

H.264/AVC에서의 효율적인 고속 다중 참조 프레임 선택 기법

준회원 이 현 우*, 정회원 유 종 민*, 정 제 창*

Efficient Fast Multiple Reference Frame Selection Technique for H.264/AVC

Hyunwoo Lee* Associate Member, Jongmin Ryu*, Jechang Jeong* Regular Members

요 약

H.264/AVC 동영상 부호화 표준은 높은 부호화 효율을 위해서 가변 블록 크기 부호화, 1/4 화소 정확도의 움직임 예측, 다중 참조 프레임 사용, 율-왜곡 최적화 등과 같은 기법을 도입하였다. 그러나 이러한 부호화 방법들은 움직임 예측에 필요한 복잡도를 크게 증가시키는 단점이 있다. 특히 다중 참조 프레임 움직임 예측을 수행함으로써 증가되는 계산량은 탐색되는 참조 프레임의 개수에 따라 선형적으로 비례하여 늘어난다. 따라서 본 논문에서는 다중 참조 프레임 움직임 예측에서 탐색되는 참조 프레임의 개수를 효율적으로 조절함으로써 복잡도를 크게 줄일 수 있는 방법을 제안한다. 제안하는 알고리즘들은 인터 16×16 모드와 인접 블록의 최적 참조 프레임 정보를 이용하여 나머지 화면간 모드들에서의 불필요한 탐색 과정을 생략한다. 실험 결과는 제안하는 방법을 적용함으로써 영상의 화질과 비트율의 큰 변화 없이 부호화 수행 시간을 평균 58.54% 정도의 큰 폭으로 감소시킴을 보인다. 또한 이 방법은 다른 고속 움직임 예측 알고리즘들 같이 적용함으로써 추가적인 성능 향상을 얻을 수 있는 확장성을 가진다.

Key Words : Frame Selection, Motion Estimation, Fast Algorithm, Video Coding, H.264/AVC

ABSTRACT

In order to achieve high coding efficiency, H.264/AVC video coding standard adopts the techniques such as variable block size coding, motion estimation with quarter-pel precision, multiple reference frames, rate-distortion optimization, and etc. However, these coding methods have a defect to greatly increase the complexity for motion estimation. Particularly, from multiple reference frame motion estimation, the computational burden increases in proportion to the number of the searched reference frames. Therefore, we propose the method to reduce the complexity by controlling the number of the searched reference frames in motion estimation. Proposed algorithm uses the optimal reference frame information in both P16×16 mode and the adjacent blocks, thus omits unnecessary searching process in the rest of inter modes. Experimental results show the proposed method can save an average of 57.31% of the coding time with negligible quality and bit-rate difference. This method also can be adopted with any of the existing motion estimation algorithm. Therefore, additional performance improvement can be obtained.

※ 본 연구는 서울시 산학연협력사업으로 구축된 서울 미래형콘텐츠 컨버전스 클러스터 지원으로 수행되었습니다.

* 한양대학교 전자컴퓨터통신공학과 영상통신 및 신호처리 연구실 (girun@hanyang.ac.kr)

논문번호 : KICS2008-06-282, 접수일자 : 2008년 6월 23일, 최종논문접수일자 : 2008년 9월 4일

I. 서론

H.264/AVC는 ITU-T Video Coding Experts Group (VCEG)과 ISO/IEC Motion Picture Expert Group (MPEG)에 의해 공동으로 제정된 새로운 동영상 부호화 표준이다^[1]. MPEG-2, MPEG-4, H.263과 같은 이전의 표준들과 비교하여 높은 부호화 성능을 얻기 위해서 H.264/AVC는 다양한 새로운 부호화 기법들을 적용하였다. 그 예로써 가변 블록 크기 부호화, 1/4 화소 정확도의 움직임 예측, 다중 참조 프레임 사용, 율-왜곡 최적화 등의 방법들은 특히 화면간 부호화에 큰 이점을 지닌다. 이러한 부호화 방식들은 이전의 부호화 표준에 비해서 율-왜곡 성능 관점에서 뛰어난 향상을 보이지만, 더불어 부호화시 복잡도가 지나치게 증가하는 단점이 있다^{[1][2][3]}.

이러한 도구들 중에서 H.264/AVC는 기존의 표준들이 이전에 재구성된 바로 이전 프레임만을 참조 영상으로 사용한 것과는 달리 그림 1과 같이 여러 장의 프레임을 참조 영상으로 사용한다. 이는 움직임이 반복적으로 일어날 때나, 어떤 영역이 다른 객체에 의해 가려져 있다가 나타날 경우에 뛰어난 성능을 보이며, 압축 효율을 극대화할 수 있다^{[2][3]}.

또한 기존의 동영상 부호화 표준들이 16×16 또는 8×8 크기만으로 움직임 예측을 하는 것과는 달리, H.264/AVC에서는 그림 2와 같이 블록의 크기를 더욱

다양하게 하여 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4 단위 크기로 움직임 예측을 수행한다. 특히 8×8 블록크기 이하의 서브매크로블록(Sub-Macroblock)으로 수행되는 움직임 예측은 복잡한 영상에서 보다 높은 부호화 이득을 취할 수 있게 한다^{[2][3]}.

