제4회 정보통신분야학회

합동학술대회

<u>- 프로그램 -</u>

JCICT 2010

2010 Joint Conference on Information and Communication Technology

일시: 2010년 10월 22일(금) ~ 23일(토) 장소: 목포대학교 정보종합센터, 프라자60

- 주최 : (사)한국콘텐츠학회, (사)한국정보과학회
- 주관: (사)한국인터넷정보학회, (사)한국통신학회, (사)한국정보처리학회, (사)한국정보보호학회, (사)한국정보기술학회, (사)한국전자파학회, (사)한국컴퓨터정보학회, (사)한국해양정보통신학회, (사)한국대S학회, (사)한국정보교육학회
- 후원 : (주)삼성SDS, (주)kt, (주)동하테크, (주) 아이티센시스템즈, 한국휴렛패커드(유), 아이비씨엔씨(주)

■ JCICT 2010 논문 목차

	오후 13:20 ~ 14:20 - 정보종합센터 208호	T1-1 : 영상처리 / 멀티미디어 (1)	
번호	좌장 : 김영철 교수 (전남대학교)		
1	스테레오스코픽 디스플레이의 시각 피로도 평가 - 강민구, 신인용, 호요성	p. 3	
2	조기 종료를 이용한 H.264/AVC 고속 움직임 예측 - 이교윤, 호요성	p. 7	
3	☆ 향상된 예측 지역 탐색 방법을 위한 적응적인 탐색 - 문지희, 호요성	영역 결정 방법과 개선 과정 적용 p. 11	
4	는 초점에 기반한 색상 동일성 요소의 색상 가중치 방 - Nguyen Cao Truong Hai, 박요셉, 박혁로	법 및 이미지 검색에 적용 p. 15	

논문 제목 앞의 (☆)는 우수논문임

	오후 14:25 ~ 15:25 - 정보종합센터 208호	T1-2 : 영상처리 / 멀티미디어 (2)	
번호	좌장 : 김수형 교수 (전남대학교)		
1	복합형 카메라 시스템을 이용한 고속 다시점 깊이 영성 - 이은경, 호요성	: 생성 방법 p. 23	
2	Tensor Voting for Text Processing - Chengdong Zhang, GueeSang Lee		
3	☆ 유방 초음파 영상에서 종양 영역 검출 - 이명은, 진연연, 서인혜, 김수형		
4	모바일 환경에서의 립리딩을 위한 실시간 얼굴 및 입술 - 이재원, 홍성훈	· 영역 검출 p. 34	

	오후 15:30 ~ 16:30 - 정보종합센터 208호	T1-3 : 영상처리 / 멀티미디어 (3)	
번호	화장 : 김수형 교수 (전남대학교)		
1	허프 변환을 이용한 후판 영상에서의 라인 검출 - 강의성, 박상현, 김종호	p. 41	
2	방향성 투영 및 허프 변환에 기반한 후판 영상의 문자 영역 추출 기법 - 김종호, 김경민, 김도현, 박상현, 강의성		
3	최단경로 알고리즘을 이용한 접합 영문자 분할 - 이득용, 김민우, 오일석	p. 49	
4	☆ Analysis of Color Image Segmentation Based on T - Kavitha Madhubalan, GueeSang Lee	Tensor Voting p. 53	

조기 종료를 이용한 H.264/AVC 고속 움직임 예측

이교원, 호요성2

Fast motion estimation using early termination in H.264/AVC

Gyo-yoon Lee1, Yo-Sung Ho2

요약

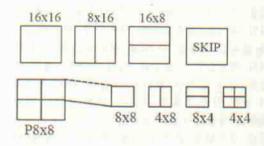
H.264/AVC는 비디오 압축 표준으로 블록 단위로 최적의 부호화 모드를 결정하기 위해 비트용-왜곡 최적화 기법(Rate-distortion optimization)을 사용하지만 복잡도가 높다는 단점이 있다. 특히, 움직임 예측은 H.264/AVC 부호기 복잡도의 50-60%를 차지한다. 그러므로 H.264/AVC의 부호화 시간을 단축하기 위해 고속으로 움직임 벡터를 예측하는 방법이 필요하다. H.264/AVC에서 사용하는 전 영역 고속 움직임 예측 방법은 탐색 영역 내의 모든 정수 화소에 대해서 움직임 벡터를 계산한다. 불필요한 영역까지 움직임 벡터를 예측하므로 계산량이 증가한다. 본 논문에서는 움직임 벡터 예측 과정에서 비트율-왜곡값의 특성과 움직임 벡터의 특징을 이용하여 조기 종료 조건을 가변적으로 결정한다. 설정된 조기 종료 조건을 통하여 불필요한 영역에 대한 예측 과정을 조기에 종료하여 움직임 예측에 소요되는 시간이 40% 감소되는 것을 실험을 통해 확인하였다.

