

특집논문-08-13-2-04

## H.264|MPEG-4 AVC 비디오 부호화에서 움직임 벡터 정보와 16x16 및 4x4 화면 내 예측 최종 모드간 통계적 연관성을 이용한 화면 간 프레임에서의 4x4 화면 내 예측 고속화 방법

나태영<sup>a)</sup>, 정윤식<sup>a)</sup>, 김문철<sup>a)\*</sup>, 함상진<sup>b)</sup>, 박창섭<sup>b)</sup>, 박근수<sup>b)</sup>

### A Fast 4x4 Intra Prediction Method using Motion Vector Information and Statistical Mode Correlation between 16x16 and 4x4 Intra Prediction in H.264|MPEG-4 AVC

Taeyoung Na<sup>a)</sup>, Yunsik Jung<sup>a)</sup>, Munchurl Kim<sup>a)\*</sup>, Sangjin Hahm<sup>b)</sup>, Changseob Park<sup>b)</sup>, Keunsoo Park<sup>b)</sup>

#### 요약

H.264|MPEG-4 AVC는 ITU-T와 ISO/IEC 공동으로 결성된 JVT (Joint Video Team)에 의해서 정의된 가장 최신의 영상 압축 표준이다. H.264|MPEG-4 AVC는 효율적 부호화를 위하여 여러 방법이 제안되었는데, 화면 간 프레임(P-frame)에서의 화면 내 예측(Intra Prediction)의 경우 매크로블록마다 후보 모드 결정 및 유효-왜곡 비용 계산에 따른 부호화 시간의 급격한 증가를 초래하여 고속화 방법의 필요성이 대두되고 있다. 본 논문에서는 16x16과 4x4 화면 내 예측 부호화 결과를 바탕으로, 두 예측 결과의 통계적 상관관계를 규정한 후, 이를 활용한 4x4 화면 내 예측의 후보 모드 수를 감소시키는 방법을 제안한다. 구체적으로는 화면 간 예측(Inter Prediction) 단계에서 결정된 움직임 벡터 정보를 이용하여 현재 매크로블록의 화면 내 예측이 필요한지를 미리 판정한 후, 매 화면 내 프레임(I-frame)의 16x16 화면 내 예측의 최종 후보 모드에 따른 4x4 화면 내 예측의 최종 결정 모드들의 발생분포를 누적 확률 순으로 배열하여 특정 누적 확률에 도달하기까지 만의 후보 모드들만을 예측에 포함하는 참조 테이블을 부호화 과정 중에 생성한 후 동일 GOP 내에 위치하는 모든 화면 간 프레임의 화면 내 예측 시 활용하게 된다. 제안하는 방법은 H.264|MPEG-4 AVC의 참조 소프트웨어인 JM11.0을 사용하여 실험하였으며, 총 부호화 시간을 최대 51.24% 감소시킬 수 있었으며 PSNR 감소와 비트율 증가는 무시할 정도의 작은 변화만 있었다.

#### Abstract

H.264| MPEG-4 AVC is a new video coding standard defined by JVT (Joint Video Team) which consists of ITU-T and ISO/IEC. Many techniques are adopted for the compression efficiency: Especially, an intra prediction in an inter frame is one example but it leads to excessive amount of encoding time due to the decision of a candidate mode and a RDcost calculation. For this reason, a fast determination of the best intra prediction mode is the main issue for saving the encoding time. In this paper, by using the result of statistical relation between intra 16x16 and 4x4 intra predictions, the number of candidate modes for 4x4 intra prediction is reduced. Firstly, utilizing motion vector obtained after inter prediction, prediction of a block mode for each macroblock is made. If an intra prediction is needed, the correlation table between 16x16 and 4x4 intra predicted modes is created using the probability during each I frame-coding process. Secondly, using this result, the candidate modes for a 4x4 intra prediction that reaches a predefined specific probability value are only considered in the same GOP. For the experiments, JM11.0, the reference software of H.264|MPEG-4 AVC is used and the experimental results show that the encoding time could be reduced by 51.24% in maximum with negligible amounts of PSNR drop and bitrate increase.

Keywords: video coding, H.264|MPEG-4 AVC, intra prediction, inter prediction, motion vector, block mode decision

## 1. 서론

H.264|MPEG-4 AVC는 ISO/IEC의 MPEG(Moving Picture Expert Group)진영과 ITU-T의 VCEG(Video Coding Expert Group)가 공동으로 결성한 JVT(Joint Video Team)에 의해 표준화된 가장 최신의 비디오 압축 표준 기술이다<sup>[1][2]</sup>. H.264|MPEG-4 AVC는 이전의 압축 표준인 H.261, MPEG-2, H.263, MPEG-4 Visual (Part 2)등에 비해서 고효율, 고화질로의 압축 부호화가 가능하다. 이러한 고성능의 영상 압축을 가능하게 하기 위해서 H.264|MPEG-4 AVC에는 다양한 부호화 도구들이 포함되어 있는데, 이러한 도구들은 부호화 복잡도를 고려하지 않고 압축 성능을 향상시키기 위함에 초점이 맞추어져 있다. 구체적으로는 다양한 예측 블록 크기(Variable Block Size) 지원, 1/4-화소 단위 움직임 예측, 화면 간 프레임(P or B-frame)을 위한 다중 참조 프레임 지원, 디블록킹 필터, 엔트로피 부호화 도구로서 CAVLC(Context-Adaptive Variable Length Coding)/CABAC(Context-Adaptive Binary Arithmetic Coding) 사용 등을 예로 들 수 있다. 다양한 예측 블록 크기의 경우 이전 영상 압축 표준인 MPEG-4 Visual의 경우, 8x8까지의 예측 블록 크기를 지원한 데에 반해 H.264|MPEG-4 AVC의 경우 더욱 세밀한 예측을 위해 서브매크로블록이라 불리는 8x8 부터 4x4 블록단위까지 고려된 움직임 예측 부호화를 가능하게 하였다. 따라서 식 (1)에 표현된 바와 같이 총 10개의 다른 크기와 예측방법으로 매크로블록을 예측하여 최소의 윌-왜곡 비용을 갖는 블록 크기를 최종 블록 모드로 선택하게 된다. 여기에 FRExt(Fidelity Range Extensions)를 사용한 부호화 시 18x8 모드가 추가된다<sup>[1][12]</sup>.

$$mode \in \{SKIP, 16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8, 8 \times 4, 4 \times 8, 4 \times 4, I16 \times 16, I4 \times 4\} \quad (1)$$

식 (1)에서 I16x16, I4x4로 표현된 모드는 화면 내 예측

(intra prediction)으로 윌-왜곡 비용을 구한 블록이다.

H.264|MPEG-4 AVC에서는 화면 간 프레임 부호화의 경우에도 움직임 벡터를 통한 화면 간 예측 (inter prediction) 뿐만 아니라 화면 내 예측을 통해 계산된 윌-왜곡 비용의 비교를 통해 최종 블록 모드를 결정하게 된다. 화면 간 예측은 인접한 두 프레임간의 시간적인 연관성(temporal correlation)을 이용하여 현재 매크로블록과 가장 유사한 블록을 참조 프레임에서 찾는 것이며 그 위치를 움직임 벡터를 통해 표시한다. 16x16 화면 내 예측의 경우에는 부호화 중인 매크로 블록의 주위 블록, 4x4 화면 내 예측의 경우에는 매크로 블록 내부 혹은 외부의 4x4 블록간의 공간적인 연관성(spatial correlation)을 이용하여 현재 매크로블록과 가장 유사한 블록을 찾는 것이다. 화면 내 프레임(I-frame)의 경우, 모든 매크로블록을 화면 내 예측을 통해 부호화하게 되지만 화면 간 프레임의 경우 두 가지 방법 모두를 활용하여 예측 모드들의 윌-왜곡 비용을 계산한 후 그 값이 가장 작은 모드를 한 매크로블록의 최종 모드로 결정하게 된다. 이러한 정교한 비용의 비교를 통해 H.264|MPEG-4 AVC는 다른 영상 압축 표준에 비해 윌-왜곡 성능을 크게 향상시킬 수 있다<sup>[3]</sup>.

