

H.264/AVC에서 영상 복잡도를 이용한 고속 인터 블록 모드 결정

정희원 김 성 희*, 종신회원 오 정 수**

Fast Inter Block Mode Decision Using Image Complexity in H.264/AVC

Seong-hee Kim** *Regular Member*, Jeong-su Oh* *Lifelong Member*

요 약

동영상 표준인 H.264/AVC에서 가변 블록 모드 알고리즘은 압축 성능을 향상시키나 다양한 블록 모드와 모드 결정을 위해 방대한 계산을 필요로 한다. 한편, 결정된 인터 블록 모드들은 블록 영상의 복잡도에 의존하여 복잡한 매크로블록일수록 블록 크기는 더 작아지고 있다. 본 논문은 고속 인터 블록 모드 결정 알고리즘을 제안한다. 제안된 알고리즘은 영상 복잡도를 이용해 유효 인터 블록 모드를 결정 가능성이 큰 블록 모드로 제한하고, 유효 블록 모드들에서만 움직임 추정 및 율-왜곡 최적화를 수행한다. 거기에 16×16 블록 모드만을 갖는 유효 블록 모드들에는 고속 움직임 추정 알고리즘인 PDE (partial difference elimination)를 적용시키고 있다. 제안된 알고리즘의 성능 평가를 위해 참조 소프트웨어 JM 9.5에서 화질, 비트율, 부호화 시간이 기존 알고리즘과 비교되었다. 실험 결과는 제안된 알고리즘이 평균 움직임 추정 시간의 약 24.12%를 감소시키며 화질과 비트량을 각각 -0.02dB와 -0.12% 정도로 유지시킬 수 있는 것을 보여주었다.

Key Words : Motion estimation, Block mode, Image complexity, H.264/AVC

ABSTRACT

In video coding standard H.264/AVC, variable block size mode algorithm improves compression efficiency but has need of a large amount of computation for various block modes and mode decision. Meanwhile, decided inter block modes depend on the complexity of a block image, and then the more complex a macroblock is, the smaller its block size is. This paper proposes fast inter block mode decision algorithm. It limits valid block modes to the block modes with a great chance for decision using the image complexity and carries out motion estimation rate-distortion optimization with only the valid block modes. In addition to that, it applies fast motion estimation PDE to the valid block modes with only the 16×16 block mode. The reference software JM 9.5 was executed to estimate the proposed algorithm's performance. The simulation results showed that the proposed algorithm could save about 24.12% of the averaged motion estimation time while keeping the image quality and the bit rate to be -0.02dB and -0.12% on the average, respectively.

※ 이 논문은 2007년도 부경대학교 연구년 교수 지원사업에 의하여 연구되었음 (PS-2007-025)

* 코아로직(주), ** 부경대학교 화상정보공학부 이미지시스템공학과 디지털영상처리 연구실 (ojs@pknu.ac.kr)

논문번호 : KICS2008-04-153, 접수일자 : 2008년 4월 8일, 최종논문접수일자 : 2008년 10월 14일

I. 서 론

동영상 표준인 H.264/AVC는 부호화 효율 개선을 위해서 가변 블록 크기, 1/4 화소 정확도, 다중 참조 프레임의 움직임 보상, DCT 기반의 정수 변환, CABAC 등의 알고리즘들을 채택하고 있다^{1,2,3}. H.264/AVC는 이들에 의해 기존 표준인 MPEG-4보다 30~40%, H.263보다 40~50%의 압축 효율을 보이고 있다². 그래서 H.264/AVC는 최근 개발되는 휴대이동방송과 HD-DVD 등에 채택되고 있어 많은 관련자들의 관심이 집중되고 있다³. 움직임 추정과 관련해 새롭게 적용된 기술들은 H.264/AVC의 성능을 크게 향상시켰으나 H.264/AVC 부호기의 복잡도를 크게 증가시켜 실시간 구현을 어렵게 하고 있다. 그래서 H.264/AVC 발표 이후에 많은 고속 알고리즘들이 발표되고 있다^{4,5,6}.

H.264/AVC는 기존 동영상 표준인과 달리 인터 블록 모드 (inter block mode), 인트라 블록 모드 (intra block mode), SKIP 모드 같이 다양한 형태와 다양한 크기의 블록 모드를 채택하고 있고, 최적의 부호화 효율을 얻기 위해 율-왜곡 최적화 (rate distortion optimization) 알고리즘이 수행되고 있다^{1,2,3}. 율-왜곡 최적화는 재생된 영상의 화질과 부호화를 위해 발생하는 비트량을 고려하고 있어 부호화는 물론 복호화가 수행되어야 하고 이는 방대한 계산을 요구하게 된다. 한편 율-왜곡 최적화는 모든 매크로블록들에 동일하게 처리되는데 실제로는 결정된 인터 블록 모드들은 블록 영상의 내용에 의존적이다. 단순한 블록 영상은 큰 블록 모드들로 결정되고, 복잡한 블록 영상은 작은 블록 모드들로 결정되고 있다. 따라서 본 논문은 매크로블록 영상의 복잡도에 의존하여 인터 블록 모드들에서 결정 가능성이 큰 블록 모드들을 유효 블록 모드들로 제한하여 이들에 대해서만 움직임 추정은 물론 율-왜곡 최적화를 수행하여 계산량을 줄이고 있다.