그러나 H.264/AVC의 이러한 7가지 가변 블록 크기의 움직임 예측 방식은 이전의 표준에 비해 복잡도가 크게 늘어나는 문제점을 야기한다. 또한 움직임 예측을 위해 여러 장의 참조 프레임을 사용할 경우 계산량이 더욱 늘어날 수 있다. 그러나 이러한 계산량의 증가에도 불구하고 예측 오차의 감소는 오히려 영상의 특성과 관계가 더 깊다. 그러므로 어떤 경우에는 더 많은 참조 프레임을 사용하여 얻을 수 있는 예측 이득이 클지 모르지만, 그렇지 않은 경우에는 만족할 만한 이득을 얻지 못한 채 계산량만 낭비되는 문제가 존재한다. 이러한 문제를 해결하기 위한 방법들 중 움직임 예측시 탐색될 참조 프레임의 개수를 효율적으로 감소시킴으로써 복잡도를 최소화하는 방법이 있고, 현재 효과적인 방법들이 다수 제안되고 있다^{[4][5][6][7][8][9][10][11]}.

특히 Huang은 이전 프레임의 움직임 예측 결과에 기반하여 이전 프레임 또는 모든 참조 프레임을 탐색하는 방법을 제안하였다^[8]. 또한 Hsu는 이웃 블록의 참조 프레임만을 현재 블록의 움직임 추정을 위해 탐색하는 방법을 제안하였다. 그러나 현재 블록과 이웃 블록사이에 경계 성분이 존재한다면 부호화 성능이 하락할 수 있는 단점이 존재한다^[10].

본 논문에서는 인터 16×16 모드에서 움직임 예측을 수행한 결과 얻어진 최적의 참조 프레임 정보를 이용하여 나머지 모드에서 사용하는 참조 프레임을 제한하고, 또한 인접 블록들의 참조 프레임 정보를 이용하여 추가적으로 참조 프레임 개수를 줄이는 알고리즘을 제안한다. 이 방법은 영상의 화질과 비트율을 기존의 전체 참조 프레임을 모두 사용하는 방법에 가깝게 유지하면서 부호화 수행 시간을 대폭 감소시킨다.

본 논문은 다음과 같이 구성되어 있다. 먼저 II장에서 H.264/AVC의 부호화 과정에서 결정되는 최적 참조 프레임들의 통계적 특성에 대해서 살펴 보고, 이러한 통계적 특성에 기반한 제안하는 고속 다중 참조 프레임 선택 방법에 대해 III장에서 설명한다. IV장에서는 제안하는 알고리즘과 기존의 H.264/AVC 표준과의 비교를 통해서 제안하는 알고리즘의 타당성을 보이고, 마지막으로 V장에서 결론을 맺는다.

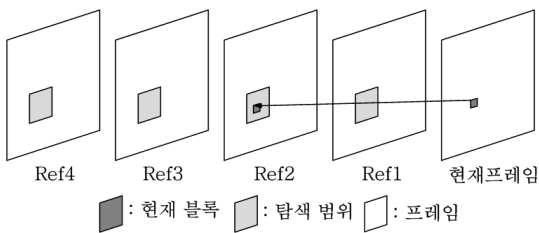


그림 1. H.264/AVC 에서의 다중 참조 프레임을 사용한 움직임 벡터 예측

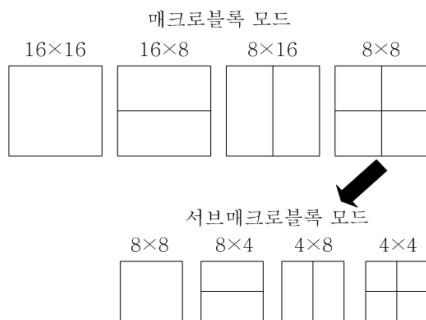


그림 2. H.264/AVC의 화면간 예측을 위한 부호화 모드

II. 최적 참조 프레임의 통계적 특성

2.1 인터 16×16 모드와 최종적으로 결정된 최적 참조 프레임간의 관계

H.264/AVC 표준의 움직임 예측 과정은 16×16 크기의 매크로블록을 16×8, 8×16, 8×8 의 형태로 분리한 각각의 블록에 걸쳐 모두 수행하여 최적 모드와 최적의 참조 프레임을 결정하게 된다. 또한 움직임 예측은 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4 의 크기가 큰 블록에서부터 작은 블록으로의 순서로 수행된다. 움직임 예측 과정이 각 블록별로 수행되기는 하지만 분리된 블록이라 하더라도 그 블록은 결국 16×16 크기의 매크로블록의 일부분이다.

로 인터 16×16 모드와 각각의 하위 모드들은 동일한 최적 참조 프레임이 선택될 확률이 비교적 높다는 것을 생각해 볼 수 있다. 따라서 최종적으로 결정되는 최적 참조 프레임 또한 인터 16×16 모드에서의 최적 참조 프레임과 밀접한 연관성을 보일 것이다.

