

H.264/AVC에서 다중 참조 픽처를 이용한 고속 움직임 추정

준회원 김 성 희**, 정회원 오 정 수*

Fast Motion Estimation Using Multiple Reference Pictures In H.264/Avc

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

요 약

동영상 압축 표준인 H.264/AVC에서 다중 참조 픽처를 이용한 움직임 추정은 압축 효율을 향상 시켰으나 그 효율은 참조 픽처의 수가 아닌 영상 내용에 의존적이다. 그래서 이 움직임 추정은 영상에 따라 많은 무의미한 계산을 포함하고 있다. 본 논문은 다중 참조 픽처를 이용한 움직임 추정의 무의미한 계산을 제거하는 고속 움직임 추정 알고리즘을 제안한다. 제안된 알고리즘은 영상 복잡도와 예측 움직임 벡터를 이용하여 다중 참조 픽처가 유효한 블록과 무효한 블록을 구분하고 무효한 블록에 단일 참조 픽처를 적용하여 무의미한 계산을 제거한다. 제안된 알고리즘의 성능 평가를 위해 참조 소프트웨어 JM 9.5에서 화질, 비트율, 움직임 추정 시간이 기존 알고리즘과 비교되었다. 실험 결과는 제안된 알고리즘이 평균 움직임 추정 시간을 약 38.67%로 크게 감소시키며 화질과 비트량을 각각 기존 알고리즘 정도인 -0.02dB와 -0.77% 정도로 유지시킬 수 있는 것을 보여주고 있다.

Key Words : Motion Estimation, Multiple Reference Pictures, Image Complexity, Predicted MV, H.264

ABSTRACT

In video coding standard H.264/AVC, motion estimation using multiple reference pictures improves compression efficiency but the efficiency depends upon image content not the number of reference pictures. So, the motion estimation includes a large amount of computation of no worth according to image. This paper proposes fast motion estimation algorithm that removes worthless computation in the motion estimation using multiple reference pictures. The proposed algorithm classifies a block into valid and invalid blocks for the multiple reference pictures and removes the worthless computation by applying a single reference picture to the invalid block. To estimate the proposed algorithm's performance, image quality, bit rate, and motion estimation time are compared with ones of the conventional algorithm in the reference software JM 9.5. The simulation results show that the proposed algorithm can considerably save about 38.67% the averaged motion estimation time while keeping the image quality and the bit rate, whose are average values are -0.02dB and -0.77% respectively, as good as the conventional algorithm.

1. 서 론

ISO/IEC의 MPEG(moving picture experts group)

과 ITU-T의 VCEG(video coding experts group)가 함께 구성한 JVT(joint video team)는 기존 동영상 압축 표준인보다 뛰어난 새로운 동영상 압축 표준인

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

논문번호 : KICS2007-01-030, 접수일자 : 2007년 1월 26일, 최종논문접수일자 : 2007년 4월 12일

을 제정하였다^[1]. 새로운 표준안은 ISO/IEC와 ITU-T에서 각각 사용되는 명칭인 MPEG-4 Part 10 AVC와 ITU-T Recommendation H.264를 근거로 H.264/AVC라 불린다. 다양한 새로운 기술들이 도입된 H.264/AVC는 그 성능이 매우 우수하여 새로 개발되는 미디어인 휴대이동방송과 HD-DVD 등에 채택되거나 채택될 예정이어서 관심의 대상이 되고 있다^[2]. 그러나 새롭게 적용된 기술들, 특히 가변 블록 크기의 움직임 추정과 다중 참조 픽처를 이용한 움직임 추정은 H.264/AVC 부호기의 복잡도를 크게 증가시켜 실시간 구현을 어렵게 하고 있다. 그래서 H.264/AVC 발표 이후에 상기의 움직임 추정과 관련된 많은 고속 알고리즘들이 연구되고 있다^[4,5,6,7].