화면 간 예측이 현재 블록과 가장 유사한 블록을 참조 프레임에서 찾는 것이라면 화면 내 예측은 가장 유사한 블록을 미리 정의된 방향성 후보 모드에서 찾게 된다. 이 방향성 후보 모드는 현재 블록의 주위 블록 화소값으로부터 결정되며 화면 간 예측의 블록 크기에 따라 미리 정해진 수만큼 필요하게 된다. 4x4 화면 내 예측을 예로 든다면, 4x4 블록 당 아홉 개의 방향성 후보 모드가 있으며, 이 모드들 모두에 대해 윌-왜곡 비용을 계산하여 최적의 4x4 블록을 결정하게 된다. 이러한 블록 16개가 모여 한 매크로 블록을 이루게 되므로 휘도 성분만 고려하더라도 한 매크로블록 당 144번의 윌-왜곡 비용 계산이 필요하게 된다. 16x16 화면 내 예측과 색차 성분에 대한 화면 내 예측까지 고려한다면 이로 인한 부호화 복잡도가 매우 높음을 알 수 있다. H.264|MPEG-4 AVC는 이 뿐만 아니라 고성능 압축 효과를 얻기 위해 복잡도가 높은 많은 부호화 도구들을 포함하고 있다. 따라서 실시간 부호화가 가능한 H.264|MPEG-4 AVC 소프트웨어 부호화기를 구현을 위해서는 복잡도의

a) 한국정보통신대학교(ICU)

Information and Communications University

b) 한국방송공사 기술연구소(KBS)

Korea Broadcasting System

‡ 교신저자 : 김문철(mkim@icu.ac.kr)

많은 부분을 차지하고 있는 화면 내 예측 고속화의 필요성이 있다. 4x4 화면 내 예측은 후보 모드 수가 아홉 개나 되므로 복잡도 측면에 비해 최종 모드로의 선택 비율이 낮음을 감안하면 반드시 부호화 과정에서 4x4 화면 내 예측 수행의 고속화 방안이 반드시 마련되어야 하며 이미 많은 연구자들에 의해서 많은 고속화 방법들이 제안되었다<sup>[4]-[7][9][10][13]-[15]</sup>. 하지만 이러한 방법들에도 불구하고 여전히 화면 내 예측은 부호화기 복잡도의 많은 부분을 차지하고 있으며, 더욱 효과적인 고속화 방법이 절실히 요구된다.

본 논문에서는 우선 다양한 영상에 대한 실험을 통해 결정된 움직임 벡터의 크기에 따른 최종 모드의 결정 비율을 활용하여 매크로블록의 화면 간 예측 후에 산출된 움직임 벡터 정보를 활용하여 화면 내 예측이 필요한지 여부를 확인한 후, 화면 내 예측이 불필요하다고 판단되면 이후 예측 과정을 생략하여 큰 복잡도 감소를 얻게 된다. 만약 필요하다고 판단될 경우 4x4 화면 내 예측 이전에 수행되는 16x16과 4x4 화면 내 예측을 통해 최종 결정된 후보 모드들 간의 방향적 유사성을 통계적으로 표현한 후 이를 활용하여 4x4 예측에 사용되는 후보 모드들의 수를 감소시키는 방법을 소개한다. 영상의 특성과 부호화 단계 이전에 선택되는 QP 값에 따라 사용하게 되는 통계의 분포가 상이하므로 몇 개의 표본들로부터 자료를 취합하여 일괄적인 적용을 위한 하나의 참조 표를 만들지 않고, 통계 정보가 담긴 참조 표의 작성과 갱신 과정을 부호화 수행과정 내부로 포함시켜 영상의 종류와 크기 및 QP에 따른 적응적인 통계 자료를 활용할 수 있게 하였다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 H.264/MPEG-4 AVC 표준에서의 울-왜곡 최적화 관점에 따른 화면 내 예측 방법과 최종 블록 모드선택 측면에서의 통계적 특성에 대해 설명하며 3장에서는 화면 내 예측 관련 기존 고속화 연구와 본고에서 제안하는 화면 내 예측 고속화 방법을 설명한다. 4장에서는 화면 내 예측 고속화 방법을 제안한다. 5장에서는 제안된 방법을 다른 고속화 방법 및 원래의 H.264/MPEG-4 AVC 표준안과 부호화 시간 및 울-왜곡 성능 측면에서 비교하며 마지막으로 6장에서 결론을 맺으며 본 논문을 마친다.

## II. 화면 내 예측 방법 및 최종 블록 모드의 분포 특성

H.264/MPEG-4 AVC 바로 이전의 비디오 부호화 표준인 MPEG-4 Visual의 경우 화면 내 예측을 위해 고려되는 블록은 8x8이었다. H.264/MPEG-4 AVC는 더욱 정확한 예측을 위해 고려하는 블록 크기가 다양해 졌으며 사용되는 후보 모드의 수도 증가하였다. 휘도 성분의 경우 16x16과 4x4 블록 크기가 화면 내 예측에 사용되며 FExt 사용 시에는 8x8 화면 내 예측이 지원된다. 따라서 총 3가지 블록 크기의 화면 내 예측을 각각 독립적으로 수행하게 되고, 각각의 블록 크기에 대해 각각 4, 9, 9개의 후보 모드를 가진다. 색차 성분에 대해서는 4:2:0의 경우 8x8 블록 크기로 4개의 후보 모드를 예측에 사용한다. 1장에서 언급하였듯이 각 후보 모드는 주위 블록의 화소값으로부터 결정되어 현재 블록과의 울-왜곡 비용과의 비교를 통해 최소의 비용을 갖는 후보 모드를 최적 모드로 결정하게 된다.

후보 모드의 수와 고려해야 하는 블록 크기로 볼 때 각 블록 별, 화소 종류별로 수행되는 화면 내 예측에서 가장 복잡도가 높은 것은 4x4 휘도 성분에 대한 화면 내 예측임을 알 수 있다.

그림 1의 (a)와 (b)에서 화면 내 예측 블록 크기에 따른 각각의 후보 모드들의 방향과 이름이 같이 표시되어 있다. 16x16 화면 내 예측의 경우에 Vertical mode(모드 0), Horizontal mode (모드 1), DC mode(모드 2), Plane mode(모드 3)가 있으며, 4x4 화면 내 예측의 경우에는 0부터 2까지의 후보 모드의 방향은 16x16 화면 내 예측의 그것과 동일함을 알 수 있지만 더욱 정교한 예측을 위하여 6개의 방향을 더 고려하고 있음을 알 수 있다. 16x16 화면 내 예측의 경우 후보 모드들에 대해 각각 현재 부호화 대상 블록과 후보 모드와의 차분을 통해 잔차값을 구하여 이를 수평방향, 수직방향으로 하다마드 행렬을 사용한 변환을 실행한 후에 절대치 오차합을 산출하며 이를 SATD(Sum of Transformed Differences)라 한다. 4x4 화면 내 예측의 경우는 각각의 4x4 블록 단위에서 아홉 개의 후보 모드 각각의 최적 모드를 결정하기 위한 울-왜곡 비용을 구하여 매크로블록 내 각기 다른 4x4 블록 위치에서의 울-왜곡 비용을 구하는

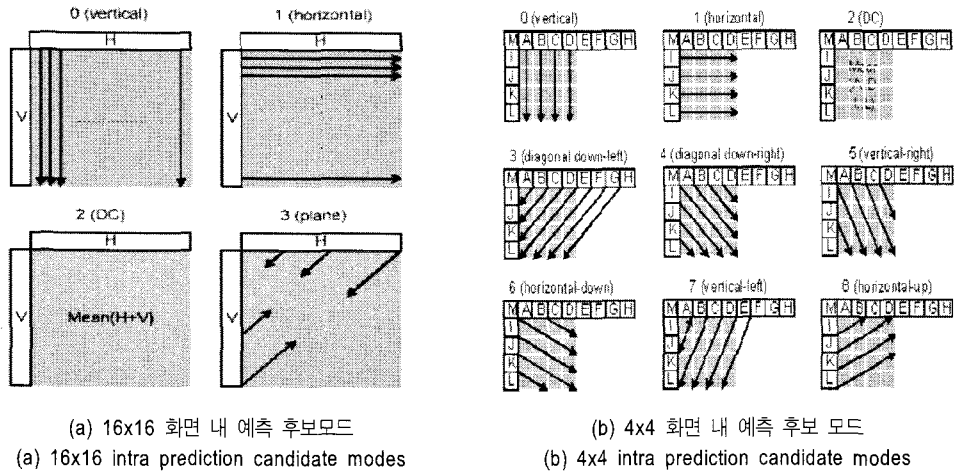


그림 1. 휘도 성분에 대한 16x16 및 4x4 화면 내 예측의 후보 모드<sup>[10]</sup>  
Fig.1. Candidate modes for 16x16 and 4x4 prediction for luma component<sup>[10]</sup>