최적 참조 프레임의 특성을 확인하기 위해서 두 개의 QCIF 해상도의 테스트 영상(Coastguard, Stefan)을 이용하여 H.264/AVC 참조 소프트웨어(JM 11.0) [12]로 시뮬레이션을 수행하였다. 참조 프레임의 개수는 5개, 양자화 파라미터는 24, 32, 36을 사용하고, 탐색 범위는 16, 움직임 예측은 전역 탐색(Full Search) 방법을 사용하였으며, 총 100프레임을 부호

표 1. Coastguard 영상의 인터 16×16 모드에 대한 최적 참조 프레임 연관성

QP	Inter16×16	최종적으로 결정된 최적 참조 프레임				
		Ref0	Ref1	Ref2	Ref3	Ref4
24	Ref0	30195	860	390	289	194
	Ref1	318	1387	44	40	31
	Ref2	132	37	560	21	22
	Ref3	96	49	33	772	46
	Ref4	47	20	18	11	424
32	Ref0	21406	740	437	136	141
	Ref1	359	1321	72	20	16
	Ref2	137	39	825	9	10
	Ref3	42	10	18	518	8
	Ref4	35	15	12	2	280
36	Ref0	15126	398	247	88	89
	Ref1	228	1316	27	10	11
	Ref2	112	22	766	9	7
	Ref3	54	9	12	256	5
	Ref4	62	10	2	2	228

표 2. Stefan 영상의 인터 16×16 모드에 대한 최적 참조 프레임 연관성

QP	Inter16×16	최종적으로 결정된 최적 참조 프레임				
		Ref0	Ref1	Ref2	Ref3	Ref4
24	Ref0	18829	1728	1288	729	586
	Ref1	1047	2293	293	206	153
	Ref2	611	204	1526	173	130
	Ref3	345	87	135	1085	80
	Ref4	378	94	97	94	789
32	Ref0	16567	1276	793	427	333
	Ref1	850	2293	259	107	87
	Ref2	522	178	1688	70	62
	Ref3	244	81	72	886	73
	Ref4	291	79	68	53	733
36	Ref0	16060	786	435	156	207
	Ref1	579	2236	117	37	39
	Ref2	423	86	1417	32	30
	Ref3	152	38	19	564	35
	Ref4	215	23	33	24	645

표 3. 인터 16×16 모드와 최종적으로 결정된 최적 모드간의 최적 참조 프레임 일치율

테스트 영상	QP	최적 모드			평균
		16×8	8×16	8×8	
Coastguard	24	83.44%	85.62%	90.41%	82.83%
	32	77.56%	81.63%	85.75%	
	36	78.82%	79.27%	82.95%	
Stefan	24	56.70%	58.63%	64.01%	65.56%
	32	58.81%	62.10%	72.01%	
	36	67.05%	69.27%	81.51%	

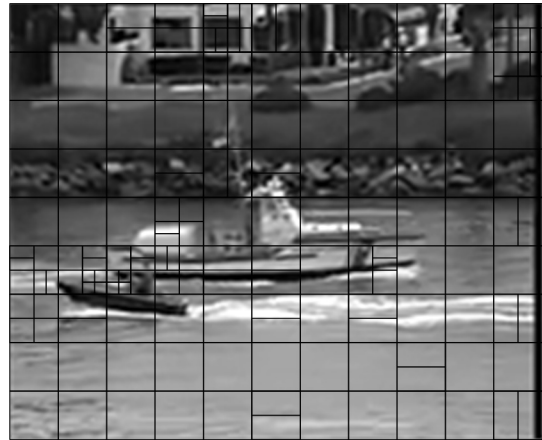
화하였다. 그 결과는 표 1부터 표 3까지 각 테스트 영상별로 나타내었다.

표 1, 2를 살펴 보면 인터 16×16 모드에서 결정된 최적의 참조 프레임이 최종적으로 결정되는 최적 참조 프레임과 매우 밀접한 관계를 가짐을 알 수 있고, 또한 이는 실험한 두 개의 테스트 영상에 대해서 일관된 결과를 보인다. 이러한 현상은 바꾸어 말하면 움직임 예측 과정에서 사용되는 많은 참조 프레임의 대부분이 최종적으로 최적의 참조 프레임으로 결정되지도 않으면서 쓸모 없이 탐색되어 계산량의 낭비를 일으킨다고 할 수 있다. 또한 표 3에서는 최적 모드가 16×8, 8×16, 8×8과 같은 하위 모드로 결정될 때, 인터 16×16 모드에서 결정된 최적 참조 프레임이 하위 모드의 최적 참조 프레임과 비교적 정확하게 일치함을 확인할 수 있다.

따라서 이와 같은 통계적 정보들에 근거하여 인터 16×16 모드의 최적의 참조 프레임 정보를 사용해서 불필요하게 탐색되는 참조 프레임의 개수를 적응적으로 감소시키는 방법을 생각해 볼 수 있다.