본 논문에서는 편의상 다중 참조 픽처를 이용한 움직임 추정을 “다중 참조 픽처 움직임 추정”이라 하고, 이를 위한 효율적인 고속 알고리즘을 제안한다. H.264/AVC 부호기의 주요 복잡도 요인 중 하나인 다중 참조 픽처 움직임 추정에서 요구되는 계산량은 참조되는 픽처의 수에 비례하지만 그 효율은 픽처의 수에 비례하기 보다는 영상 내용에 의존적이다. 또한 동일한 움직임을 갖는 객체는 동일한 픽처에 존재할 가능성이 크고 예측움직임 벡터는 주변 블록의 움직임을 나타낸다. 이런 사실에 근거하여 본 논문은 영상 내용에 따라 다중 참조 픽처 움직임 추정의 효율이 떨어지는 블록과 예측 움직임 벡터를 이용해 직전 참조 픽처를 참조할 가능성이 큰 블록을 구분하여 다중 참조 픽처 대신 직전 픽처만을 사용하여 움직임을 추정하는 고속 알고리즘을 제안한다. 다섯 장의 참조 픽처를 사용한 모의 실험에서 제안된 알고리즘은 기존 알고리즘과 비교하여 PSNR (peak signal to noise ratio)에서 평균 0.025dB 정도의 미미한 감소가 있지만, 비트량에서 평균 0.77% 정도 감소하고, 움직임 추정 소요시간은 최대 54.06%, 평균 38.67%를 감소시키는 것을 확인할 수 있었다.

II. 다중 참조 픽처 움직임 추정

H.264/AVC에서는 최적의 예측 영상을 생성하기 위해 여러 참조 픽처들을 사용하고 있는데 이를 다중 참조 픽처 움직임 추정이라 한다. 본 장에서는 다중 참조 픽처 움직임 추정을 기술하고, 이 알고리즘의 성능을 평가하고 문제점을 제시한다.

2.1 다중 참조 픽처 움직임 추정

H.264/AVC 이전의 MPEG-4, H.263 등의 표준안은 현 픽처 (P-픽처)를 예측하기 위해 직전 픽처 참조하는 단일 참조 픽처 움직임 추정을 수행한다^[7,8]. 그러나 H.264/AVC에서는 최소 8×8 블록 단위로 여러 픽처를 참조하는 다중 참조 픽처 움직임 추정을 수행하여 더 향상된 움직임 추정이 가능해졌다^[1,2]. 이때 참조되는 픽처의 수는 프로파일과 레벨에 따라 정의되어 있다. 그림 1은 P-픽처 예측을 위한 다중 참조 움직임 추정 방식으로 현 픽처를 예측하기 위해 이전의 여러 픽처들을 참조하는 것을 보여주고 있다. 여기서 흰 픽처는 현 픽처이고, 회색 픽처들은 참조되는 이전 픽처들이다.

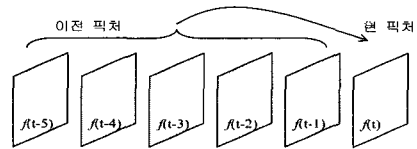


그림 1. P-픽처 예측을 위한 다중 참조 픽처 움직임 추정

2.2 다중 참조 픽처 움직임 추정의 성능 분석

H.264/AVC의 다중 참조 움직임 추정의 성능을 분석하기 위해 H.264/AVC의 참조 소프트웨어인 JM9.5를 이용하였고^[9], 실험 환경은 추후 모의 실험에서 기술하는 실험 환경과 동일하다.

표 1은 단일 참조 픽처와 5장의 이전 픽처를 사용한 다중 참조 픽처의 움직임 추정 및 보상에서 PSNR, 비트율, 움직임 추정 시간을 비교하고 있다. PSNR은 최소 0.07dB 최대 0.43dB, 평균 0.21dB 향상되고, 비트율은 최소 0.39% 최대 17.96%, 평균 6.45% 절약되고, 움직임 추정 시간은 최소 4.96배 최대 5.01배, 평균 4.98배 증가한다. 영상에 따른 결과는 Mobile과 Tempete 영상같이 움직이는 물체에 의해 가려지거나 나타나는 영역이 있는 영상이나 반복적인 움직임이 있는 영상에서 다중 참조 픽처 움직임 추정에 의해 PSNR과 비트율 성능이 크게 개선되었으나, 단순하고 정적인 영상 Container와 News에서는 성능 개선이 미미하다. 즉 다중 참조 픽처 움직임 추정의 성능이 영상 내용에 의존적인 것을 나타낸다. 반면에 다중 참조 픽처 움직임 추정 시간은 단일 참조 픽처 움직임 추정 시간의 약 5배로 영상에 따른 차이는 없고, 참조 픽처의 수에 비례하여 증가하는 것을 알 수 있다. 비트율에서 괄호안 값은 상대적인 비교를 위해 QCIF 기준으로 한 픽처당 요구되는 비트량이다.