제한된 알고리즘의 성능을 평가하기 위해 H.264/AVC 참조 소프트웨어 JM 9.5가 수행되었다. 모의 실험에서 제안된 알고리즘은 기존 알고리즘과 비교하여 평균적인 화질과 비트율에서 무시할 정도의 저하가 있었지만 계산량을 크게 감소시키고 있다.

II. H.264/AVC의 인터 블록 모드 결정

본 장에서는 H.264/AVC에서 수행되는 가변 블록 크기의 움직임 추정과 인터 블록 모드를 결정하는 방법에 대해 기술한다.

2.1 가변 블록 크기의 움직임 추정

인터 프레임인 P/B-프레임을 위해, MPEG-1, MPEG-2에서는 16×16 크기의 고정 블록을 사용하였고, MPEG-4에서는 16×16과 8×8의 두 종류의 가변 블록을 사용하여 움직임 추정을 수행하였지만^{7,8}, H.264/AVC에서는 성능을 더욱 향상시키기 위해 더 다양한 크기의 가변 블록을 사용한다. H.264/AVC에서 매크로블록의 휘도 성분(Y, 16×16)을 위해 P16×16, P16×8, P8×16, P8×8, I16, I4, SKIP 모드 (mode) 중 한 모드가 결정되고, P8×8 모드가 결정되면 4개의 8×8 블록들은 독립적으로 P8×8, P8×4, P4×8, P4×4 모드 중 한 모드가 다시 결정된다¹¹. P* 모드는 인터 블록 모드이고, I* 모드는 인트라 블록 모드이고, SKIP 모드는 16×16 블록에서 움직임 벡터가 예측 움직임 벡터와 같고 예측 오차가 충분히 작은 경우 결정되는 모드이다. 본 논문에서는 움직임 추정이 수행되는 휘도 성분의 인터 블록 모드를 대상으로 하고 있고 이들을 그림 1에 보여주고 있다.

가변 블록 크기의 움직임 추정은 모든 인터 블록 모드들에서 최소 MVcost를 갖는 블록을 찾는 것이다. MVcost는 식 (1)과 같이 계산된다.

$$MVcost = SAD + \lambda_{Motion} \times R_{mv}(MV - PMV) \quad (1)$$

여기에서 SAD (sum of absolute difference)는 현재 블록과 후보 블록사이의 절대 정합 오차의 합이고, λ_{Motion} 은 Lagrangian 승수로 양자화 파라미터 (QP, quantization parameter)에 대응하기 위한 상수이다. $R_{mv}()$ 는 움직임 벡터 정보를 위한 비트량으로 현재 블록의 움직임 벡터 (MV)와 예측 움직임 벡터 (PMV, predicted MV)의 차에 대한 함수이다.

다양한 크기의 블록들에 대해 독립적인 블록 정합을 수행하면 방대한 계산량이 요구될 것이다. 그래서 JM 9.5의 일반적인 움직임 추정에서는 매크로블록을 최소 움직임 추정 블록인 4×4 블록 단위로 블록 정합 오차를 계산한 후, 블록 모드에 따라서 블록에 포함되는 4×4 블록들의 정합 오차를 더하여

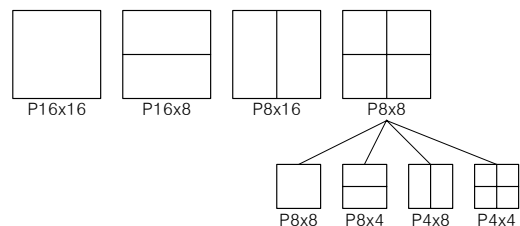


그림 1. 움직임 추정이 수행되는 인터 블록 모드들

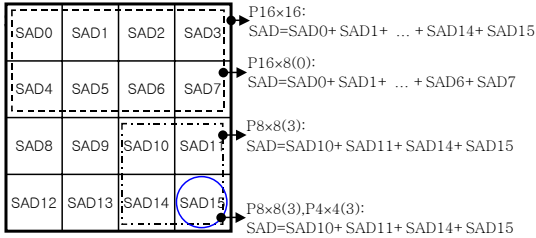


그림 2. 매크로블록에서 4x4 단위의 블록 정합 오차

각 블록 모드의 정합 오차를 얻는다. 그림 2는 4x4 단위로 계산된 블록 정합 오차와 이를 이용한 일부 블록 모드들의 정합 오차를 계산하는 방법을 보여 주고 있다.