2. 인접 블록간의 최적 참조 프레임 관계

영상 신호는 그 특성상 이웃하는 블록들간의 연관성이 매우 밀접하다. 따라서 특정 블록을 부호화하는데 있어서 이웃 블록의 정보를 이용한다면 효과적일 수 있다. 그림 3은 Coastguard 테스트 영상(QCIF)의 88번째 프레임의 움직임 예측 결과 결정된 최적 모드와 최적 참조 프레임을 나타낸 것이다. 양자화 파라미터는 32, 탐색 범위는 16, 참조 프레임 개수는 5로 부호화를 설정하였다. 그림에서 확인할 수 있듯이, 움직임 예측에서 결정된 최적 참조 프레임은 영상 전체를 살펴 보았을 때, 동일한 참조 프레임이 공간적으로 집중되어 나타나는 경향을 보인다. 이와 같은 경향으로부터 인접한 블록들끼리 참조 프레임에 있어서 서로 밀접하게 연관되어 있음을 알 수 있다. 이러한 인접 블록간의 관계를 바탕으로, 부호화하고자 하는 블록의 이웃 블록들의 참조 프레임 정



(a)

0	3	0	0	0	1	1	1	2	3	3	4	0	0
		3	3	0	1	1	1	1	4	0		0	0
3	3	3	3	0	1	1	1	1	4	0		0	0
												0	0
3	4	3	0	0	0	1	1	1	SKIP	SKIP			0
3	3	0		0	0	0		SKIP	SKIP	SKIP	SKIP		0
3	0	0	0	0	0	1	0	0	0	0	0	0	0
2	0	0	0	0	4	4	0	0	0	SKIP	3	0	0
0	0	4	0	0	0	0	0	0	0	0	0	0	0
2	0	0		0	0	0	0	0	0	0	SKIP	0	0
0	0		0	0	0	0	0	0	0	0	SKIP	0	0
0	0	SKIP	1	1	SKIP	SKIP	SKIP	SKIP		2	SKIP		0
0	0	0	SKIP	SKIP	4	0	SKIP	SKIP	SKIP	0	SKIP	0	0

(b)

그림 3. Coastguard 영상의 (a) 최적 분할 모드, (b) H.264/AVC의 부호화 과정에서 결정된 최적 참조프레임

보를 이용하여 현재 블록의 참조 프레임 개수를 효과적으로 줄이는 방법을 고려해 볼 수 있다.

III. 제안하는 알고리즘

이전 장의 최적 참조 프레임의 통계적 특성의 분석에 따라 본 논문에서는 탐색하는 참조 프레임의

개수를 효율적으로 감소시키는 고속 다중 참조 프레임 선택 알고리즘을 제안한다. 제안하는 알고리즘은 먼저 인터 16×16 모드에서 결정된 최적의 참조 프레임을 효과적으로 이용하여 움직임 예측 과정의 복잡도를 크게 줄인다. 먼저 기존의 움직임 예측 방식과 동일하게 이용 가능한 모든 참조 프레임을 탐색하여 인터 16×16 모드의 최적 참조 프레임을 결정한다. 그러면 현재 프레임의 바로 이전 프레임으로부터 앞에서 결정된 인터 16×16 모드의 최적 참조 프레임까지가 하위 모드(인터 16×8, 인터 8×16, 인터 8×8, 인터 8×4, 인터 4×8, 인터 4×4)에 대한 후보 참조 프레임(Candidate Reference Frame, CRF)이 된다. 예를 들어, 만약 인터 16×16 모드에서 바로 이전 프레임(Ref0)이 최적의 참조 프레임으로 선택되었다면 하위 모드에서는 단지 한 개의 참조 프레임(Ref0)만이 후보 참조 프레임이 된다. 만약 현재 프레임에서 두 장 전의 프레임이 최적 참조 프레임으로 결정되었다면 하위 모드는 두 개의 참조 프레임(Ref0, Ref1)을 후보 참조 프레임으로 가진다.

다음으로 이렇게 선택된 참조 프레임의 개수를 현재 매크로블록의 인접 매크로블록의 참조 프레임 정보를 이용하여 좀 더 줄인다. 그림 4의 빗금 친 영역은 현재 매크로블록에서 이용 가능한 이전에 부호화된 인접 매크로블록을 나타낸다. 각 인접 매크로블

0	1	4	5	8	9
2	3	6	7	10	11
12	13	current MB			
14	15				

그림 4. 제안하는 알고리즘에서 사용되는 인접 매크로블록

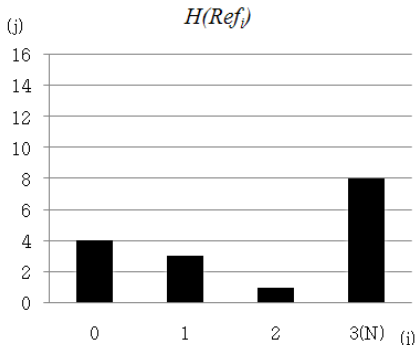


그림 5. 매크로블록의 참조프레임 분포에 따른 히스토그램

록들은 모드들간의 서로 다른 크기를 고려하기 위하여 최적 참조 프레임 결정의 최소 단위인 8×8 크기로 분할되어 16개의 블록으로 표현된다.