표 1. 단일 참조 픽처와 다중 참조 픽처의 성능 비교

images	single picture			multiple picture		
	PSNR (dB)	bit rate (kbit/s)	ME time (sec)	PSNR (dB)	bit rate (kbit/s)	ME time (sec)
Container	33.21	13.97(1.40)	44.1	33.28	13.48(1.35)	221.1
News	33.70	28.12(2.81)	43.7	33.81	28.01(2.80)	217.5
Foreman	33.16	45.56(4.56)	45.8	33.45	44.56(4.46)	227.7
Paris	32.33	187.6(3.13)	837.2	32.43	181.39(3.02)	4148.9
Silent voice	32.91	35.92(2.40)	86.9	33.02	34.56(2.30)	335.5
Mobile	30.13	893.44(7.45)	1688.2	30.56	732.97(6.11)	8411.8
Tempete	31.33	634.69(5.29)	1473.3	31.69	545.75(4.55)	7335.4

(5 reference pictures, QP:32)

또한 다중 참조 픽처 움직임 추정을 수행했을 때 선택되어지는 참조 픽처의 분포를 각 영상에서 직전 픽처를 참조한 블록 (previous), 직전 이외의 픽처를 참조한 블록 (other), 인트라 블록 (intra)으로 구분하여 표 2에 보여 주고 있다. 평균적으로 72%의 블록이 직전 픽처를 참조하고, 26% 정도만이 직전 픽처 이외의 픽처를 참조하고 있다. Mobile과 Tempete 같은 영상에서는 직전 픽처 이외의 픽처를 많이 참조하여 다중 참조 픽처 움직임 추정의 효과를 얻고 있으나 Foreman 영상을 제외한 나머지 영상들에서는 85% 이상의 블록이 직전 픽처를 참조하고 있어 다중 참조 픽처 움직임 추정의 효과가 매우 낮다. 즉, 다중 참조 픽처 움직임 추정에서 직전 픽처를 참조하는 블록들이 압도적으로 많고, 이 들에서 움직임 추정 시간의 80%는 이득 없는 불필요한 계산임을 의미한다. 이는 효율성 측면에서 다중 참조 픽처 움직임 추정의 큰 문제이다.

표 2. 선택된 참조 픽처의 분포(%)

	previous	other	intra
Container	87.34	11.04	1.62
News	86.90	10.55	2.55
Foreman	63.68	29.49	6.83
Paris	87.36	11.27	1.37
Silent voice	85.74	11.21	3.05
Mobile	41.23	58.61	0.16
Tempete	50.05	47.88	2.07
Average	71.76	25.72	2.52

(5 reference pictures, QP:32)

III. 제안된 알고리즘

본 장에서는 영상 내용과 예측 움직임 벡터를 이용하여 블록에 따라 단일 참조 픽처 움직임 추정과 다중 참조 픽처 움직임 추정을 적응적으로 적용하는 고속 다중 참조 픽처 움직임 추정 알고리즘을

제안한다.

3.1 영상 복잡도를 이용한 고속 움직임 추정

사전 실험으로부터 다중 참조 픽처 움직임 추정은 움직임은 물체에 가려지거나 나타나는 영역과 반복적인 움직임이 있는 영역에서 효과적이었다. 그래서 본 논문에서는 블록 단위로 상기 영상 내용을 평가할 수 있는 척도를 이용해 다중 참조 픽처 움직임 추정이 유효한 블록과 무효한 블록을 구분하여 선택적으로 다중 참조 픽처 움직임 추정을 수행하는 고속 알고리즘을 제안한다.

영상 복잡도(image complexity:IC)는 블록 단위의 영상에서 밝기 변화의 정도를 나타내는 척도로 식 (1)에 의해 계산된다.

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

여기서 (i, j)와 N은 각각 화소의 위치와 블록의 크기이다. 큰 영상 복잡도는 블록 영상 내에 에지와 같은 큰 변화를 갖는 것을 의미하고, Taylor series expansion을 통해서 증명된 것처럼 정합 오차를 크게 발생시키고, 이로 인해 비트량도 많아지게 된다. 실제로 표 1과 표 2에서 비트량을 많이 요구하는 복잡한 영상들이 직전이 아닌 다른 픽처를 더 많이 선택하여 움직임 추정 효율을 높이고 있다. 그래서 영상 복잡도는 다중 참조 픽처 움직임 추정이 유효한 블록을 선택하는 충분한 척도가 될 수 있다.

