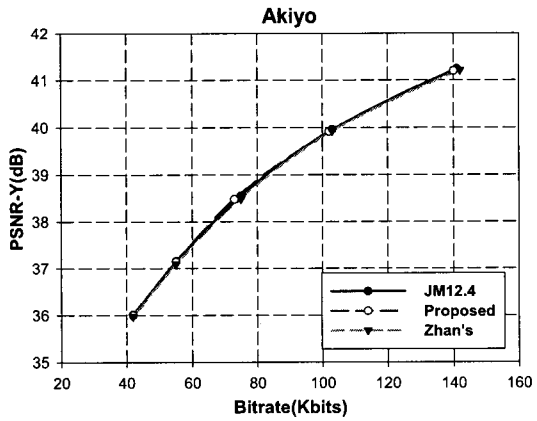


그림 4. RD Curve for Akiyo

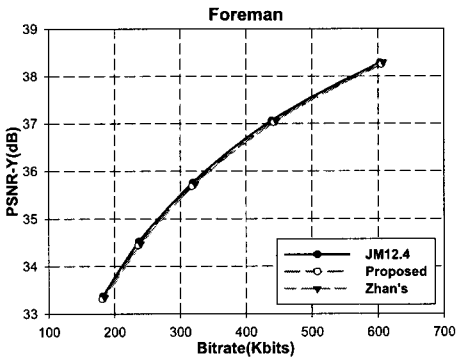


그림 5. RD Curve for Foreman

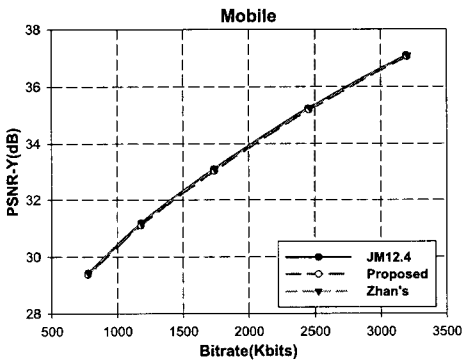


그림 6. RD Curve for Mobile

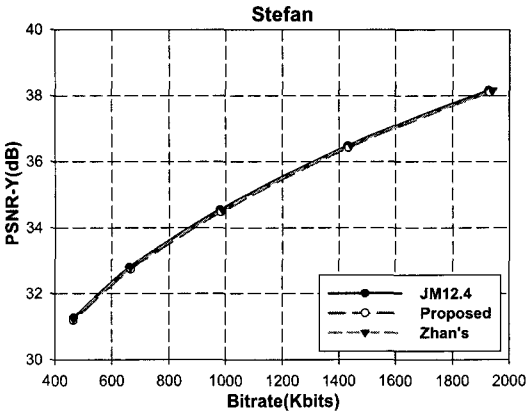


그림 7. RD Curve for Stefan

V. 결 론

본 논문에서는 H.264에서의 inter prediction을 위한 고속 모드 결정 알고리즘을 제안하였다. 2가지 기술들을 이용해 고속 알고리즘을 효율적으로 구현하고 있는데, 그 첫 번째가 최적 모드로써의 모드01을 조기에 검출하기 위해 사용되는 조기 종료 기술이다. 여기서는 주변 MB들의 RDC 관계를 이용하는 방법과 모드01과 모드3의 RDC관계를 이용하는 방법 이렇게 두가지를 이용했다. 또 한가지의 기술은 모드8에 대한 조기 스킵 알고리즘이다. 모드 결정과정에서 불필요하다고 판단되는 모드8을 조기에 스킵함으로써 상당량의 계산상의 복잡도를 줄였다. 제안된 알고리즘은 14개의 CIF 시퀀스에 대해 평균 42%의 시간 감소율을 보여주었으며 성능감소는 거의 없었다.

참 고 문 헌

[1] T. Wiegand, G. Sullivan, G. Bjonstegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., Vol.13, No.7, pp.560-576, Jul., 2003.

[2] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. Sullivan, "Rate-constrained coder control and comparison of video coding standards," IEEE Trans. Circuits Syst. Video Technol., Vol.13, No.7, pp.688-703, Jul., 2003.

[3] 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 Trans. Circuits Syst. Video Technol., Vol.15, No.7, pp.953-958, Jul., 2005.

[4] F. Pan, X. Lin, R. Susanto, K.-P. Lim, Z.-G. Li, G.-N. Feng, D.-J. Wu, and S. Wu, "Fast mode decision algorithm for intraprediction in H.264/AVC video coding," IEEE Trans. Circuits Syst. Video Technol., Vol.15, No.7, pp.813-822, Jul., 2005.

[5] J.Ren, N. Kehtarnavaz, and M. Budagavi, "Computationally efficient mode selection in H.264/AVC video coding", IEEE trans. Consumer Electronics, Vol.54, No.2, pp.877-886, May, 2008.

[6] A.-C.-W. Yu, G.-R. Martin, and H.-C. Park, "Fast inter mode selection in the H.264/AVC standard using a hierarchical decision process," IEEE Trans. Circuits Syst. Video Technol., Vol.18, No.2, pp.186-195, Feb., 2008.

[7] B.-G. Kim, "Novel inter-mode decision algorithm based on macroblock (MB) tracking for the P-slice in H.264/AVC video coding," IEEE Trans. Circuits Syst. Video Technol., Vol.18, No.2, pp.273-279, Feb., 2008.

[8] JVT reference software version JM 12.4 http://iphome.hhi.de/suehring/tml/download

[9] B. Zhan, B. Hou, and R. Sotudeh, "Temporal-spatial correlation based mode decision algorithm for H.264/AVC encoder," in Proc. 4th IEEE DELTA 2008, pp.351-355.

조 영 섭 (Youngsub Jo) 준회원

2001년 2월 한양대학교 전자전기컴퓨터공학부 학사

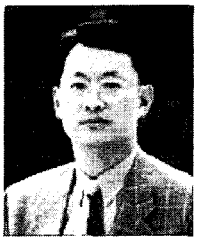
2010년 2월 한양대학교 전자통신컴퓨터공학과 석사

2004년~2008년 LG전자 MC연구소(Handset Mobile broadcasting SW 개발)

2010년~현재 현대 모비스 기술연구소

<관심분야> 영상 처리, 영상 압축

정 제 창 (Jechang Jeong) 정회원



1980년 2월 서울대학교 전자공학과 학사
1982년 2월 KAIST 전기전자 공학과 석사
1990년 미국 미시간대학 전기공학과 공학박사
1980년~1986년 KBS 기술연구소 연구원(디지털 TV 및 뉴미디어 연구)
1990년~1991년 미국 미시간대학 전기공학과 연구교수(영상 및 신호처리 연구)
1991년~1995년 삼성전자 멀티미디어 연구소(MPEG, HDTV, 멀티미디어 연구)
1995년~현재 한양대학교 전자통신컴퓨터공학과 교수(영상통신 및 신호처리 연구실)
1998년 11월 27일 과학기술자상 수상
1998년 12월 31일 정보통신부장관상 표창
<관심분야> 영상처리, 영상압축