이들 16개 인접 블록들의 참조 프레임 정보를 식 (1)에 적용하여 그림 5와 같은 히스토그램이 생성된다. 그림 5는 현재 매크로블록의 인터 16×16 모드에서 결정된 최적 참조 프레임이 Ref3인 경우의 히스토그램이다.

$$H(Ref_i) = \sum_{j=0}^{15} id_{xref_{ij}} \tag{1}$$

$$id_{xref_{ij}} = \begin{cases} 1, & \text{if block } j\text{'s reference frame} = i \\ 0, & \text{otherwise} \end{cases}$$

여기서 i 는 앞에서 결정된 후보 참조 프레임들의 인덱스(0...N), j 는 그림 4에서의 인접 매크로블록이 8×8 크기로 분할되었을 때의 인덱스(0...15), N 은 인터 16×16 모드에서 결정된 참조 프레임 인덱스를 각각 나타낸다.

이렇게 만들어진 히스토그램을 이용하여 현재 매크로블록의 인터 16×16 모드를 제외한 나머지 모드에서 사용되는 후보 참조 프레임의 개수를 추가로 줄일 수 있다. 먼저 히스토그램의 재정렬이 필요하다. 현재 매크로블록의 인터 16×16 모드에서 결정된 최적 참조 프레임이 인접 매크로블록에서도 나타날 가능성이 가장 높기 때문에 그림 5를 예로 들면 그에 대응하는 히스토그램 항목인 $H(Ref_3)$ 을 가장 왼쪽으로 이동시킨다. 따라서 그림 5와 같은 히스토그램은 그림 6과 같이 재정렬된다.

그림에서 보여지듯이 히스토그램의 재정렬을 통해 왼쪽에서 오른쪽으로 향하는 순으로 각 항목의 도수가 대체적으로 점점 작아지는 순서로 정렬이 된다. 먼저 인터 16×16 모드에서 선택된 최적 참조 프레임인 Ref_3 은 후보 참조 프레임 집합 내에 그대로 포

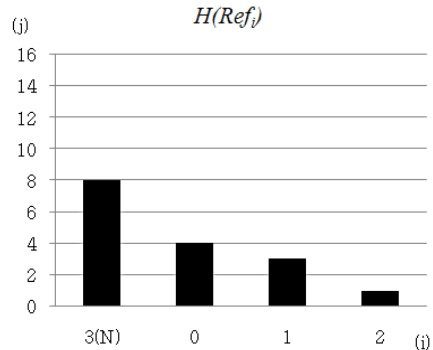


그림 6. 재정렬된 히스토그램

```

1: Conduct motion estimation(ME) for Inter16×16
2: Determine the optimal reference frame(N) for Inter16×16
3: Calculate  $H(Ref_i)$  using expression (1)
4: Set  $Ref_N$  as Candidate Reference Frame(CRF)
5: If  $\lambda \cdot H(Ref_N)$  is less than  $H(Ref_0)$ 
6:     Set  $Ref_0$  as CRF
7: Else If
8:     Goto 17
9: End If
10: Loop  $i=0$  to  $N-2$ 
11:     If  $\lambda \cdot H(Ref_i)$  is less than  $H(Ref_{i+1})$ 
12:         Set  $Ref_{i+1}$  as CRF
13:     Else If
14:         Break Loop
15:     End If
16: End Loop
17: Conduct ME for the mode excluding Inter16×16 using the CRFs only
18: Determine both the mode and the optimal reference frame for the current macroblock
    
```

그림 7. 제안하는 알고리즘의 의사 코드

함된다. 다음으로 그에 대응하는 히스토그램 도수인 $H(Ref_3)$ 에 1보다 작은 특정 상수(λ)를 곱한 값을 바로 오른쪽 항목의 도수인 $H(Ref_0)$ 와 비교한다. 만약 전자가 값이 더 작다면 오른쪽 항목에 대응하는 Ref_0 에 해당하는 참조 프레임은 후보 참조 프레임 집합 내에 남아 있게 된다. 그리고 나서 이와 동일한 과정을 $H(Ref_0)$ 와 왼쪽에서 세 번째 항목인 $H(Ref_i)$ 을 사용하여 수행한다. 이 과정은 히스토그램의 가장 오른쪽까지 반복된다. 만약 반대로 후자의 값이 더 작다면, 이 반복 비교 과정은 그 순간 종료되고 나머지 오른쪽 항목에 대한 참조 프레임은 후보 참조 프레임으로부터 제거된다. 최종적으로 결정된 이 후보 참조 프레임들만이 현재 매크로블록의 인터 16×16 모드를 제외한 나머지 모드에서 탐색된다. λ 는 실험을 통하여 0.35로 결정하였고, 그림 7에서 제안하는 방법의 전체적인 수행 과정을 나타내었다.