과정이 필요하다. 다시 말하면 최종 모드는 화면 간 / 화면 내 예측 여부에 관계없이 16x16 블록 크기로 울-왜곡 비용만을 고려하여 결정되기 때문에 4x4 화면 내 예측을 16번 수행하는 과정이 필요하게 된다. 여기서는 SATD가 아닌 잔차값의 절대치의 오차합인 SAD(Sum of Absolute Differences)가 사용된다. 울-왜곡 비용 계산 시 필요한 연산량은 식 (2)에 나타나 있다.

$$\text{Chroma} \times (16 \times \text{Luma}_{4\text{by}4} + \text{Luma}_{16\text{by}16} + 4 \times \text{Luma}_{8\text{by}8}) \quad (2)$$

Chroma로 표기된 것은 색차 신호에 대한 화면 내 예측이므로 4개의 후보 모드를 가진다. 따라서 식 (2)의 결과는 736이 되며, 8x8 화면 내 예측이 일반적인 경우가 아니므로 제외한다면 총 592번의 연산이 필요하게 된다. 이 수치는 한 매크로블록에서 필요한 연산의 수이므로 프레임 단위에서 본다면 부호화기의 복잡도를 크게 높이는 원인이 된다. 표 1을 살펴보면 이렇게 높은 복잡도에도 불구하고 최종 모드로 화면 내 예측 블록이 결정되는 비율은 매우 낮음을 관찰할 수 있다.

표 1. 최종 블록 모드 분포(50 프레임 / QP 24, 28, 32 / CIF)  
Table 1. Distribution of final block mode (50 frames / QP 24, 28, 32 / CIF)

영상	QP	SKIP	16x16	16x8	8x16	P8x8	I4x4	I16x16
Coastguard	24	7.78%	31.05%	13.17%	13.99%	31.84%	0.18%	0.00%
	28	13.65%	34.49%	13.70%	14.91%	20.50%	0.30%	0.12%
	32	23.19%	35.70%	12.70%	13.95%	11.09%	0.80%	0.57%
Tempete	24	7.90%	29.18%	11.23%	10.38%	36.23%	2.82%	0.26%
	28	15.31%	30.90%	11.56%	10.91%	26.28%	2.52%	0.50%
	32	26.91%	30.50%	11.21%	11.14%	15.22%	2.35%	0.65%
Foreman	24	22.53%	26.28%	13.50%	14.40%	21.18%	1.24%	0.88%
	28	34.18%	27.88%	11.37%	13.18%	11.45%	1.04%	0.91%
	32	46.29%	27.40%	9.09%	9.54%	5.49%	0.77%	1.43%
Football	24	8.37%	15.12%	8.53%	9.75%	32.32%	23.67%	0.23%
	28	12.00%	16.78%	9.32%	10.47%	26.23%	22.50%	0.70%
	32	15.72%	18.51%	10.45%	11.67%	19.12%	19.96%	2.56%

표 1에 사용된 실험 영상은 카메라의 움직임과 객체의 움직임이 느린 *Coastguard*부터 빠른 *Football*까지 다양하며 *Tempete*의 경우는 줌 아웃으로 화면이 구성되어 있다. 화면 간 예측 시 움직임 예측과 보상을 통해 얻을 수 있는 총 7가지 블록 모드와 화면 내 예측으로 얻어진 116x16, 14x4 두 가지 경우에 대해 24, 28, 32의 다양한 양자화 계수에 따라 부호화하였을 때의 최종 모드 분포를 보여주고 있다. 결과적으로 *Football*을 제외한 모든 영상에서 4x4 화면 내 예측으로 최종 모드가 결정되는 경우는 3% 미만의 낮은 비율로 나타나며 이는 *Football*과 같이 카메라와 객체의 움직임이 극단적으로 빠른 영상에 대해 화면 내 예측이 유효-왜곡 성능 측면에서 유용함을 보여준다. 하지만 그렇지 않은 경우, 화면 내 예측으로 최종 모드가 결정되는 비율은 3% 미만이며, 이는 유효-왜곡 성능을 유지하면서 부호화기의 복잡도를 크게 증가시키는 4x4 화면 내 예측 고속화를 영상 특성에 따라 적응적으로 할 필요가 있음을 보여주고 있다. 움직임이 적은 정적인 영상에 대해서는 4x4 화면 내 예측을 수행하지 않더라도 부호화 성능에 큰 영향을 주지 않음을 알 수 있다. 또 한 가지로 QP에 따른 최종 모드 분포를 고려할 필요가 있는데 QP가 커질수록 영상의 세밀한 영역의 감소로 인하여 SKIP 모드 비율이 증가하며 상대적으로 P8x8과 I4x4 모드의 비율은 감소함을 확인할 수 있다. 따라서 이러한 영상 특성과 QP에 따른 모드 분포가 상이하므로 화면 내 예측의 고속화에 있어서도 이 두 가지가 모두 고려되어야 함을 알 수 있다.

### III. 움직임 벡터 정보와 16x16 및 4x4 화면 내 예측에 의한 최종 후보 모드들간 상관성을 이용한 4x4 화면 내 예측 고속화 방법

2장에서 언급한 바와 같이 최종 모드로 선택되는 비율이 낮은 화면 내 예측의 고속화를 위하여 이미 많은 연구가 진행되었다. 대표적으로 Sim *et al.*<sup>[4]</sup>, Park *et al.*<sup>[7]</sup>, Na *et al.*<sup>[9]</sup> 등은 주위 화면 내 기 결정 모드 정보를 바탕으로 화면 내 예측 고속화를 실현하였으며, Pan *et al.*<sup>[5]</sup>은 블록의 에지(Edge) 정보를 바탕으로 그래디언트(Gradient) 값을 도출하

여 방향성을 미리 예측한 고속화 방법을 여러 차례 제안하였다. Hwang *et al.*<sup>[6]</sup>과 같이 화면 간 예측에서 결정된 움직임 벡터 등의 정보를 이용하여 화면 내 예측을 고속화하는 방법도 다수 제안되었으며, Yang *et al.*<sup>[14]</sup>, Chun *et al.*<sup>[15]</sup>과 같이 블록의 수평 혹은 수직 성분의 강도를 미리 예측하여 화면 내 예측 고속화에 활용하는 방법도 많이 제안되는 방법들이다. 하지만 이들 대부분 고속화 이전에 비해 부호화 시간의 단축이 약 25-35%정도에 그쳐 여전히 실시간 부호화와는 거리가 멀다. 제안하는 방법은 단순히 몇 개의 후보 모드를 줄이는 것이 아닌 부호화 과정에서의 복잡도를 급감시킬 수 있으며 유효-왜곡 성능의 감소 또한 최소화하도록 설계되었다.

#### 1. 움직임 벡터 정보를 이용한 화면 내 예측 생략

제안하는 화면 내 예측 고속화 방법은 화면 간 예측을 통해 산출된 움직임 벡터 정보를 활용하여 움직임을 예측하게 된다. 표 2를 통해 움직임 벡터와 최종 블록 모드의 결정과의 관계를 도출할 수 있다.

표 2. 매크로블록 내의 움직임 벡터 크기와 화면 내 예측블록으로의 결정 확률과의 관계

Table 2. Relationship between motion vector and intra-predicted block as final block mode

(Mother and daughter, Foreman, Stephan / 100 frames / QP 28 / CIF)

$ MV $	inter block	intra block	$P_{intra}$
0~5	88462	758	0.01%
5~10	14036	470	3.24%
10~20	6109	380	5.86%
20~30	3301	241	6.80%
30~40	2382	496	17.23%
40~50	298	62	17.23%
50~60	90	43	32.33%
60~70	34	18	34.62%
70~80	18	13	41.94%
80~90	8	8	50.00%
90~100	40	12	23.08%

앞서 표 1에서 설명된 바와 같이 최종 모드 결정에 있어

영상의 균일한 영역은 SKIP모드나 16x16 블록 모드로 결정되는 경우가 많아 16x16 화면 내 예측 블록이 선택되는 경우는 드물다. 텍스처가 세밀한 영상의 경우라 할지라도 4x4 서브 매크로블록 단위까지의 화면 간 예측으로 인하여 보통 수준의 움직임 영상의 경우 화면 내 예측 블록으로 최종 모드가 선택되는 경우는 무시할 수 있는 수준이다. 그러나 *Football* 영상과 같이 카메라나 객체의 움직임이 큰 경우에는 화면 내 예측 블록이 최종 모드로 결정되는 비율이 높게 증가하며 QP 변화와는 큰 상관없이 표 1에서 확인하였다. 따라서 이러한 실험결과를 유추해보면 화면 내 예측 블록의 결정 비율은 움직임 벡터와 큰 연관성을 가질 것이라고 예상할 수 있으며 그 결과가 표 2와 그림 2에 잘 나타나 있다.