2.2 가변 블록 크기의 움직임 추정

H.264/AVC에서 궁극적으로 결정되어지는 최적의 인터 블록 모드는 모든 블록 모드에서 움직임 추정을 수행하여 최적 후보 블록을 결정하고 나서 윌-왜곡 최적화를 수행하여 결정되어진다^{1,2,3}. JM 9.5의 윌-왜곡 최적화는 식 (2)를 이용하여 모든 블록 모드에서 최소의 RDcost를 갖는 모드를 찾는 것으로 매크로블록의 부호화를 위한 최종 블록 모드를 결정하는 것이다.

$$RDcost = SSD + \lambda_{Mode} \times R_{REC}(header, redual) \quad (2)$$

여기에서 SSD (sum of square difference)는 현재 영상과 복원된 영상 사이의 오차를 나타내고, λ_{Mode} 은 Lagrangian 승수로 양자화 파라미터에 대응하기 위한 상수이다. $R_{REC}()$ 는 매크로블록을 재구성하기 위해 요구되는 헤더, 예측 오차를 위한 비트량이다. 재구성을 위해 복호화 과정이 수행되어야 하고, 이를 모든 블록 모드들에 적용할 때 큰 계산이 요구된다. 블록 모드와 부호화 사이의 관계는 큰 블록 크기의 모드를 결정하면 움직임 벡터를 포함한 부가 정보량이 적지만 예측 오차에 대한 정보가 많아 질 수 있다. 반대로 작은 블록 크기의 모드를 결정하면 예측 오차에 대한 정보가 적어 질 수 있는데 움직임 벡터를 포함한 부가 정보량이 많아진다. 일반적으로 영상의 움직임이 작고 물체가 클 경우 큰 블록 모드로, 영상의 움직임이 복잡하고 물체가 작을 경우 작은 블록 모드로 결정되는 것이 효과적이다.

III. 가변 인터 블록 모드의 성능 분석

가변 블록 모드의 성능을 분석하기 위해 추후 모

의 실험에서 기술하는 실험 환경에서 JM 9.5가 수행되었다. 표 1은 고정 블록 크기와 가변 인터 블록 크기에서 각각 화질 (PSNR), 비트율, 부호화 시간을 비교하고 있다. 가변 블록 크기를 사용하면 화질은 평균 0.47dB 정도 향상되고, 비트율은 평균 14.11% 정도 감소되지만, 부호화 시간은 QCIF 영상에서는 약 1.5배로 증가하고, CIF 영상에서는 약 1.2배로 증가한다. 비트율은 프레임 주파수와 영상 크기의 영향을 받으므로 단순히 비교할 수 없다. 그래서 상대적인 비교를 위해 10Hz QCIF 기준으로 비트율을 환산한 추정 비트율을 괄호 안에 보여주고 있다. 표 2는 각 시퀀스에서 최종적으로 결정되는 전체 블록 수에 대한 블록 모드의 비율을 나타내고 있다. Taylor series expansion를 통해 증명된 것처럼 복잡한 영상은 큰 정합 오차를 발생시키고, 이로 인해 비트량도 많아진다. 따라서 표 1에서 비트량이 많은 Mobile, Tempete, Forman 시퀀스들은 복잡한 영상이고, 비트량이 적은 Container와 News 시퀀스들은 단순한 영상이라 할 수 있다. 표 2에서는 복잡한 영상일수록 작은 블록 크기의 모드가 많이 결정된다는 것을 알 수 있다. 실제 표 1에서 비트율이 큰 순서와 표 2에서 8x8 이하의 작은 블록 모드들이 많은 순서가 일치하고 있다. 또한 모든 시퀀스들에서 작은 블록 모드 보다는 큰 블록

표 1. 고정 블록 크기와 가변 블록 크기에서 성능 비교

	PSNR(dB)		bit rate(kbit/s)		codingtime(sec)	
	fixed	var	fixed	var	fixed	var
Container	32.87	33.28	16.42(16.42)	13.48(13.48)	182.38	273.37
News	33.19	33.81	32.19(32.19)	28.01(28.01)	182.52	273.05
Foreman	32.81	33.45	50.01(50.01)	44.56(44.56)	184.04	280.83
Paris	32.01	32.43	216.59(36.10)	181.39(30.23)	3767.33	4511.74
Silent voice	32.52	33.02	39.0(26.00)	34.56(23.04)	278.03	415.67
Mobile	30.21	30.56	825.3(68.77)	732.97(61.08)	7654.19	9236.31
Tempete	31.33	31.69	806.25(55.52)	545.75(45.48)	6637.31	7980.03