IV. 실험 결과

제안하는 알고리즘은 JM 11.0의 실험결과와 비교되었다. 메인 프로파일을 사용하여 QCIF(176×144) 해상도의 5개의 테스트 영상(News, Foreman, Table Tennis, Coastguard, Stefan)을 100 프레임까지 부호화하였다. 주요 실험 조건은 다음과 같다. 프레임율은 30으로 설정하였고, 각 영상은 첫번째 프레임만 I

픽처로 부호화하고 나머지 프레임은 P 픽처로 부호화하였으며, 움직임 추정 방식으로는 전역 탐색 방법을 사용하였다. 또한 탐색 범위는 16으로 설정하였고, 화면간 모드를 모두 사용하였다. 양자화 파라미터는 24, 28, 32, 36 의 네 가지 값을 사용하였고, 윌-왜곡 최적화를 적용하였다. 실험은 Pentium IV 2.8GHz, 1GB 메모리의 시스템에서 수행되었다.

표 4는 참조 프레임의 개수를 5장으로 설정한 JM 11.0(JM_Ref5)과 제안하는 알고리즘의 성능을 객관적 화질, 비트율, 움직임 예측 시간 측면에서 각각 비교한 결과이다. 표에서 확인할 수 있듯이, 먼저 PSNR 측면에서 살펴 보았을 때 H.264/AVC 표준과 제안하는 방법은 무시할 만한 수준의 화질 차이를 가짐을 알 수 있다. 제안하는 방법은 JM 11.0과 비교하여 평균적으로 0.042dB 가량의 화질 저하를 보인다. PSNR 결과와 마찬가지로 제안하는 방법의 비트율 차이 또한 거의 무시할 만하다. 제안하는 방법은 JM 11.0과 비교하여 평균적으로 1.10%의 비트율 증가를 보인다. 또한 표의 결과에서 나타나듯이 제안하는 방법을 사용함으로써 JM 11.0과 비교하여 상당한 움직임 예측 시간을 절약할 수 있음을 확인할 수 있다. 평균 58.54% 가량의 움직임 예측 시간의 이득을 얻을 수 있고, 최대 64% 정도까지 움직임 예측 시간의 절약이 가능하다.

표 5는 제안하는 알고리즘을 참조 프레임 개수를

표 4. 제안하는 방법과 참조 프레임 개수를 5개로 설정한 JM 11.0과의 성능 비교 결과

테스트영상	QP	PSNR			비트율			움직임 예측 시간		
		JM_Ref5	PRO	△PSNR (dB)	JM_Ref5	PRO	△bitrate(%)	JM_Ref5	PRO	Speedup (%)
News	24	39.722	39.687	-0.035	124114	125203	0.88	83987	32050	61.84
	28	36.708	36.686	-0.022	75322	76013	0.92	80166	30658	61.76
	32	33.742	33.744	+0.002	45334	45286	-0.11	76513	30505	60.13
	36	30.900	30.887	-0.013	27461	27660	0.72	73922	30415	58.86
Foreman	24	39.075	38.974	-0.101	219979	223378	1.55	125976	56978	54.77
	28	36.440	36.323	-0.117	132430	134148	1.30	116774	52390	55.14
	32	33.752	33.627	-0.125	81682	81552	-0.16	108263	46356	57.18
	36	31.064	31.305	+0.241	50563	51413	1.68	100778	42938	57.39
Table Tennis	24	38.267	38.211	-0.056	332251	337130	1.47	176530	73167	58.55
	28	35.561	35.521	-0.040	189566	192838	1.73	165098	69588	57.85
	32	33.120	33.069	-0.051	105132	106846	1.63	148020	61741	58.29
	36	31.045	30.985	-0.060	61049	61189	0.23	128130	53627	58.15
Coastguard	24	37.175	37.147	-0.028	448480	471122	0.56	179813	65086	63.80
	28	34.022	33.990	-0.032	244685	247183	1.02	168549	61586	63.46
	32	30.994	30.986	-0.008	108173	109258	1.00	157560	60075	61.87
	36	28.489	28.481	-0.008	48283	48523	0.50	141613	57871	59.13
Stefan	24	37.786	37.688	-0.098	613483	624413	1.78	153335	68498	55.33
	28	34.343	34.275	-0.068	355322	365232	2.79	147171	66590	54.75
	32	30.872	30.762	-0.110	189876	193238	1.77	141750	62953	55.59
	36	27.758	27.643	-0.115	104422	105187	0.73	136709	58785	57.00