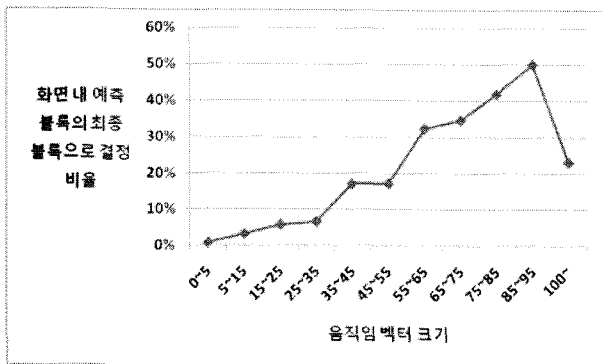


그림 2. 움직임 벡터 크기에 따른 화면 내 예측 블록 결정 확률 곡선  
Fig.2. Final mode decision curve according to the amplitude of motion vectors

움직임이 커져서 화면 내 예측 이전의 화면 간 예측에서 도출된 움직임 벡터의 크기가 커질수록 최종 모드의 화면 내 예측 블록으로의 결정 확률이 커지고 있음을 알 수 있다. 표 2에 사용된 움직임 벡터 크기는 정수 화소 단위의 값을 사용하였으며 식 (3)으로부터 도출된다.

$$\text{Amplitude of Motion Vector} = \overline{MV} = \sqrt{MV_x^2 + MV_y^2} \quad (3)$$

표 2와 그림 2는 영상의 움직임에 따른 결과의 편차를 최소화하기 위해 움직임이 느린 영상부터 빠른 영상들을

다양하게 사용하여 화면 내 예측 / 화면 간 예측 블록으로 최종 결정된 매크로블록의 수를 합하여 움직임 벡터의 크기에 따라 분류해 놓은 다음 화면 내 예측 블록의 발생 확률을 정리한 것이다. *Mother and daughter, Foreman, Stephan* 영상을 사용하여 QP 28로 100 프레임 부호화를 수행한 결과를 이용하였다. 이 결과를 바탕으로 움직임 벡터의 크기가 작아 화면 내 예측 블록으로 최종 모드가 결정될 확률이 낮은 경우에 대해 16x16, 4x4 화면 내 예측을 모두 생략할 수 있으며 다양한 영상에 이를 적용한 실험 결과, 움직임 벡터의 크기가 40보다 작은 화면 내 예측 결정 확률( $P_{intra}$ )이 10% 미만인 경우에 화면 내 예측을 생략하면 울-왜곡 성능을 크게 감소시키지 않고도 부호화기의 복잡도를 크게 낮출 수 있다. 또한 최종 모드가 움직임 벡터 크기와 함수 관계에 따라 화면 내 혹은 화면 간 블록으로 결정될 때에 QP의 변화에 따른 편차는 무시할 수 있는 수준이기에 QP 28만을 고려한 결과인 표 2를 다양한 QP에 대해 적용할 수 있다. 또한 모든 움직임 벡터 크기가 40 이상인 경우는 *Stephan* 영상을 제외하고는 거의 발생하지 않았다. 따라서 확률을 도출하기 위한 샘플 수가 적어 신뢰하기 어렵고 울-왜곡 성능 측면에서도 화면 내 예측 생략을 위한 식 (3)의 임계값은 40이 가장 최적의 값으로 판정한다. 정리하면 식 (4)와 같은 조건에서는 화면 내 예측을 생략한다.

$$\sqrt{MV_x^2 + MV_y^2} \leq 40 \quad (4)$$

식 (4)는 비록 실험을 통해 얻어진 값을 임계값으로 사용하지만 매우 느린 영상, 중간 움직임의 영상 및 매우 빠른 영상을 샘플 데이터로 사용하였으며 이를 다양한 다른 영상에 변경 없이 동일하게 적용하였다.

## 2. 16x16 및 4x4 화면 내 예측에 의한 최종 후보 모드들간 상관성을 이용한 4x4 화면 내 예측 고속화 방법

앞서 언급한 움직임 벡터 정보를 이용하여 화면 내 예측 부호화 수행 여부를 판단한 후 예측 부호화를 수행하는 것으로 결정되면 16x16 및 4x4 화면 내 예측을 수행한다. 이

경우 4x4 화면 내 예측의 복잡도를 낮추기 위한 방법으로 Park *et al.*은 화면 내 예측을 수행할 때 앞서 수행된 16x16 화면 내 예측 결과를 사용하였다<sup>[7]</sup>. 이를 위해 16x16 과 4x4 화면 내 예측에서 최종 결정되는 후보 모드들간의 방향적인 유사성이 매우 높은 것으로 가정하고 관련 있는 후보 모드들만을 고려하였다. 예를 들어 16x16 화면 내 예측에서 수직 방향 후보 모드인 0번으로 결정되면 4x4 예측에선 유사한 방향성을 갖는 모드인 7 (vertical-left), 0 (vertical), 5 (vertical-right)와 DC 모드만을 고려한다. 그 외의 경우에 대해선 표 3에 나타나 있다.

표 3. Park *et al.* 방법의 16x16 화면 내 예측 후보 모드 선택에 따른 고려되는 4x4 예측 후보 모드

Table 3. Candidates 4x4 modes according to 16x16 mode by Park *et al.*'s method

16x16 mode	Candidate 4x4 modes
mode 0 (vertical)	mode 7, 0, 5, 2
mode 1 (horizontal)	mode 8, 1, 6, 2
mode 2 (DC)	mode 0, 1, 3, 4, 2
mode 3 (plane)	mode 0, 1, 3, 2

그림 3은 16x16 화면 내 예측에서 최적 후보 모드가 결정 되었을 때, 그에 종속적으로 4x4 화면 내 예측에서 어떤 방향성 후보 모드가 선택 되는지를 조사한 것이다. 그림 3에서 확인할 수 있듯이 4x4 화면 내 예측 후의 최적 결정 모드들은 16x16 화면 내 예측 수행의 최종 결정 모드와 큰 연관성이 없음을 알 수 있다. 이는 Park *et al.*<sup>[7]</sup>의 논문에서 가정된 바와는 다르게 16x16 화면 내 부호화 예측 시 최적

결정 모드는 4x4에서의 최종 결정 모드들과 영상의 특성이나 크기 변화에 따라 연관성이 없을 수 있음을 의미한다. 또한 QP의 변화도 모드의 최종 결정에 영향을 끼칠 수 있다<sup>[9]</sup>.

16x16과 4x4 블록의 크기 차이에 의해 블록 내부에 포함되는 영상의 방향적인 연관성이 적은 경우가 발생할 수 있으며 세밀한 객체가 많이 포함된 영상에선 이러한 차이가 더욱 두드러지는데 그림 3의 *Mobile* 영상이 대표적이다. 특히 16x16 예측에서 3번 모드가 최적으로 결정된 경우에는 4x4에서는 그 후보 모드 선택 비율이 영상마다 상이함을 알 수 있다. 이는 대각선 방향성을 갖는 영상에선 4x4와 같이 작은 블록으로의 예측이 16x16에서의 그것과 매우 상이하다는 결론을 얻을 수 있다.

도출된 결과를 바탕으로 최적으로 결정된 16x16 블록 후보 모드에 따라 특정한 4x4 블록 후보 모드들만이 높은 선택 확률을 가짐을 확인할 수 있다. 이는 영상의 특성(텍스처 복잡도, 움직임 양, 해상도 등)과 QP 값 등을 고려하여 16x16 최적 후보 모드에 따른 9가지 4x4 후보 모드들의 선택 확률 분포를 이용하여 무시할 수 있을 정도의 선택 확률을 갖는 후보 모드들은 예측 부호화 과정에서 생략이 가능하다는 것을 의미한다.