(5 reference pictures, QP:32, var:variable)

표 2. 블록 모드의 비율 (%)

	SKIP, P16x16	P16x8, P8x16	P8x8	P8x4, P4x8	P4x4	intra
Container	83.65	9.65	3.28	1.53	0.28	1.60
News	75.74	11.55	5.87	3.78	0.58	2.48
Foreman	52.05	25.98	9.31	5.17	0.56	6.94
Paris	75.91	9.65	7.14	4.77	0.97	1.55
Silent voice	76.64	12.75	5.27	2.49	0.21	2.64
Mobile	45.23	33.26	12.96	7.31	1.05	0.19
Tempete	58.63	26.56	8.12	4.74	0.60	1.35
average	66.84	18.49	7.42	4.26	0.61	2.39

(5 reference pictures, QP:32)

모드가 훨씬 많이 결정된다. 큰 블록 모드로 결정될 블록에서 작은 블록 모드에 대한 비교는 이득 없이 계산량을 낭비하는 것이다.

IV. 제안된 고속 인터 블록 모드 결정

본 장에서는 영상 복잡도를 이용해 H.264/AVC의 최적의 인터 블록 모드를 결정하기 위한 고속 알고리즘을 제안한다.

4.1 영상 복잡도

영상 복잡도는 블록 영상에서 밝기 변화의 정도를 나타내는 척도이고, 이는 표 1과 표 2에 보이는 것처럼 블록 모드 결정에 큰 영향을 주고 있다. 즉 단순한 블록 영상일수록 큰 블록 모드가 결정될 확률이 높고, 복잡한 블록 영상일수록 작은 블록 모드가 결정될 가능성이 높다. 본 논문에서는 영상 복잡도를 표현하기 위해 두 가지 정보를 활용한다.

첫번째는 블록 영상의 공간적인 밝기 변화를 나타내는 공간적 영상 복잡도 (IC)로 식 (3)에 의해 계산된다.

$$IC = \sum_{i=1}^N \sum_{j=1}^N (|f(i,j) - f(i+1,j)| + |f(i,j) - f(i,j+1)|) \quad (3)$$

여기서 $f(i,j)$ 와 N 은 각각 (i,j) 의 화소 밝기와 블록의 크기이다. 본 논문에서 매크로블록의 영상 복

표 3. 블록 모드들에서 평균 IC

	SKIP, P16x16	P16x8, P8x16	P8x8	P8x4, P4x8	P4x4	intra
Container	2757.08	6598.62	8284.53	8875.17	10493.78	603.03
News	3139.03	3486.16	5080.75	5433.10	6565.51	3703.61
Foreman	2035.34	2733.71	3220.84	3574.49	4137.08	1208.78
Paris	4715.16	4186.21	6430.97	6913.86	7762.86	2749.34
Silent voice	2012.86	1851.89	2456.22	2628.71	3238.21	1175.10
Mobile	6239.64	8128.55	9380.90	9881.53	10828.83	935.41
Tempete	3076.16	5173.27	7247.08	7615.66	8446.97	1435.56

(5 reference pictures, QP:32)

표 4. 블록 모드에서 평균 MVcost

	PSNR(dB)		bit rate(kbit/s)		codingtime(sec)	
	fixed	var	fixed	var	fixed	var
Container	32.87	33.28	16.42(16.42)	13.48(13.48)	182.38	273.37
News	33.19	33.81	32.19(32.19)	28.01(28.01)	182.52	273.05
Foreman	32.81	33.45	50.01(50.01)	44.56(44.56)	184.04	280.83
Paris	32.01	32.43	216.59(36.10)	181.39(30.23)	3767.33	4511.74
Silent voice	32.52	33.02	39.0(26.00)	34.56(23.04)	278.03	415.67
Mobile	30.21	30.56	825.3(68.77)	732.97(61.08)	7654.19	9236.31
Tempete	31.33	31.69	506.25(55.52)	545.75(45.48)	6637.31	7980.03

(5 reference pictures, QP:32, var:variable)

잡도를 계산하기 위해 16x16 블록을 8x8 크기의 부분 블록들로 나눈다. 그리고 각 부분 블록에서 식 (3)을 이용해 영상 복잡도를 구하고, 이중 최대값을 매크로블록의 영상 복잡도로 결정한다. 이는 매크로블록 전체에 작은 변화보다 부분적인 큰 변화를 중시하기 위한 것이다. 표 3은 각 시퀀스들과 블록 모드들에서 평균 IC를 보여 주고 있다. 복잡한 배경을 갖는 Paris와 Silent voice 시퀀스의 P16x8, P8x16 모드를 제외하고 모든 시퀀스들에서 앞에서 기술한 것처럼 작은 크기의 블록 모드일수록 큰 영상 복잡도를 갖고 있다.