표 5. 제안하는 방법과 참조 프레임 개수를 3개로 설정한 JM 11.0과의 성능 비교 결과

테스트영상	QP	PSNR			비트율			움직임 예측 시간		
		JM_Ref3	PRO	△PSNR (dB)	JM_Ref3	PRO	△bitrate(%)	JM_Ref3	PRO	Speedup (%)
News	24	39.694	39.687	-0.007	124553	125203	0.52	48772	32050	34.29
	28	36.701	36.686	-0.015	75468	76013	0.72	45654	30658	32.85
	32	33.740	33.744	+0.004	45192	45286	0.21	45173	30505	32.47
	36	30.867	30.887	+0.020	27446	27660	0.78	43833	30415	30.61
Foreman	24	39.020	38.974	-0.046	223486	223378	-0.05	73689	56978	22.68
	28	36.369	36.323	-0.046	133817	134148	0.25	65317	52390	19.79
	32	33.618	33.627	+0.009	82006	81552	-0.55	62398	46356	25.71
	36	31.095	31.305	+0.210	51115	51413	0.58	57435	42938	25.24
Table Tennis	24	38.234	38.211	-0.023	332988	337130	1.24	98596	73167	25.79
	28	35.539	35.521	-0.018	190296	192838	1.34	92774	69588	24.99
	32	33.098	33.069	-0.029	105924	106846	0.87	83381	61741	25.95
	36	30.992	30.985	-0.007	60626	61189	0.93	73891	53627	27.42
Coastguard	24	37.164	37.147	-0.017	473966	471122	-0.60	98966	65086	34.23
	28	33.994	33.990	-0.004	247090	247183	0.04	93591	61586	34.20
	32	30.982	30.986	+0.004	108902	109258	0.33	87494	60075	31.34
	36	28.445	28.481	+0.036	47914	48523	1.27	79592	57871	27.29
Stefan	24	37.736	37.688	-0.048	622572	624413	0.30	85563	68498	19.94
	28	34.290	34.275	-0.015	361560	365232	1.02	81965	66590	18.76
	32	30.770	30.762	-0.008	191489	193238	0.91	78897	62953	20.21
	36	27.662	27.643	-0.019	104890	105187	0.28	76295	58785	22.95

3장으로 설정한 JM 11.0(JM_Ref3)과의 성능 비교 결과를 나타낸다. 제안하는 방법을 사용하지 않은 JM 11.0에서도 시간적으로 먼 위치에 있는 참조 프레임들은 거의 선택되지 않기 때문에 참조 프레임의 개수를 줄인 참조 소프트웨어와의 비교 결과가 요구된다. 참조 프레임의 개수를 5장으로 설정한 표 4의 결과에서 대략 60% 정도의 움직임 추정 시간이 감소하기 때문에 참조 프레임의 개수를 3장으로 설정

한 JM 11.0과의 성능 비교를 수행하였다. 표 5에서 확인할 수 있듯이 참조 프레임 개수를 3장으로 설정한 JM 11.0과 비교해서도 제안하는 방법이 수용할 만한 수준의 성능을 나타내는 것을 알 수 있다. 평균적으로 0.001dB의 PSNR 저하와 0.52%의 비트율 증가를 보이면서 26.84%의 움직임 예측 시간의 속도가 증가하였다.

그림 8은 표 4와 5의 각 방법에 대한 율-왜곡 곡

V. 결 론

본 논문에서는 먼저 인터 16×16모드의 최적 참조 프레임과 현재 블록의 최종 참조 프레임간의 관계와 서로 이웃하는 블록간의 참조프레임에 있어서의 유사성을 분석하였고, 그에 기반한 고속 다중 참조 프레임 선택 방법에 대해 제안하였다. 제안하는 방법은 인터 16×16 모드에서 선택되는 최적의 참조 프레임과 현재 매크로블록의 인접 매크로블록의 참조 프레임 정보에 기반하여 하위 화면간 모드에서 탐색되는 참조 프레임의 개수를 효과적으로 줄여 움직임 예측 과정의 복잡도를 크게 감소시킨다. 실험 결과는 제안하는 알고리즘이 H.264/AVC 표준과 비교하여 거의 유사한 PSNR과 비트율을 가지면서 부호화 시간을 평균 58.54% 정도 감소시키는 것을 보여 준다. 따라서 제안하는 방법은 특히 낮은 복잡도를 요구하는 실시간 시스템에 널리 적용 가능하다.

참 고 문 헌

- [1] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Draft ITU-T recommendation and final draft international of joint video specification (ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC)," JVT-G050, Geneva, Switzerland, 23-27 May 2003.
- [2] I. Richardson, "H.264 and MPEG-4 Video Compression," Wiley, 2003.
- [3] T. Wiegand, G. J. Sullivan, G. Bjøntegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. Circuits and Systems for Video Technology*, Vol.13, pp.560-576, July 2003.
- [4] G. L. Li and M. J. Chen, "Adaptive Search Range Decision and Early Termination for Multiple Reference Frame Motion Estimation for H.264," *IEICE Trans. Communications*, Vol. E89-B, No.1, pp.250-253, Jan. 2006.
- [5] T. Y. Kuo and H. J. Lu, "Efficient Reference Frame Selector for H.264," *IEEE Trans. Circuits and Systems for Video Technology*, Vol.18, No.3, pp.400-405, Mar. 2008.
- [6] Y. Shen, C. Huang, L. Yu, D. Zhang and J. Li, "Fast Multiframe Motion Estimation Algorithm in H.264," *Proc. Int. Conf. Signal Processing*,