본 논문에서 제안하는 방법은 선택된 16x16 화면 내 예측의 최종 선택모드에 대해 발생확률이 매우 낮은 4x4 화면 내 예측 방향성 후보 모드들을 배제한다. 발생 확률 변화에 영향을 끼치는 영상 크기, 특성 및 QP 등을 고려하기 위해 부호화 과정 중 매 화면 내 프레임(I-frame)마다 16x16 화

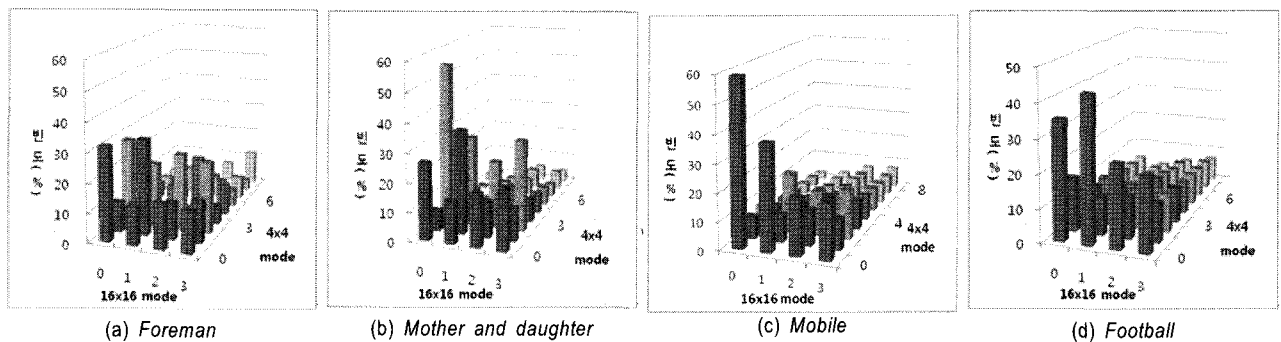


그림 3. 16x16 결정 모드에 종속인 4x4 최적 결정 분포 (50 frames / QP28 / CIF)

Fig.3. Best 4x4 intra predicted block mode dependent on 16x16 modes (50 frames / QP28 / CIF)

면 내 예측을 통해 최적으로 결정된 모드에 따른 모든 경우의 4x4 화면 내 예측 방향성 각 후보 모드의 선택 확률 분포표를 생성해 확률 순서에 따라 내림차순으로 정렬 하도록 한다. 즉, 매 I-프레임 마다 4x4 화면 내 예측 최종 선택 방향성 모드의 확률을 갱신한다. 이는 영상의 패턴이 매 프레임마다 변할 수 있으므로 4x4 화면 내 예측 최종 방향성 모드 선택 확률 분포가 변화할 수 있기 때문에 매 화면 내 프레임에 대해서만 확률 분포표를 갱신하여 다음 I-프레임까지 적용하여 확률 갱신에 대한 복잡도를 증가를 최소화한다. 식 (5)를 통해서 최초 16x16 최종 모드 결정에 따른 4x4 모드들의 결정 확률을 표현한다.

$$P_x(x|y) \tag{5}$$

여기서  $y$ 는 16x16 화면 내 예측을 통해 최적으로 결정된 후보 모드이고 0부터 3까지의 값을 갖는다.  $x$ 는 그에 따른 4x4 예측에서 결정된 후보 모드로서 0부터 8까지의 값을 가진다. 발생확률이 낮은 후보 모드들은 화면 내 예측에서 고려하지 않을 것이므로 식 (6)과 같이 내림차순으로 정렬하는 것이 효과적이다. 식 (5)에서 4x4 화면 내 예측 방향성 모드들( $x=\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ )에 대한 조건부 확률 값을 내림차순으로 정렬한 확률을  $P_z(z|y)$ 이라 하며 모든 16x16 모드 결정에 따라 그림 4와 같은 확률 분포가 부호화기 내에 저장되어야 하며 발생 빈도가 극히 낮은 방향성 모드들을 화면 내 예측 수행 시에 고려대상에서 제외할 수 있음을 알 수 있다.

마지막으로 제안하는 고속화 방법을 적용하기 위하여 상기

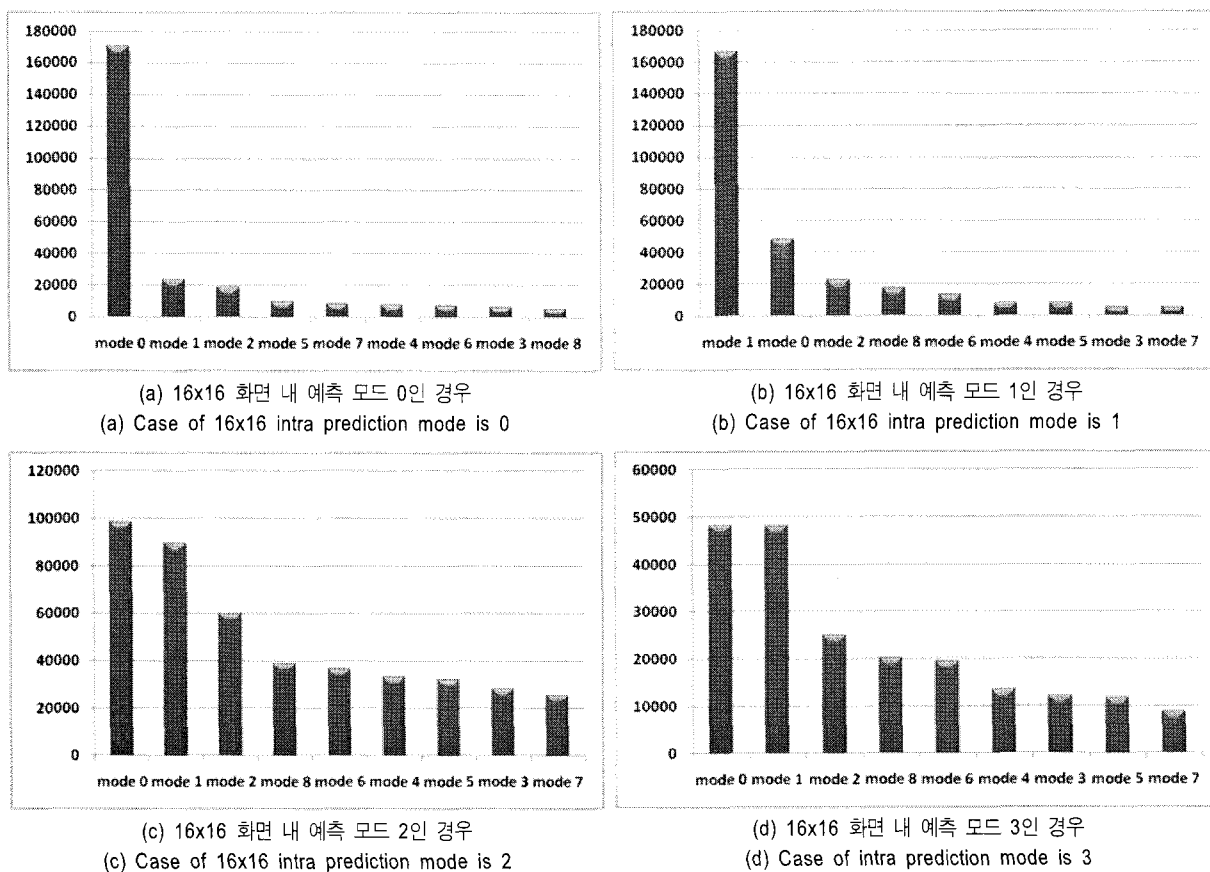


그림 4. Paris 영상에서 16x16 결정 모드에 따른 4x4 최적 결정 분포

Fig.4. 4x4 intra predicted block mode distribution dependent on 16x16 modes using Paris sequence



조건부 확률을 누적 분포 확률로 변환한다. 예를 들어, 만약 발생 확률 순서가 모드 0,1,2,3,4,5,6,7,8 이라면  $P_z(z=0|y) \geq P_z(z=1|y) \geq P_z(z=2|y) \geq \dots \geq P_z(z=8|y)$  과 같이 누적 확률( $F_D(d|y)$ )로 다음 식 (6)과 같이 표현 할 수 있다. 즉 주어진 조건인  $y$ 에 따른  $z$ 의 분포를 누적 확률로 바꾸어 준다.

$$\sum_{z=0}^d P_z(z|y) = F_D(d|y) \leq 1 \quad (6)$$

식 (6)에서  $d = 0, 1, \dots, 8$  이다. 제안하는 방법을 적용하기 위해 4x4 화면 내 예측 수행 시에 발생 빈도가 높은 방향성 후보 모드를 선택하기 위해 누적 확률 ( $F_D(d|y)$ ) 값 0.8을 임계값으로 사용하였으며 이는 누적 발생 분포 확률이 80%가 될 때까지의 방향성 후보 모드들만을 4x4 화면 내 예측 수행에 고려한다는 의미이다. 이 경우에 물론 20%의 확률로 최적 모드 결정에 잘못된 방향성 모드를 선택할 경우가 발생하나 다양한 실험을 통해 울-왜곡 성능에 큰 영향을 끼치지 않음을 확인하였다. 상기 조건을 만족하지만, 무시할 정도 크기의 발생 확률을 갖는 후보 모드까지 화면 내 예측에 고려하는 것은 부호화 복잡도 증가를 초래하여 실시간 부호화에 상당한 부담으로 작용할 뿐만 아니라 화질에도 큰 영향을 끼치지 않기 때문에 후보 모드 결정 시 식 (7)의 조건을 추가한다.