제안된 알고리즘은 16×16 블록 단위로 수행되나 영상 복잡도의 평가는 다중 참조 픽처가 적용되는 최소 블록인 8×8 블록 단위로 수행된다. 그래서 16×16 블록은 8×8 크기의 부분 블록으로 나누어지고, 각 부분 블록에서 식 (1)를 이용해 영상 복잡도를 구하고, 부분 블록의 영상 복잡도들에서 최대값을 블록의 영상 복잡도로 선택한다. 그리고 영상 복잡도가 임계값 (th)보다 크면 다중 참조 픽처 움직임 추정을, 그렇지 않으면 단일 참조 픽처 움직임 추정을 수행한다. 제안된 알고리즘은 임계값 계산을 위해 초기 5 픽처 이후 적용되고, 영상 복잡도를 위해 부분 블록의 영상 복잡도의 합이 아닌 최대값을 선택한 것은 블록 전체에 작은 변화를 갖는 영상보다 부분적으로 큰 변화를 갖는 영상을 잘 예측하기 위한 것이다. 임계값은 식 (2)에 의해 계산된다.

$$th = ave_multi_ref_IC * w^2 \quad (2)$$

여기서 ave_multi_ref_IC는 다중 참조 픽처 움직임

추정이 수행되는 블록들의 평균 영상 복잡도로 초기 5 픽처들에서의 평균을 초기값으로 하여 픽처 단위로 갱신되는 값이고, w 는 영상과 QP에 대응하기 위한 값으로 초기 5 픽처들에서 직전 픽처를 참조하는 블록의 비율이다. 단순한 영상의 경우 큰 w 에 의해 임계값이 커져 다중 참조 픽처 움직임 추정을 수행하는 블록 수가 적어지고, 고화질을 위해 낮은 QP를 선택하면 w 가 작아지고 임계값이 작아져 다중 참조 픽처 움직임 추정을 수행하는 블록 수가 많아진다.

3.2 예측 움직임벡터를 이용한 고속 움직임 추정

일반적으로 객체들이 동일한 움직임을 가지거나 큰 객체가 여러 블록들로 나누질 때 주변 블록들은 동일한 움직임을 갖는다. 그래서 주변 블록들과 같은 움직임을 갖는 현재 블록은 주변 블록과 같은 픽처를 참조할 것이다. 또한 H.264/AVC에서 현 블록의 예측 움직임 벡터는 인접한 좌측, 상단, 우측 상단 블록들의 움직임 벡터들의 중간 (median) 값이며 영상 영역을 벗어난 무효 블록의 움직임 벡터는 유효 블록의 움직임 벡터나 (0,0)로 대체되고, 현 참조 픽처와 다른 참조 픽처의 움직임 벡터를 갖는 블록이 2개이면 그들은 제외된다^{1,2)}. 따라서 현재 블록의 움직임 벡터와 예측 움직임 벡터가 같다면 주변 블록들은 현재 블록과 같은 픽처를 참조하고 같은 움직임을 갖는 것을 의미한다. 만약 직전 참조 픽처에서 계산된 움직임 벡터와 예측 움직임 벡터가 같으면 현 블록은 직전 픽처를 참조할 가능성이 커진다. 이는 다중 참조 픽처 움직임 추정이 무효한 블록을 구분하는 척도가 될 수 있다.

제안된 알고리즘은 16×16 블록 단위로 직전 픽처만 참조하여 움직임을 추정하여 움직임 벡터를 얻고, 계산된 예측 움직임 벡터와 비교한다. 두 움직임 벡터가 같으면 블록은 주변 블록과 같은 움직임을 가지는 객체로 취급하여 직전 픽처에서 얻은 움직임 벡터를 최종 움직임 벡터로 결정하고, 그렇지 않으면 블록은 주변 블록과 다른 움직임을 갖는 객체로 판단하여 다른 픽처를 참조하여 움직임 추정을 계속 진행하여 최종 움직임 벡터를 구한다.

그림 2은 H.264/AVC에서 영상 복잡도와 예측 움직임 벡터를 함께 이용하여 제안된 고속 다중 참조 픽처 움직임 추정 알고리즘의 전체 흐름도이다. pMV 는 H.264/AVC에서 제공하는 예측 움직임 벡터이다.