두번째는 식 (1)에 보여준 MVcost를 이용하여 영상 복잡도를 나타낸다. Taylor series expansion에 의해 복잡한 영상은 큰 정합 오차를 갖고, MVcost는 정합 오차 항을 포함하고 있어 MVcost는 영상 복잡도를 나타낼 수 있다. 한편 MVcost는 IC와 달리 추가적인 계산을 요구하지 않고 예측의 정확성을 포함하는 장점이 있다. 표 4는 각 시퀀스들과 블록 모드들에서 평균 MVcost를 보여주고 있다. 모든 시퀀스들에서 작은 크기의 블록 모드일수록 큰 MVcost를 갖고 있다.

따라서 IC와 MVcost는 영상 복잡도를 적절히 나타내고 있다고 할 수 있고, 이들은 블록 모드를 분류할 수 있는 적절한 정보라 할 수 있다.

4.2 영상 복잡도를 이용한 고속 인터 블록 모드 결정

영상 복잡도와 블록 모드 사이에 큰 상관 관계가 있고, 결정될 블록 모드를 제한할 수 있다면 블록 모드 결정을 위한 계산량을 크게 줄일 수 있는 것을 사전 실험을 통해 확인하였다. 그래서 본 논문은 영상 복잡도를 이용해 움직임 추정과 울-왜곡 최적화를 최적 블록 모드 가능성이 큰 블록 모드들로 제한하여 수행하고 있고, 특히 P16x16 모드로 제한된 블록에는 고속 움직임 추정 알고리즘인 PDE를 적용하여 계산량을 감소시키는 고속 블록 모드 결정 알고리즘을 제안한다.

제안된 알고리즘은 두 종류의 영상 복잡도를 이용해 두 단계에 걸쳐 모드를 제한하고 있다. 첫번째 단계는 매크로블록 단위로 IC를 계산하여 임계값 (TH_{IC}) 이하이면 그 블록을 P16x16 모드와 SKIP 모드로 제한하여 고정 크기 블록처럼 다루어 계산량을 줄이고, 또한 모드 제한으로 나머지 모드는 고려되지 않아 그림 2와 같은 부분적 SAD가 재사용되지 않으므로 고속 움직임 추정 알고리즘 PDE를 적용할 수 있어 추가적으로 계산량을 줄일 수 있다.

PDE는 일정 주기마다 그 때까지 계산된 부분 블록 정합 오차와 최소 블록 정합 오차를 비교하여 불가능한 후보 블록을 조기에 제거하여 성능 저하 없이 움직임 추정 계산을 70%정도 줄여주는 효율적인 고속 움직임 추정 알고리즘이다^[10]. TH_{IC}은 첫 번째 P-프레임에서 선택된 블록 모드를 근거로 식 (4)와 같이 계산되고 두 번째 P-프레임부터 적용된다.

$$TH_{IC} = \begin{cases} 0.2 \times aveIC_{16 \times 16}, & \text{the case1} \\ 0.2 \times aveIC_{16 \times 8 \times 16}, & \text{the case2} \end{cases} \quad (4)$$

여기서 aveIC_{16×16}와 aveIC_{16×8×16}는 각각 P16×16 모드와 P16×8, P8×16 모드의 평균 IC로 모드가 발생할 때마다 갱신되고, case1과 case2는 P16×16 모드의 IC가 P16×8, P8×16 모드의 IC보다 큰 경우와 그렇지 않은 경우이다. 두번째 단계에서 P16×16 모드로 다시 제한할 기회가 있어 양자화 파라미터를 고려하지 않았고, 낮은 가중치 0.2와 0.15를 실험적으로 결정하였다.