$$P_{Candidate Mode} \geq \frac{1}{4} P_{Most Probable Mode} \quad (7)$$

좌변의 확률은  $P_z(z|y)$ 에 의해 구해진 4x4 화면 내 예측 방향성 후보 모드 각각의 확률을, 우변의 확률은 이들 중 가장 발생 확률이 높은 후보 모드의 확률 값이다. 상수 1/4은 반복적인 실험을 통해 얻어진 값이며 결론적으로 발생 확률에 따른 내림차순 정리 후, 누적 확률이 80%에 도달하며, 가장 높은 발생 확률 값의 1/4 이상이 되어야만 4x4 화면 내 예측 부호화 수행 시에 고려되는 후보 모드로 지정된다. 즉, 16x16 화면 내 예측에서 결정된 후보모드에 따라 부호화 조건(영상 종류, 텍스처, QP)이 최적으로 반영되어

확률 통계적으로 80% 연관성이 있는 4x4 후보 모드들을 예측에 고려하게 되어 화면 내 예측의 고속화를 가능하게 한다. 제안하는 고속화 방법의 순서를 정리하면 다음과 같다.

#### 제안하는 4x4 화면 내 예측 고속화 방법순서

- 순서 1-1: 매 I-프레임에서 화면 내 프레임에서 16x16 및 4x4 블록의 화면 내 예측을 통해 각 16x16 화면 내 예측의 최적 방향성 후보 모드에 따른 4x4 화면 내 예측 최적 방향성 후보 모드 확률 표를 생성 및 저장한다.
- 순서 1-2: 발생 확률 순서에 따른 내림차순으로 정렬하여 누적 확률 분포 표를 생성한다.
- 순서 1-3: 정렬된 모드 순으로 누적 확률에 대한 임계값과 식 (8)의 조건을 충족시키는 후보 모드들만을 화면 내 예측에 고려한다.
- 순서 1-4: 순서 1-3에 의한 후보 모드들만을 다음 I-프레임 전까지 화면 간 프레임 부호화 과정에서 4x4 화면 내 예측 수행 시에 후보모드로 고려한다.
- 순서 1-5 : 마지막 매크로블록까지 순서 1-4을 반복한다.
- 순서 1-6 : 마지막 프레임까지 순서 1-4을 반복한다. 단 화면 간 프레임 부호화 단계이면 순서 1-1로 돌아간다.

제안하는 방법은 H.264/MPEG-4 AVC 부호화 과정 내에서 누적 확률 표를 생성 및 갱신하기 때문에 부호화되는 영상의 특성, 크기, QP값 등이 반영된 참조 확률 표를 생성하여 관련된 후보 모드들만을 고려한다. 영상마다 통계적 연관성이 다르기 때문에 고속화 되는 정도는 다르며 보통 후보 모드의 수는 4-6 사이에서 결정이 되며, 공간적 연관성이 높은 영상의 경우 최대 7개까지의 후보 모드를 감소시킨다.

마지막으로 앞서 언급한 움직임 벡터 및 블록 크기 별 최종 후보 모드 선택 상관도에 의한 두 가지 화면 내 예측 고속화 방법을 포괄적으로 정리하면 그림 5와 같다.

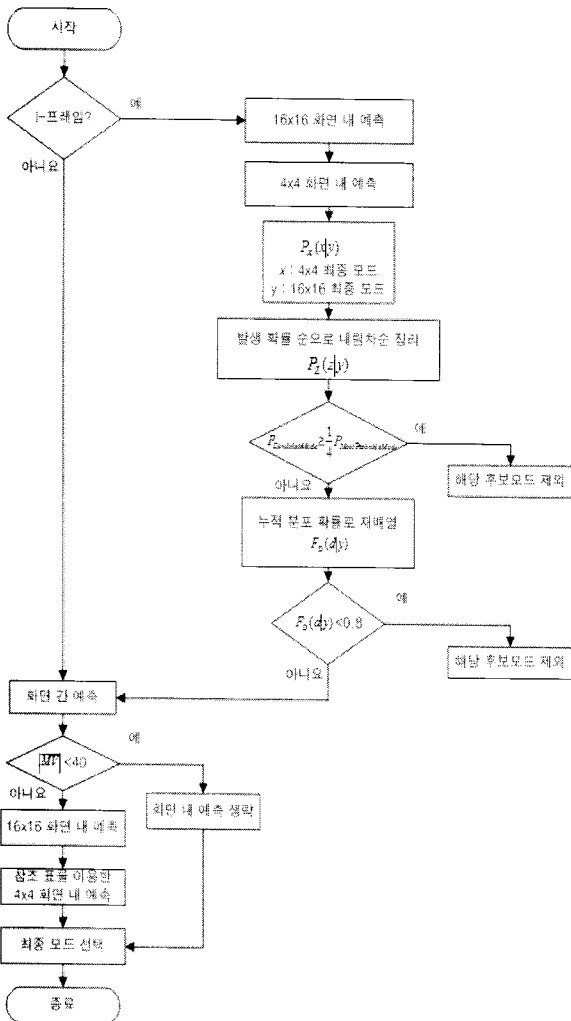


그림 5. 제안하는 방법의 순서도  
Fig.5. Flowchart of proposed algorithm

### IV. 실험 결과

실험을 위해 H.264/MPEG-4 AVC의 참조 소프트웨어인 JM 11.0<sup>[8]</sup>을 사용하였다. 총 6개의 CIF 영상 (*News*, *Hall*, *Coastguard*, *Mobile*, *Flower*, *Football*)들을 사용하였으며, IPPPP구조와 I의 주기가 5인 IPPPPIPPP(GOP 5) 구조에 대한 실험을 분리하여 수행하였다. 실험은 모든 실험 영상에 대해 각각 200프레임을 부호화하였으며, RDO ON(High Complexity Mode)의 조건으로 다양한 QP (28, 32 and 36)

값에 대해 수행되었다. 또한 참조 프레임 수는 한 장, 움직임 벡터 탐색 범위는 16화소, 탐색은 ‘UMHexagonS (Unsymmetrical-cross Multi-Hexagon-grid Search)<sup>[13]</sup>’를 사용하였으며 실험 환경은 Intel® Pentium®4 CPU 3.2GHz 1GByte 메모리 사양을 갖는 PC를 사용하였으며 실험 목적은 제안하는 방법을 사용하여 화면 내 예측 복잡도 감소를 통해 얼마나 많은 부호화 시간을 단축하였는지를 측정하는 것이고 이에 따른 화질의 열화 및 총 부호화된 비트의 양을  $\Delta P$ ,  $\Delta B$ 를 이용하여 동시에 측정하며 다음과 같이 정의한다.

$$\Delta P = \frac{PSNR_{proposed} - PSNR_{original}}{PSNR_{original}} \times 100(\%)$$

$$\Delta B = \frac{Bitrate_{proposed} - Bitrate_{original}}{Bitrate_{original}} \times 100(\%) \quad (7)$$

$$\Delta T = \frac{TotalEncodingTime_{proposed} - TotalEncodingTime_{original}}{TotalEncodingTime_{original}} \times 100(\%)$$

원래의 참조 소프트웨어 및 Park *et al.* 방법과 제안하는 방법의 비교 실험 결과를 표 4에 나타내었다.