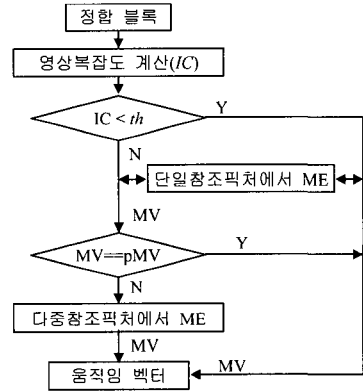


그림 1. 제안된 고속 다중 참조 픽처 움직임 추정의 흐름도

IV. 모의 실험

제안된 알고리즘의 성능 평가를 위해 H.264/AVC 표준안에서 권고하는 공통 실험 조건을 참조하여 표 3의 실험 환경¹⁰⁾에서 10Hz, 100 픽처의 QCIF 포맷 비디오인 Container, News, Forman 영상, 15Hz, 130 픽처의 QCIF 포맷 비디오인 Silent voice 영상, 15Hz, 150 픽처의 CIF 포맷 비디오인 Paris 영상, 30Hz, 300 픽처의 CIF 포맷 비디오인 Mobile, Tempete 영상들이 사용되어 H.264/AVC 참조 소프트웨어 JM 9.5가 수행되었다.

표 3. 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-웨어 최적화, 가변 크기 움직임 추정, CALC, 하다마드 변환무효화, B픽처 사용안함, Rs, FMO, ASO 사용안함

표 4는 제안된 알고리즘의 성능을 PSNR, 비트량, 움직임 추정 시간의 관점에서 비교된 결과를 보여주고 있다. 여기에서 Pro1, Pro2, Pro3은 각각 영상 복잡도, 예측 움직임 벡터, 그리고 두 가지 정보인 영상 복잡도와 예측 움직임 벡터를 동시에 이용한 고속 다중 참조 픽처 움직임 추정 알고리즘이고, $\Delta PSNR$, $\Delta Bits$, $\Delta Time$ 는 각각 다음 식들에 의해 정의된다.

$$\Delta PSNR = PSNR[pro] - PSNR[ref] \quad (3)$$

$$\Delta Bits = \frac{Bits[pro] - Bits[ref]}{Bits[ref]} \times 100[\%] \quad (4)$$

$$\Delta Time = \frac{Time[pro] - Time[ref]}{Time[ref]} \times 100[\%] \quad (5)$$

여기에서 *ref*는 표준안에 의한 실험 결과이고, *pro*는 제안된 알고리즘에 의한 실험 결과이다. 영상 복잡도 정보만을 이용한 고속 알고리즘에서는 평균 0.017dB 정도의 PSNR이 감소하고 평균 0.22% 정도의 비트량이 증가한다. 그리고 예측 움직임 벡터 정보만 이용한 고속 알고리즘에서는 평균 0.0025dB 정도 PSNR이 감소하고, 평균 0.38% 정도 비트량이 증가한다. 마지막으로 두 가지 정보를 함께 이용한 고속 알고리즘은 평균 0.025dB 정도 PSNR이 감소하고 비트량은 오히려 0.77% 정도 감소한다. 실험 결과로 알 수 있듯이 PSNR과 비트량은 모든 영상과 QP에서 미미한 변화만 있을 뿐이다. 그러나 계산량에 있어 영상 복잡도 정보만 이용한 고속 알고리즘은 최소 3.56%서 최대 32.41%, 평균 20.38% 정도 움직임 추정 시간을 감소시키고, 예측 움직임 벡터 정보만 이용한 고속 알고리즘은 최소 2.55% 최대 41.3%, 평균 34.97% 정도 움직임 추정 시간을 감소시키고, 두 가지 정보를 적용한 고속 알고리즘은 최소 9.35% 최대 54.06%, 평균 38.67% 정도 움직임 추정 시간을 감소시킨다. 결과적으로 PSNR과 비트량에서 아주 작은 손실로 계산량을 크게 감소시켜 주고 있다. 한편 다중 참조 픽처 움직임 추정에 크게 영향을 주는 영상 내용 측면에서 무의미한 계산이 많았던 단순한 영상 Container와 News에서는 많은 블록이 단일 참조 움직임 추정을 수행하여 계산량을 감소시키고 있고, 복잡한 영상 Mobile과 Tempete에서는 많은 블록이 다중 참조 픽처 움직임 추정을 수행하여 화질과 비트율을 개선시키고 있다. 따라서 제안된 알고리즘은 블록 영상의 내용에 따라 적절하게 움직임 추정을 수행한다고 판단할 수 있다.