1. 서론

H.264/AVC는 비디오 압축 성능을 높이기 위해 다양한 크기의 블록 단위 움직임 보상, 1/4화소 단위의 움직임 예측, 문맥 기반의 가변 길이 부호화(Context-based Adaptive Variable Lengh Coding) 및 문맥 기반의 이진 산술 부호화(Context-based Adaptive Binary Arithmetic Coding)등으로 높은 압축 효율을 보인다.[1]

H.264/AVC는 시간적 상관도를 이용한 인터 모드 예측과 공간적 상관도를 이용한 인트라 예측 방식을 이용한다. 인터 모드에서는 5개의 모드(SKIP, 16×16, 16×8, 8×16, P8×8)를 가지 며, P8×8 모드는 다시 각각의 8×8 블록 안에서 8×8, 8×4, 4×8, 4×4 중의 하나로 나뉜다. (그림 1)은 인터 모드에서 사용되는 블록을 나타낸 것 이다.

H.264/AVC는 압축 효율을 높이기 위해 인터 모드에서 각 블록의 크기에 맞는 최적의 움직 임 벡터를 선택한다. 최적의 움직임 벡터 선택



(그림 1) 인터모드의 블록

은 비<mark>트율-왜곡 최적화</mark> 기법을 사용하여 수행 한다.

H.264/AVC는 최적의 비트율-왜곡 지점을 찾기 위하여 참조 영상의 일정한 탐색 범위 내에서의 모든 점에 대하여 비트율-왜곡을 계산한다. 이 과정에서 매우 많은 계산적인 복잡도가요구된다.

본 논문에서는 각 참조 영상의 탐색 범위 내에서 불필요한 지점의 왜곡 계산을 생략하여, 움직임 예측 및 모드 결정 과정을 빠르게 수행하는 새로운 고속 모드 결정 방법을 제안한다.

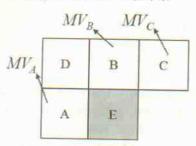
¹ 광주과학기술원 정보통신공학과 영상통신연구실(gyoyoon@gist.ac.kr),

² 광주과화기술원 정보통신공학과 교수(hoyo@gist.ac.kr)

[※] 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구 센터 지원 사업의 연구 결과로 수행되었음 (NIPA-2010-(C1090-1011-0003))

2. H.264 움직임 예측 방법

H.264/AVC 참조 소프트웨어에서는 움직임 벡터탐색 과정을 수행하기 이전에 움직임 벡터 예측 과정을 수행한다[2]. (그림 2)는 움직임 벡터 데측 방법을 나타내는 그림이다.



(그림 2) 움직임 베터 예측

현재 부호화해야 할 E블록의 움직임 벡터 부 호화 효율을 높이기 위해서 움직임 벡터 계산 이전에 인접한 A. B. C블록의 움직임 벡터 중 간값을 E블록의 움직임 벡터 예측값으로 사용 한다. A, B, C블록 중 1개의 블록만 사용할 수 있을 경우에는 사용 가능한 하나의 불록의 움 직임 벡터를 예측값으로 사용하며, 주위에 2개 의 블록만 이용할 수 있을 경우에는 움직임 벡 터의 평균값을 이용한다. 움직임 벡터 예측 방 법은 실제 움직임 벡터와 비슷한 값을 예측하 여 높은 압축 효율을 보인다. 또한 움직임 벡터 예측의 정확도가 높다면, 예측된 지점 근처에서 현재 부호화할 블록과 가장 유사한 블록이 나 올 확률이 높다. H.264는 이런 특성을 이용하여 확률이 높은 지점부터 나선형으로 움직임 벡터 탐색 과정을 수행한다. (그림 3)은 나선형 탐색 순서를 나타낸다.