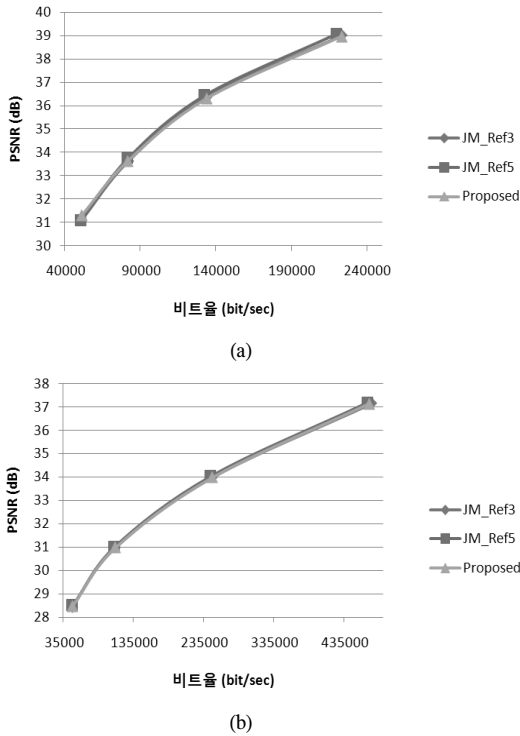


그림 8. 제안하는 방법과 참조 소프트웨어간의 율-왜곡 곡선 비교 결과. (a) Foreman. (b) Coastguard

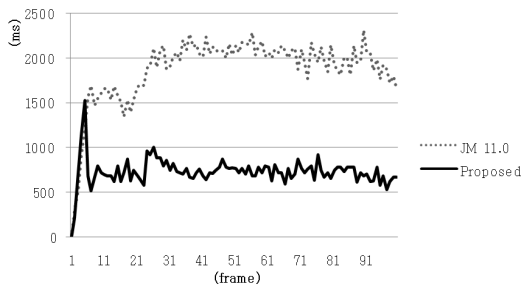


그림 9. 프레임별 움직임 추정 시간 비교 (Table Tennis 영상, QP=24)

선 비교 결과를 도시한 것이다. 그림에서 보여지듯이 제안하는 방법은 참조 소프트웨어와 비교해서 율-왜곡 성능 측면에서 거의 차이가 존재하지 않음을 확인할 수 있다.

그림 9는 Table Tennis 테스트 영상에서의 참조 소프트웨어와 제안하는 방법의 움직임 예측 시간을 첫번째 프레임부터 100번째 프레임까지 프레임별로 비교한 결과를 도시한 것이다. 이 결과 또한 제안하는 알고리즘이 JM 11.0 과 비교하여 움직임 추정 시간을 크게 줄이는 것을 명시적으로 나타낸다.

Vol.2, pp.1187-1190, Aug. 2004.

[7] H. J. Li, C. T. Hsu and M. J. Chen, "Fast Multiple Reference Frame Selection Method for Motion Estimation in JVT/H.264," *Proc. Asia-Pacific Conf. Circuits and System*, Vol.1, pp.605-608, Dec. 2004.

[8] Y. W. Huang, B. Y. Hsieh, S. Y. Chien, S. Y. Ma and L. G. Chen, "Analysis and Complexity Reduction of Multiple Reference Frames Motion Estimation in H.264/AVC," *IEEE Trans. Circuits and Systems for Video Technology*, Vol.16, No.4, pp.507-522, Apr. 2006.

[9] L. Shen, Z. Liu, Z. Zhang and X. Shi, "An Adaptive and Fast H.264 Multi-Frame Selection Algorithm Based on Information from Previous Searches," *Proc. Int. Conf. Multimedia and Expo*, pp.1591-1594, July 2007.

[10] C. T. Hsu, H. J. Li and M. J. Chen, "Fast Reference Frame Selection Method for Motion Estimation in JVT/H.264," *IEICE Trans. Communications*, Vol.E87-B, No.12, pp.3827-3830, Dec. 2004.

[11] Z. Wang, J. Yang, Q. Peng and C. Zhu, "An Efficient Algorithm for Motion Estimation with Multiple Reference Frames in H.264/AVC," *Int. Conf. Image and Graphics*, pp.259-262, Aug. 2007.

[12] Joint model (JM) 11.0, [Online] Available: <http://iphome.hhi.de/suehring/>

이 현 우 (Hyunwoo Lee)

준회원



2007년 2월 한양대학교 화학공학과 졸업
 2007년 3월~현재 한양대학교 대학원 전자컴퓨터통신공학과 석사과정
 <관심분야> 영상처리 및 영상압축, 에러은닉

유 증 민 (Jongmin Ryu)

정회원



2003년 2월 한양대학교 전자전기컴퓨터공학부 졸업
 2005년 2월 한양대학교 대학원 전자컴퓨터통신공학과 석사
 2005년 3월~현재 한양대학교 대학원 전자컴퓨터통신공학과 박사과정

<관심분야> 영상처리 및 영상압축, 트랜스코딩

정 제 창 (Jechang Jeong)

정회원



1980년 2월 서울대학교 전자공학과 졸업
 1982년 2월 KAIST 전기전자공학과 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 전자컴퓨터통신공학과 교수 (영상통신 및 신호처리 연구실)
 1998년 11월 27일 과학기술자상 수상
 1998년 12월 31일 정보통신부장관상 표창
 <관심분야> 영상처리 및 영상압축