그림 3은 QCIF와 CIF의 대표적인 영상 Foreman과 Mobile에 대해 표준안의 단일 참조 픽처 움직임 추정, 다중 참조 픽처 움직임 추정과 두 정보가 동시에 사용된 제안된 움직임 추정의 성능을 윌왜곡 곡선(rate-distortion curve)을 이용해 비교하고 있다. 제안된 알고리즘의 윌왜곡 곡선은 표준안의 다중

참조 픽처 움직임 추정의 윌왜곡 곡선과 거의 일치하고 있다.

표 4. 제안된 고속 다중 참조 픽처의 움직임 추정의 결과

Sequences	QP	$\Delta PSNR$ (dB)			$\Delta Bits$ (%)			$\Delta Time$ (%)		
		Pro1	Pro2	Pro3	Pro1	Pro2	Pro3	Pro1	Pro2	Pro3
Container	28	0.01	0.01	-0.01	0.25	0.17	-0.21	-29.64	-32.86	-50.25
	32	0.00	0.01	0.00	0.45	0.07	-1.15	-31.46	-34.32	-50.61
	36	-0.01	-0.01	0.00	-0.12	-0.12	-1.62	-31.05	-38.03	-51.99
	40	-0.03	0.01	-0.02	0.00	1.42	-2.64	-32.41	-37.93	-51.48
News	28	-0.04	-0.01	-0.05	-0.42	-0.02	-0.44	-25.11	-36.97	-51.40
	32	-0.02	-0.02	-0.03	-0.39	-0.46	-1.04	-24.56	-37.98	-51.32
	36	0.02	-0.01	0.03	-0.18	-0.18	-1.05	-24.82	-39.43	-52.15
	40	0.01	0.05	0.00	0.66	0.28	-1.70	-24.01	-41.30	-54.06
Foreman	28	-0.03	-0.03	-0.04	0.43	0.32	-0.68	-21.74	-19.08	-29.70
	32	-0.03	-0.03	-0.02	0.18	0.49	-0.99	-22.76	-24.99	-34.24
	36	-0.01	0.01	-0.03	0.51	0.80	-2.05	-21.83	-27.65	-39.44
	40	0.04	0.06	0.10	0.06	0.64	-0.23	-21.07	-29.08	-42.70
Paris	28	-0.01	-0.04	-0.04	-0.14	1.54	2.16	-19.00	-33.72	-45.51
	32	-0.02	-0.03	-0.06	-0.06	1.79	0.72	-19.06	-34.28	-47.33
	36	-0.03	-0.02	-0.01	-0.23	0.99	0.50	-19.07	-37.28	-47.89
	40	-0.03	0.00	-0.02	-0.12	0.42	-1.78	-18.90	-38.36	-50.12
Silent voice	28	-0.04	-0.01	-0.01	1.15	0.32	0.02	-26.42	-32.37	-42.91
	32	-0.02	-0.01	-0.02	1.22	0.24	-0.37	-26.98	-33.08	-43.40
	36	0.00	-0.03	-0.02	1.64	-0.20	-1.35	-27.08	-34.47	-44.48
	40	-0.01	0.00	-0.03	-0.08	0.31	-1.62	-26.50	-35.90	-46.81
Mobile	28	0.00	0.00	-0.04	-0.03	0.00	2.09	-3.56	-2.55	-9.35
	32	0.00	-0.01	-0.09	0.00	0.07	0.92	-3.69	-3.11	-10.21
	36	-0.01	-0.01	-0.12	-0.12	0.10	-1.68	-3.79	-5.32	-13.73
	40	0.01	-0.03	-0.08	0.08	-0.29	-3.88	-3.61	-13.45	-22.56
Tempete	28	-0.02	-0.01	-0.03	0.39	0.07	0.44	-15.78	-11.50	-17.81
	32	-0.02	0.00	-0.03	0.56	0.46	-0.10	-15.80	-13.41	-20.14
	36	-0.01	-0.02	-0.03	0.00	0.11	-1.98	-15.17	-18.50	-26.00
	40	0.02	0.00	0.02	0.43	1.26	-1.87	-15.77	-29.04	-35.20
Average in QP	28	-0.02	-0.01	-0.03	0.23	0.34	0.46	-20.18	-24.15	-35.28
	32	-0.02	-0.01	-0.04	0.28	0.38	-0.29	-20.62	-25.89	-36.75
	36	-0.01	-0.01	-0.03	0.22	0.22	-1.32	-20.40	-28.67	-39.38
	40	0.00	0.01	0.00	0.15	0.58	-1.96	-20.33	-32.15	-43.28
Average		-0.01	-0.01	-0.02	0.22	0.38	-0.77	-20.36	-27.71	-38.67