H.264/AVC 참조 소프트웨어에서 최적의 움

22	18	12	20	23
14	6	4	8	15
10	2	1	3	11
16	9	5	7	17
24	19	13	21	25

(그림 3) 나선형 탐색순서

직임 벡터를 예측하기 위해서는 탐색 영역 내 에서 나선형 방향으로 탐색 위치를 바꿔가며, 각 위치에서 원 영역과 참조 영역과의 차분값 의 절대합인 SAD(Sum of Absolute Distortion)를 구한다. 계산된 SAD는 (수식 1) 에 따라, 참조 영상과 움직임 벡터의 비트율과 더해져 움직임 예측의 선택 기준인 J_{motion} 이 된다.

$$J_{motion}(MV.REF | \lambda_{motion}) =$$

 $SAD(s, r(MV, REF)) + \lambda_{motion} \cdot R(MV, REF)$ (수식 1)

여기서 λ_{motion} 는 Lagrangian 계수이고, $0.85 \times 2^{Q/3}$ 의 제곱근 값을 갖는다. Q는 양자화계수값이다. R(s,r(MV,REF))는 움직임 벡터와참조 영상을 부호화하는데 필요한 비트량을 나타낸다. 참조 영역내의 각 지점에서 계산된 비트율-왜곡(J_{motion})값이 가장 작은 지점의 참조 영상과 움직임 벡터가 현재 블록의 움직임 정보로 선정된다.

3. 탐색 종료 조건을 이용한 고속 모드 결정

본 논문에서는 탐색 영역 내의 비트율-왜곡 값을 계산하는 과정에서 조건을 이용하여 계산 을 조기에 종료하여 움직임 벡터의 탐색 시간 을 단축하는 방법을 제안한다.

H.264에서는 탐색 영역의 크기를 사용자의 설정에 따라 조절할 수 있다. 일반적으로 많이 사용하는 탐색 영역 크기인 32를 사용할 경우, 움직임 벡터 예측된 지점으로부터 오른쪽으로 32, 왼쪽 32, 위쪽 32, 아래쪽 32의 블록 내에서 탐색을 한다. 즉 가로 65, 세로 65의 블록 내에서 탐색을 한다. 즉 가로 65, 세로 65의 블록 내에서 탐색을 수행하므로 총 4225개의 위치에서 16×16블록의 SAD를 계산하게 되어 그 복잡도가 굉장히 높아지며, 탐색 시간 또한 길어지게된다. 하지만, 일반적으로 탐색 영역 내에서도 대부분의 위치는 움직임 예측 과정에서 불필요한 지점이다. 따라서, 특정 조건을 이용하여 불필요한 지점들의 SAD 계산을 종료하여, 탐색시간을 단축시킨다.

H.264/AVC의 고속 전 영역 탐색 방법은 (그림 4)와 같이 탐색 영역내의 모든 점에서 4×4 블록 단위로 16×16블록의 SAD를 계산한다. 구해진 0~15블록의 SAD는 각 모드의 J_{motion} 계산 과정에서 모드 별 블록 크기에 맞게 더하여계산하고, 각 모드별로 SAD 계산을 수행하지 않고도 이미 계산된 16×16블록의 SAD 결과값 을 이용하여 연산을 가능하게 하여 연산 속도 를 빠르게한다.

16×16블록의 SAD 연산은 수평 방향으로 순 차적으로 계산된다. 제안된 방법에서는 한 라인 의 SAD 값이 더해지면 더해진 SAD값과 임계 값의 크기를 비교하여 SAD 값이 임계값보다 크면, SAD 계산을 종료하고 다음 탐색 위치로 이동한다.



(그림 4) 탐색 영역내의 SAD 연산

제안된 방법을 수행하기 위해서는 임계값을 정하는 과정이 중요하다. 임계값은 현재 탐색 위치까지 계산한 16×16블록 크기의 SAD값들 중 최소값과 움직임 벡터의 비트율 성분의 합 (J_{mottom})이다. 이 방법은 16×16블록 모드에서는 정확하지만, 16×8, 8×16 혹은 P8×8 모드의 움 직임 벡터를 찾는 과정에서는 약간의 오차가 발생할 수도 있다. 하지만 일반적으로 16×16불 록의 크기에서 가장 유사한 블록이 그보다 작 은 블록에서도 가장 유사한 블록일 확률이 높 다. 임계값은 그 보다 작은 16×16블록의 SAD 값이 나올 경우 최소값으로 다시 재정의 한다. 일반적으로 정확한 움직임 예측과 나선형 탐색 방식을 이용할 경우, 최적의 위치는 탐색 과정 의 초기에 나타날 확률이 높기 때문에 효율은 종대된다.

4. 실험 결과 및 분석

본 논문에서는 제안한 전 영역 고속 모드 결정 방법을 H.264/AVC 참조 소프트웨어 JM 12.4에 구현하였다[3]. 해상도 352×288 크기의 움직임이 다양한 각기 다른 5개의 비디오 시퀀스fmf 대상으로 IPPP 구조의 베이스라인 (Baseline) 프로파일로 부호화했다. 움직임 예측

에서 참조 영상은 5개를 사용했으며, 움직임 예측을 위한 탐색 범위는 32이다. 비트율-왜곡 곡선을 유도하기 위한 양자화 계수는 27을 사용하였다. [표 1]은 실험에 사용된 부호화 계수를 요약한 표이다.