두번째 단계는 첫번째 단계에서 P16×16 모드로 제한이 안된 매크로블록에 대해 MVcost를 이용하여 움직임 추정과 율-왜곡 최적화를 위한 계산량을 감소시킨다. 먼저 대상 매크로블록의 예측된 움직임 벡터 (PMV) 위치에서 P16×16 모드의 SAD를 계산하여 임계값 (TH_{PMV})와 비교한다. MVcost를 사용하지 않고 SAD를 사용한 것은 예측 움직임 벡터의 위치는 부호화를 위한 움직임 벡터의 차가 0임으로 SAD는 MVcost와 동등하게 사용되고 계산량을 줄인다. TH_{PMV}보다 작으면 아주 단순한 블록 혹은 예측이 잘된 블록으로 판단하여 그 블록은 P16×16 모드로 제한되고, 나머지 움직임을 위해 PDE가 사용된다. 최적 블록 모드는 P16×16 모드, 인트라 모드, SKIP 모드의 RDcost에 의해 결정된다. TH_{PMV}보다 크면 일반적인 가변 블록 움직임을 수행하고 단계별로 최소 MVcost (MVC_{min}) 결정하여 임계값 TH_{16×16}, TH_{16×8×16}, TH_{8×8}와 각각 비교하여 유효 블록 모드들을 결정한다. P16×16에서 MVC_{min}가 TH_{16×16} 보다 작으면 유효 블록 모드를 P16×16 모드로 제한하고, 크면 P16×8, P8×16 모드에서 MVC_{min}를 계산하고 TH_{16×8×16}와 비교하여 작으면 유효 블록 모드를 P16×16, P16×8, P8×16 모드로 제한하고, 크면 P8×8 모드에서 MVC_{min}를 계산하고 TH_{8×8}와 비교하여 작으면 유효 블록 모드를 P16×16, P16×8, P8×16, P8×8 모드로 제한하고, 크면 모든 모드들이 유효하다. 최적의 블록 모드는 유효 모드들, 인트라 블록 모드, SKIP 모드의

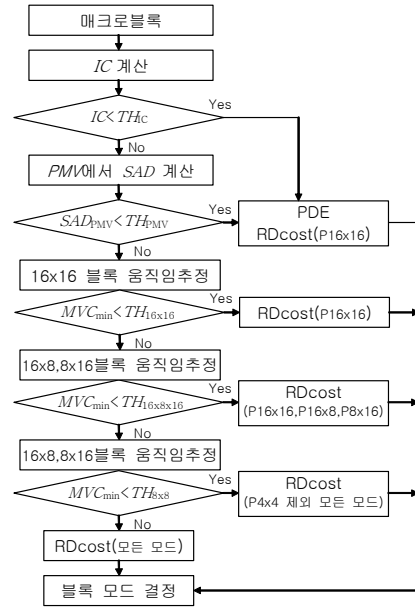


그림 3. 제안된 고속 블록 모드 결정 흐름도

RDcost를 비교하여 결정된다. 본 논문에서는 상기 임계값을 생성하기 위해 먼저 사전 연구를 통해 얻은 블록 모드와 양자화 파라미터에 따른 평균 MVcost를 초기 MVcost (MVC_{init})로 설정하였고, 이를 이용해 각각의 임계값은 다음 식과 같이 계산된다.

$$TH_{PMV} = w1 \times MVC_{init16 \times 16} \quad (5)$$

$$TH_{N \times M} = TH_{N \times M} \times w2 + MVC_{minN \times M} \times (1 - w2) \quad (6)$$

여기서 w1과 w2는 각각 실험적으로 결정된 가중치로 0.5, 0.95이고, MVC_{minN×M}는 N×M 블록 모드의 MVC_{min}이고, TH_{N×M}는 N×M 블록 모드가 발생할 때마다 갱신되고, 초기값은 MVC_{initN×M}의 75%이다.

그림 3은 본 논문에서 제안된 영상 복잡도를 이용한 고속 블록 모드 결정 알고리즘의 흐름도를 보여주고 있다. RDcost()는 인트라 모드, SKIP 모드 그리고 괄호 안의 모드에서 최소 RDcost를 갖는 모드를 찾는 것이다.

V. 모의 실험

제안된 알고리즘의 성능 평가를 위해 H.264/AVC 표준안에서 권고하는 공통 실험 조건을 참조하여 표 5의 실험 환경^[11]에서 표 6의 시퀀스들을 대상으로 H.264/AVC 참조 소프트웨어 JM 9.5가 수행되었다^[12].

표 5. 실험 환경

CPU	Intel Pentium4 2.4GHz
메모리	256MB
운영체제	Window XP Home Edition
부호기	JM 9.5
QP	28, 32, 36, 40
움직임 탐색영역	±16(QCIF), ±32(CIF)
참조 픽처 수	5
부호기 기술	베이스 라인 프로파일, 올-왜곡 최적화, 가변 크기 움직임 추정, CAVLC, 하다마드 변환부호화, B픽처 사용안함, Rs, FMO, ASO 사용안함

표 6. 실험 시퀀스들

Format	QCIF		CIF	
	10/100	15/130	15/150	30/300
images	Container, News, Foreman	Scient voice	Paris	Mobile, Tempete

표 7은 제안된 알고리즘의 성능을 PSNR, 비트량, 부호화 시간의 관점에서 보여주고 있다. ΔPSNR, ΔBits, ΔTime는 각각 JM 9.5 부호기에서 가변 블록 크기의 알고리즘이 적용된 결과와 제안된 알고리즘의 차이를 보여주는 것으로 다음 식들에 의해 계산된다.