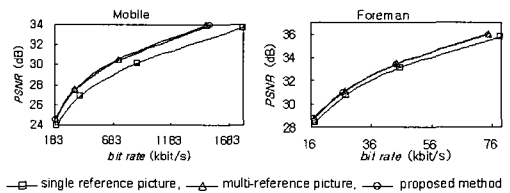


그림 3. 제안된 고속 다중 참조 픽처의 움직임 추정 알고리즘의 윌왜곡 곡선

V. 결론

H.264/AVC는 압축 성능이 가장 뛰어난 최신의 동영상 압축 표준안으로 최근 등장하는 다양한 분야의 뉴미디어에서 비디오 영상 압축기술로 채택되면서 더욱 부각되고 있다. 그러나 압축 성능을 향상시키기 위해 새롭게 적용된 기술들은 부호기의 복잡도를 가중시키고 있다. 본 논문은 H.264/AVC 부호기 복잡도의 주요 요인이 되는 다중 참조 픽처 움직임 추정에 대하여 기술하고 성능을 분석하고 계산량을 줄이기 위한 새로운 알고리즘을 제안하였다.

제안된 알고리즘은 영상 복잡도와 예측 움직임 벡터를 이용하여 블록 단위로 다중 참조 픽처 움직임 추정의 유효성을 판단하여 단일 참조 픽처 움직임 추정과 다중 참조 픽처 움직임 추정을 적응적으로 사용하는 고속 다중 참조 픽처 알고리즘을 구현하였다. 모의 실험 결과를 통해 제안된 알고리즘은 H.264/AVC의 다중 참조 픽처의 움직임 추정과 비교하여 화질은 평균적으로 -0.02dB 정도 떨어지지만 비트량은 평균 -0.77% 정도 줄이면서 움직임 추정 시간을 최대 54.06%, 평균 38.67% 감소시켜 주고 있다.

참 고 문 헌

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

[2] T. Wiegand, G. Sullivan, G. Bjontegaard, A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Transactions Circuits and Systems for Video Technology, vol. 13, pp. 560-576, July 2003.

[3] J. Y. Huang, B. Hsieh, T. Whang, S. Chien, S. Ma, C. Shen, L. Chen, "Analysis and reduction of reference frames for motion estimation in MPEG-4 AVC/JVT/H.264," in Proceedings of the IEEE, ICASSP'03, vol. 3, pp.145-148, April 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 인터/인트라 예측을 위한 적응적인 고속 알고리즘," 한국통신학회 논문지, 30권, 12C호, pp. 1153-1161, 2005.

[6] 이우호, 이정호, 조익환, 정동석, "다중 참조 영상의 적응적 선택 및 선택적 인트라 모드를 이용한 H.264/AVC의 고속 모드 결정 방법," 한국통신학회 논문지, 31권, 3C호, pp. 271-278, 2006.

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

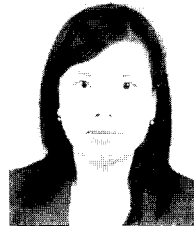
[8] Video coding for low bitrate Communication, International Telecommunications Union, ITU-T Recommendation H.263, 1998.

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

[10] 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.

김 성 희 (Seong-hee Kim)

준회원



2004년 8월 : 부경대학교 화상 정보공학부 졸업
 2006년 8월 : 부경대학교 대학원 이미지시스템공학과 석사
 2006~현재 : 엠큐브웍스(주)
 <관심분야> 디지털 영상 처리, 신호 처리

오 정 수 (Jeong-su Oh)

정회원



1990년 2월 : 중앙대학교 전자공학과 졸업
 1992년 8월 : 중앙대학교 대학원 전자공학 석사
 1993년 7월~1997년 1월:신도리 코기술연구소
 2001년 8월:중앙대학교 첨단영상대학원 영상공학과 박사
 2002년~현재 : 부경대학교 화상정보공학부
 <관심분야> 디지털 영상 처리, 적외선 신호처리