표 4. 제안하는 방법의 실험 결과  
Table 4. Experimental results of Park's and proposed algorithm with the reference software.  
(a) IPPPPP

Sequences		QP 28			QP32			QP 36		
		$\Delta T$ (%)	$\Delta P$ (dB)	$\Delta B$ (%)	$\Delta T$ (%)	$\Delta P$ (dB)	$\Delta B$ (%)	$\Delta T$ (%)	$\Delta P$ (dB)	$\Delta B$ (%)
News	Park's	-14.76	-0.03	0.76	-14.35	-0.04	0.77	-12.81	-0.05	1.14
	Proposed	-48.18	-0.01	0.66	-46.54	-0.03	0.63	-44.24	-0.05	1.14
Hall	Park's	-14.79	0.01	0.83	-14.12	0.01	0.67	-13.21	-0.04	0.90
	Proposed	-47.74	0.01	0.7	-46.48	0.02	0.64	-44.92	-0.02	0.43
Mobile	Park's	-15.98	-0.01	0.01	-14.49	0.00	0.22	-13.74	0.01	0.14
	Proposed	-49.96	-0.01	-0.1	-47.30	0.00	0.09	-44.44	0.01	0.12
Coastguard	Park's	-16.72	0.00	0.10	-14.77	-0.01	0.00	-13.22	0.01	0.07
	Proposed	-47.00	-0.01	0.17	-43.50	-0.02	0.25	-40.75	0.00	0.17
Flower	Park's	-15.95	0.00	0.06	-14.92	-0.01	-0.07	-13.67	-0.01	-0.01
	Proposed	-51.24	0.00	0.00	-49.08	-0.01	-0.09	-46.22	0.00	0.12
Football	Park's	-13.49	-0.06	1.93	-11.99	-0.06	1.74	-10.67	-0.08	1.30
	Proposed	-30.46	-0.09	1.68	-30.08	-0.11	1.86	-31.87	-0.12	1.81
평균	Park's	-15.28	-0.02	0.62	-14.11	-0.02	0.56	-12.89	-0.03	0.59
	Proposed	-45.76	-0.02	0.52	-43.83	-0.03	0.56	-42.07	-0.03	0.63

(b) IPPPPIPPPP (GOP 5)

Sequences		QP 28			QP 32			QP 36		
		$\Delta T$ (%)	$\Delta P$ (dB)	$\Delta B$ (%)	$\Delta T$ (%)	$\Delta P$ (dB)	$\Delta B$ (%)	$\Delta T$ (%)	$\Delta P$ (dB)	$\Delta B$ (%)
News	Park's	-15.83	-0.06	4.95	-14.84	-0.06	5.36	-14.33	-0.06	6.35
	Proposed	-39.31	0.00	0.08	-38.86	-0.01	0.14	-37.39	-0.01	0.18
Hall	Park's	-15.91	-0.02	3.44	-14.98	-0.03	4.82	-14.45	-0.06	5.07
	Proposed	-39.26	0.00	0.07	-38.86	0.00	0.05	-38.02	0.00	0.06
Mobile	Park's	-17.35	-0.04	1.56	-15.78	-0.05	2.37	-14.69	-0.05	3.54
	Proposed	-41.84	0.00	0.00	-39.69	0.00	0.00	-35.51	0.00	-0.01
Coastguard	Park's	-18.50	-0.02	1.41	-16.64	-0.03	2.01	-10.01	-0.02	-2.78
	Proposed	-39.24	0.00	0.01	-37.07	-0.01	0.02	-34.84	-0.01	-0.05
Flower	Park's	-16.90	-0.06	1.08	-15.99	-0.06	1.55	-14.83	-0.07	2.34
	Proposed	-42.53	0.00	0.00	-41.30	0.00	0.00	-39.26	0.00	0.01
Football	Park's	-15.71	-0.07	3.24	-13.61	-0.08	3.01	-12.05	-0.09	3.00
	Proposed	-29.11	-0.08	1.19	-27.89	-0.09	1.25	-27.83	-0.10	1.01
평균	Park's	-16.7	-0.05	2.61	-15.31	-0.05	3.19	-13.40	-0.06	2.92
	Proposed	-38.55	-0.01	0.23	-37.28	-0.02	0.24	-35.48	-0.02	0.2

Park et al.의 방법은 16x16 화면 내 예측 결과에 따라 8x8, 4x4 화면 내 예측 고속화를 하였지만 제안하는 방법은 8x8 화면 내 예측 고속화를 다루고 있지 않으므로 정확한 비교를 위해 Park et al.의 방법도 4x4 화면 내 예측 고속화만 구현하여 결과를 도출하였으며 'the most probable mode check condition'은 사용하지 않았다<sup>7)</sup>. 제안하는 고속화 방법은 화면 내 프레임에서의 화면 내 예측들 간의 최종 후보 모드들의 상관관계를 통계적으로 표현하므로 화면 내 프레

임이 한 장인 IPPPP 구조에서는 영상의 방향적 패턴 변화를 제대로 감지하지 못하여 울-왜곡 성능 감소를 유발할 수도 있으나 표 4(a)의 실험 결과로 볼 때 무시할 수 있는 수준의 평균 0.03dB의 PSNR 감소와 0.63% 미만의 비트율 증가를 보이고 있다. GOP 5 구조인 표 4(b)의 실험 결과에서는 화면 내 프레임을 주기 5로 포함시켜 이로 인한 확률 참조 표의 갱신으로 인하여 부호화 시간의 단축 성능이 다소 감소하지만 여전히 35% 이상의 부호화 시간 감소 성능을 보이고 있으며 이는 Park et al.에 의해 제안된 방법보다 우월한 성능을 보이고 있다. 앞서 언급하였지만 제안하는 방법은 울-왜곡 성능을 유지할 수 있도록 영상의 특성에 따른 적응적인 고속화를 하고 있어 Football과 같이 움직임 벡터의 크기가 크고 16x16 화면 내 예측 결과와 4x4 화면 내 예측 결과의 방향적 상관성이 적은 영상에 있어선 식 (6)의 80%에 도달하기까지 다른 영상들에 비해 많은 후보 모드들을 포함시켜야 하며 이로 인해 부호화 시간 단축 효과가 줄어들고 울-왜곡 측면에서 PSNR 감소는 약 0.11dB, 비트율 증가는 2% 미만으로 나타나고 있다. 마지막으로 Coastguard CIF 크기 영상 100 프레임에 대한 QP 24, 28, 32, 36에서의 울-왜곡 곡선을 나타낸다. 여기에는 고속 고화질 모드방식(Fast High Complexity Mode)의 조건을 포함한다. 그림 6과 표 5는 실험 결과를 울-왜곡 성능 측면과 부호화 시간 단축 측면에서 잘 나타내어 주고 있다. 네 가지 경우 모두에 대해서 울-왜곡 성능 차이는 없다고 해도 무방할 정도로 동일 선상에서 모

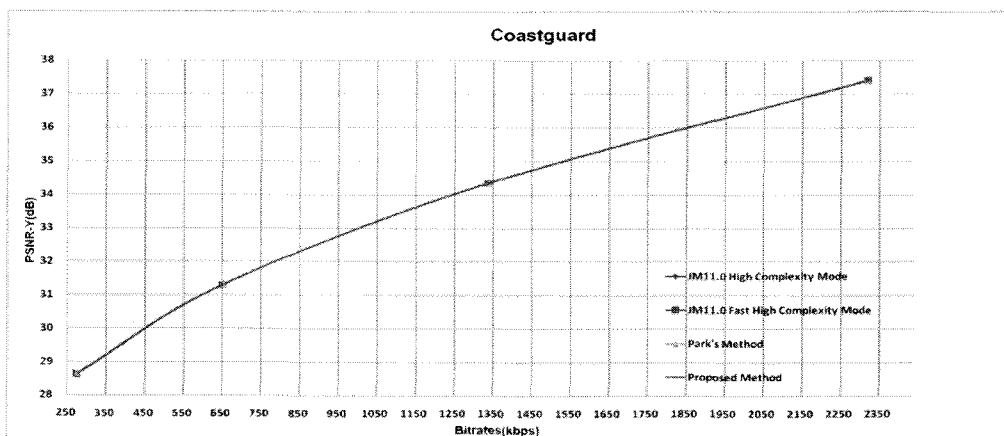


그림 6. 울-왜곡 성능 비교  
Fig.6. Comparison of RD Performance

든 곡선이 위치한다. 하지만 이에 따른 시간 단축 효과는 표 5와 같이 제안하는 방법이 두드러진다. FHCM (Fast High Complexity Mode)가 CBP값 등을 이용하여 화면 간 예측(inter prediction)에 대하여도 조기종료 알고리즘을 사용하는데 반해 제안하는 알고리즘은 화면 내 예측만의 고속화임에도 QP 36인 경우를 제외하고는 부호화 시간 단축 측면에서 더 뛰어난을 확인할 수 있다.