$$\Delta PSNR = PSNR_{pro} - PSNR_{ref} \quad (7)$$

$$\Delta Bits = \frac{Bits_{pro} - Bits_{ref}}{Bits_{ref}} \times 100 \quad (8)$$

$$\Delta Time = \frac{Time_{pro} - Time_{ref}}{Time_{ref}} \times 100 \quad (9)$$

여기에서 ref는 JM 9.5 알고리즘의 실험 결과이고, pro는 제안된 알고리즘에 의한 실험 결과이다. PSNR과 비트량은 시퀀스에 따라 다소 차이가 있으나 모든 시퀀스들과 양자화 파라미터들에서 무시할 정도이다. 이들의 평균적인 성능은 각각 0.02dB의 저하와 0.12%의 증가이다. 반면에 부호화 시간은 최저 12.93%에서 최대 32.79%, 평균 24.12%의 상당한 감소량을 보여주고 있고, 이는 고정 블록 크기를 적용할 때 비슷한 시간이며 Paris, Tempete 같은 시퀀스에서는 고정 블록 크기 때 보다 적은 시간이 소요되고 있다. 이는 제안된 알고리즘에서 P16×16 모드로 제한된 모드에 PDE 알고리즘을 적용했기 때문이다.

그림 4는 Forman과 Tempete에 대해 제안된 고

표 7. 고속 가변 블록 모드 결정 알고리즘의 성능

	QP	PSNR (dB)	bit rate (kbit/s)	time (sec)	ΔPSNR (dB)	Δbits (%)	Δtime (%)
Container	28	36.02	24.01	187.689	-0.03	-0.29	32.30
	32	33.25	13.47	183.715	-0.03	-0.07	32.79
	36	30.48	8.12	183.124	-0.04	1.12	32.33
	40	27.87	4.96	182.987	-0.02	0.81	31.78
News	28	36.82	45.14	192.472	-0.05	0.07	30.31
	32	33.81	27.95	195.315	0.00	-0.21	28.47
	36	30.96	17.14	193.310	0.02	0.35	28.38
	40	28.18	10.56	193.606	0.02	0.00	27.56
Foreman	28	36.01	74.92	234.971	-0.02	0.03	17.64
	32	33.42	44.58	223.200	-0.03	0.04	20.52
	36	30.98	27.24	217.002	-0.07	-0.40	22.14
	40	28.73	17.52	211.813	0.07	1.92	23.69
Paris	28	35.47	312.65	3052.064	-0.03	0.05	32.42
	32	32.39	181.13	3147.328	-0.04	-0.14	30.24
	36	29.52	101.65	3116.453	-0.05	0.01	30.88
	40	26.85	57.48	3172.657	-0.02	0.30	29.82
Silent voice	28	35.83	57.50	312.517	-0.06	0.38	25.72
	32	32.99	34.56	320.452	-0.03	0.00	22.91
	36	30.38	20.16	317.682	-0.03	0.00	22.88
	40	28.04	11.75	316.568	0.02	-0.42	22.65
Mobile	28	33.92	1488.66	8082.280	-0.03	0.14	12.99
	32	30.53	734.13	8041.954	-0.03	0.16	12.93
	36	27.51	354.43	7835.032	-0.02	-0.21	15.07
	40	24.61	190.65	7775.409	-0.01	-0.21	16.09
Tempete	28	34.86	1129.50	6632.295	-0.02	0.00	17.26
	32	31.66	546.54	6567.439	-0.03	0.14	17.70
	36	28.83	269.10	6466.562	-0.02	-0.42	19.01
	40	26.15	144.40	6507.530	0.00	0.31	18.93
average	28	35.56	447.48	2670.613	-0.03	0.05	24.09
	32	32.58	226.05	2668.486	-0.03	-0.01	23.65
	36	29.81	113.98	2618.452	-0.03	0.06	24.38
	40	27.20	62.47	2622.939	0.01	0.39	24.36

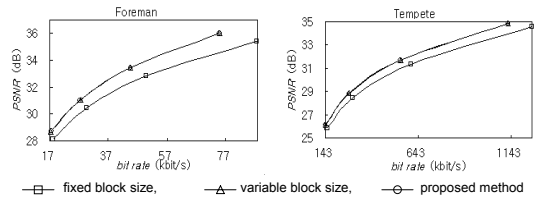


그림 4. 고속 가변 블록 모드 결정 알고리즘의 RD 곡선

속 블록 모드 결정 알고리즘과 H.264/AVC에서 고정 블록 크기와 가변 블록 크기 알고리즘을 선택했을 때의 올-왜곡 곡선 (rate-distortion curve)을 보여 준다. 제안된 알고리즘의 올-왜곡 곡선은 가변 블록 크기 알고리즘의 올-왜곡 곡선과 거의 구분할 수 없을 정도로 일치하고 있다.

VI. 결론

H.264/AVC에 새롭게 도입된 기술들은 압축 성능을 향상시키고 있으나 부호기의 복잡도를 가중시켜 실시간 구현을 어렵게 하고 있다. 본 논문은 H.264/AVC 부호기 복잡도의 주요 요인이 되는 가변 블록 모드에 대해 기술하고, 영상 복잡도와 블록

모드간 관계를 분석하여 IC와 MVcost로 표현되는 블록 영상의 복잡도를 이용한 새로운 고속 알고리즘을 제안하였다. 제안된 알고리즘은 영상 복잡도를 이용하여 결정될 가능성이 큰 블록 모드를 유효 블록 모드로 제한하고, 움직임 추정 및 RDcost 계산을 제한된 블록에서만 수행하고, 특히 P16×16 모드로 제한된 블록 모드에서는 움직임 추정을 위해 PDE를 적용시켜 계산량을 감소시키고 있다. 모의 실험 결과를 통해 제안된 알고리즘은 H.264/AVC의 가변 블록 크기 알고리즘과 비교하여 PSNR은 평균 0.02dB 감소하고 비트율은 평균 0.12% 증가하지만, 부호화 시간은 최저 12.93%에서 최대 32.79%, 평균 24.12% 감소시키고 있다. 이는 고정 블록 크기 알고리즘 정도의 계산량으로 가변 블록 크기 알고리즘 정도의 화질과 비트율은 구현할 수 있음을 의미한다.

참 고 문 헌

[1] ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services," 2005.

[2] J.Ostermann, J.Bormans, P List, D.Marpe, M.Narroschke, F.Pereira, T.Stockhammer, T.Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Transactions Circuits and Systems Magazine* Vol.4, Issue 1, pp.7-28, 2004.

[3] T. Wiegand, G. Sullivan, G.Bjontegaard, A.Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions CSVT*, Vol.13, pp.560-576, July 2003.

[4] X. Li, E. Q. Li, Y-K. Chen, "Fast multi-frame motion estimation algorithm with adaptive search strategies in H.264", in *Proceedings of the IEEE, ICASSP'04*, Vol.3, pp.369-372, 2004.

[5] 진순중, 김원기, 정제창, "H.264의 인트라 예측을 위한 적응적인 고속 알고리즘," *한국통신학회 논문지*, 30권, 12C호, pp.1153-1161, 2005.

[6] 김성희, 오정수, "H.264/AVC에서 다중 참조 프레임을 이용한 고속 움직임 추정," *한국통신학회 논문지*, 32권, 5호, pp.536-541, 2007.

[7] MPEG-4 – Information Technology – Coding of Audio-Visual Objects – Part 2: Visual, ISO/ IEC 14496-2, 2000.

[8] Video coding for low bitrate Communication, International Telecommunications Union, ITU-T Recommendation H.263, 1998.

[9] J.Kim, S.Byun, Y.Kim, and B.Ahn, "Fast full search motion estimation algorithm using early detection of impossible candidate vectors," *IEEE Trans. Signal Process.*, Vol.50, No.9, pp.2355 - 2365, Sep. 2002.

[10] ITU-T Recommendation H.263 software implementation, Digital Video Coding Group at Telenor R&D, 1995.

[11] G. Sullivan, G. Bjontegaard, "Recommended simulation common conditions for H.26L coding efficiency experiments on low-resolution Progressive-scan source material," ITU-T Q.6/16, Doc. #VCEG-N81, Sep. 2001.

[12] H.264 reference software version JM 9.5, <http://iphome.hhi.de/sueh-ring/tml/download/>.

김 성 희 (Seong-hee Kim)

정회원



2004년 8월 부경대학교 화상정보공학부 졸업
 2006년 8월 부경대학교 대학원 이미지시스템공학과 석사
 2006년 10월~2008년 1월 엠큐브웍스(주)
 2008년 1월~현재 로직코아(주)
 <관심분야> 디지털 영상 처리, 신호 처리

오 정 수 (Jeong-su Oh)

중신회원



1990년 2월 중앙대학교 전자공학과 졸업
 1992년 8월 중앙대학교 대학원 전자공학 석사
 1993년 7월~1997년 1월 신도리코 기술연구소
 2001년 8월 중앙대학교 첨단영상대학원 영상공학과 박사

2002년~현재 부경대학교 이미지시스템공학과
 <관심분야> 디지털 영상 처리, 적외선 신호처리