표 5. 부호화 시간 단축 성능 비교  
Table 5. Comparison of encoding time saving

Condition	QP24	QP 28	QP 32	QP 36
FHCM	-24.39%	-22.40%	-29.63%	-44.04%
Park's Method	-18.09%	-16.84%	-15.61%	-13.11%
Proposed Method	-49.34%	-46.71%	-43.65%	-40.34%

#### IV 결론 및 향후 과제

본 논문에서 제안하는 방법은 화면 간 프레임에서의 화면 내 예측 부호화의 연산량 감소를 위해 화면 내 예측에서 도출된 움직임 벡터의 크기를 이용하여 임계값을 정한 후 화면 내 예측 실행 여부를 조기에 결정한다. 화면 예측이 필요한 것으로 결정되면 16x16 화면 내 예측 결정된 최종 후보 모드에 따른 4x4 블록 크기 예측에서의 결정 모드들의 조건부 확률 표를 화면 내 프레임 부호화에서 생성 및 저장하여 동일 GOP내의 화면 내 예측에서 활용하며 발생 확률이 낮은 후보 모드들에 대해선 부호화 대상에서 제외함으로써 부호화기의 복잡도를 감소하게 된다. 정리하면 화면 간 프레임에서 화면 내 예측 블록이 최종 모드로 결정되는 비율은 영상의 내용 특성과 QP등에 따라 상이하므로 화면 내 예측의 필요 여부를 움직임 벡터값을 이용하여 정해진 임계값을 통해 판단한 후, 매크로블록 단위에서 화면 내 예측을 조기에 종료함으로써 많은 연산량 및 부호화 시간의 감소 효과를 얻을 수 있다. 또한 화면 내 예측의 조기 종료에 실패하더라도 다양한 실험을 통해 도출된 특정 조건을 만족하는 후보 모드들만을 예측 부호화에 사용하여 원래의 총 9가지의 후보 모드들로 예측을 수행하는 경우에 비

해 부호화 시간을 크게 단축시킨다. 마지막으로 제안하는 방법을 이용할 경우 거의 모든 실험 영상에서 만족할 만한 부호화 시간 단축과 윌-왜곡 성능 유지를 보였으나 *Football*과 같이 카메라와 화면 내 객체의 움직임이 극도로 빠른 영상의 경우에 윌-왜곡 성능 저하가 다소 두드러지는 경향을 보이고 있다. 향후 이러한 영상의 패턴을 검출하여 현재와 같이 고정적인 움직임 벡터 임계값을 적용하지 않고 적응적인 임계값을 적용할 수 있는 방법에 대한 연구를 지속할 필요가 있다고 판단된다.

#### 참 고 문 헌

- [1] ISO/IEC 14496-10, "Information Technology-Coding of Audio Visual Objects-part 10 : Advanced Video Coding," Dec. 2003.
- [2] T. Wiegand, G.J. Sullivan, G. Bjontegard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," IEEE Trnas. Circuit Syst. Video Technol., vol 13, July 2003, pp. 560-570.
- [3] Halbacg T. "Performance comparison: H.26L intra-coding vs JPEG2000," Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, Klagenfurt, Austria, July 2002.
- [4] D. Sim and Y. Kim "Context-Adaptive Mode Decision for Intra Block Coding in H.264/MPEG-4 Part10," Real Time Imaging, vol.11, 2005, pp.1-6.
- [5] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, D. Wu, S. Wu "Fast Intra Mode Decision Algorithm for H.264/MPEG-4 AVC Video Coding," IEEE 2004 International Conference on Image Processing(ICIP), vol. 2, pp781-784, Oct. 2004.
- [6] M. Hwang, J. Cho, J. Kim, J. Kim and, S. Ko "Fast Intra Prediction Mode Selection Scheme Using Temporal Correlation in H.264," TENCON 2005 2005 IEEE Region 10, Nov. 2005.
- [7] J. Park, H. Song "Fast Intra Prediction Mode Decision for H.264/AVC Encoders," IEEE Transactions on Engineering, Computing and Technology, vol.13, pp 51-55, May 2006
- [8] Joint Model (JM) - H.264/AVC Reference Software, [http://iphome.hhi.de/suehring/tmi/download/old\\_jm](http://iphome.hhi.de/suehring/tmi/download/old_jm)
- [9] 나태영, 이범식, 김문철, 함상진, 박창섭, 박근수 "영상 내용 특성과 주위 블록 모드 상관성을 이용한 H.264/MPEG-4 AVC 화면 간 프레임에서의 화면 내 예측 부호화 결정 방법과 화면 내 예측 고속화 방법", 방송공학회논문지, 2007년 제 12권 제 6호, pp 611-623.
- [10] C.Kim, J.Kuo, "Feature-based Intra/Inter Coding Mode Selection for H.264/AVC," IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, NO. 11, Nov.2006
- [11] G. J. Sullivan, P. Topiwala and A. Luthra "The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions," SPIE Conf. on applications of digital image processing XXVII, vol. 5558, pp. 53-74, Aug. 2004
- [12] G. J. Sullivan, T.McMahon, T.Wiegand and A. Luthra, "Draft

Text of H.264/AVC Fidelity Range Extensions Amendment to ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC," ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16 Joint Video Team Document JVT-L047, July 2004.

[13] Z. Chen, P. Zhou and Y. He, "Fast Integer Pel and Fractional Pel Motion Estimation for JVT," ISO/IEC JTC1/SC29/WG11, JVT-F017, Dec. 2002

[14] C. Yang, L. Po and W. Lam, "A Fast H.264 Intra Prediction Algorithm Using Macroblock Properties," IEEE 2004 International Conference on Image Processing (ICIP), vol. 1, pp 461-464, Oct. 2004

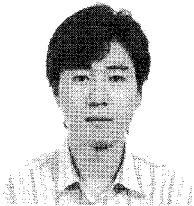
[15] S. Chun, J. Yoon and S. Sull, "Efficient Intra Prediction Mode Decision for H.264 Video," PCM 2005, Part I, LNCS 3767, pp 168-178, 2005

---

저 자 소 개

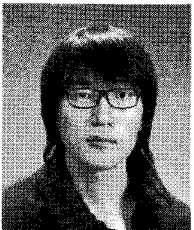
---

**나 태 영**



- 2005년 2월 : 아주대학교 전자공학부 학사
- 2006년 6월 ~ 현재 : 한국정보통신대학교 공학부 석사 과정
- 2004년 11월 ~ 2006년 4월 : LG전자 DM사업본부 AV사업부 Audio 연구실 음향파트 연구원
- 주관심분야 : H.264/MPEG-4 AVC, 영상압축, Networked Video, Multimedia Signal Processing

**정 윤 식**



- 2008년 2월 : 한국정보통신대학교 공학부 학사
- 2008년 3월 ~ 현재 : 한국정보통신대학교 공학부 석사 과정
- 주관심분야 : H.264/MPEG-4 AVC, Scalable Video Coding

**김 문 철**



- 1989년 : 경북대학교 학사
- 1992년 : University of Florida, Electrical and Computer Engineering, 석사
- 1996년 : University of Florida, Electrical and Computer Engineering, 박사
- 1997년 1월~2001년 2월 : 한국전자통신연구원 선임연구원, 방송미디어연구부 실감영상연구팀/영상미디어 연구팀 팀장
- 2001년 2월 ~현재 : 한국정보통신대학교 부교수
- 주관심분야 : 차세대 비디오 부호화, 모바일 영상 통신, 영상 및 신호처리, 패턴인식, SuperHDTV/UDTV, IPTV personalization

**함 상 진**



- 1996년 2월 : 연세대학교 전자공학과 학사
- 1998년 8월 : 연세대학교 전자공학과 석사
- 2001년 7월~현재 : 한국방송 방송기술연구팀, 현 U콘텐츠처리보호 선임연구원
- 주관심분야 : 고효율 영상압축, 콘텐츠 저작권 보호 기술, 영상 신호 처리

---

저 자 소 개

---



**박 창 섭**

- 1980년 : 한양대학교 전자공학과 학사
- 1982년 : 한양대학교 산업대학원 석사
- 2001년 : 한국과학기술원 정보 및 통신공학과 박사
- 1981년 ~ 현재 : 한국방송 방송기술연구팀
- 주관심분야 : 영상신호압축, 컴퓨터 비전, 입체 TV



**박 근 수**

- 1982년 : 서울대학교 제어계측공학과 학사
- 1984년 : KAIST 전기전자공학과, 석사
- 1984년 3월 ~ 현재 : 한국방송 방송기술연구소 근무, 현 U콘텐츠처리보호 수석연구원
- 주관심분야 : HD/SD 영상효과 및 처리 방송제작기술, DTV 방송콘텐츠 보호기술, 유비쿼터스 콘텐츠 처리기술