제안한 움직임 예측 방법과 H.264 참조 소프

[표 1] 부호화 변수

실험변수	갋
ProfileIDC	77
LevelIDC	51
Search Range	32
Reference	5
QP	27

트 웨어의 고속 모드 결정 방법의 성능 비교를 위해 PSNR과 비트율을 사용했다[4].

H.264의 고속 모드 결정 방법과 제안한 고속 모드 결정 방법의 수행 시간 비교를 위해 △T 를 다음과 같이 정의한다[5].

$$\Delta T = \frac{Time(H.264 고속모드) - Time(제안한방법)}{Time(H.264 고속모드)} \times 100[\%]$$
 (수식 ½

[표 2]는 H.264/AVC와 제안한 알고리즘의 부호화 시간을 비교한 것이다.

[표 2] 부호화 시간 비교

	H.264	Proposed	△ Time
Akiyo	746,415	447.506	-40.05%
Foreman	760.720	474.742	-37.59%
Mobile	765.302	480.513	-37.21%
News	751.967	463.351	-38.38%
Paris	746.464	441.249	-40,89%
Average			-38.82%

[표 2] 실험 결과로부터, 움직임이 적은 단순 한 영상에서의 부호화 시간이 더욱 단축되는 성능을 보인다.

[표 3]과 [표 4]는 각각 H.265/AVC와 제안한 방법의 비트율과 PSNR의 성능 비교를 나타낸 다.

제안한 알고리즘은 H.264/AVC 소프트웨어와 비교하여 화질적 성능(PSNR)은 차이가 거의 없었다. 하지만 각 영상의 특성에 따라 부호화 비트율은 평균 1.28%증가를 보였으나, Akiyo 시퀀스와 같은 움직임이 적은 시퀀스는 비트율 변화가 거의 없었다.

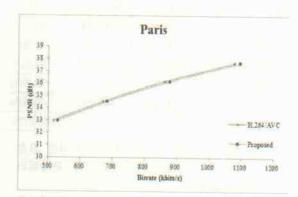
[표 3] 부호화 비트율(kb/s) 비교

실험영상	H.264	Proposed	△bitrate
Akiyo	116.01	116.06	0%
Foreman	428.62	437.76	2.13%
Mobile	2099.02	2102.64	0.17%
News	231.90	236.13	1.82%
Paris	723.67	740.58	2.30%
	Average		1.28%

[표 4] 부호화 성능(PSNR)비교

실험영상	H.264	Proposed
Akiyo	40.60	40.60
Foreman	37.93	37.91
Mobile	36.31	36.31
News	39.17	39.17
Paris	36.89	36.89

(그림 5)는 정확한 실험 결과 측정을 위하여 부호화 비트율이 가장 많이 상승한 Paris 시퀀 스에 대하여, 비트율-왜곡 그래프를 나타낸 것 이다.



(그림 5) 비트율-왜곡 그래프

양자화 계수의 변화에 따른 부호화 시간의 변화는 없다. 제안한 알고리즘의 비트율은 양자 화 계수값의 변화에 관계없이 비슷한 성능을 보이며, H.264 참조소프트웨어와 성능과 큰 차 이가 없음을 알 수 있다.

V. 결론

본 논문에서는 움직임 벡터의 선택 기준인 비트율-왜곡 값을 이용하여 움직임 예측 계산 과정의 종료 조건을 설정한다. 설정한 종료 조 건을 이용하여 탐색 영역내의 왜곡 정도가 높 은 불필요한 위치에서의 계산 과정을 조기에 종료하여 움직임 벡터를 결정하는 계산 과정을 빠르게 한다. 이 과정에서 나선형 탐색 방법과 움직임 벡터 예측 과정을 이용하여 탐색 과정 의 초기에 최적의 움직임 벡터를 선정하여 부 호화 시간을 더욱 효율적으로 감소시킨다. 종료 조건은 여러 개의 참조 화면을 이용할 경우 각 참조 화면에 공통의 종료 조건을 적용하여 참 조 영상의 숫자가 늘어나더라도 효율적으로 부 호화를 수행한다. 제안한 방법은 H.264/AVC 고속 전 영역 탐색 과정과 비교하여, 평균 1.28%의 비트율이 증가하는 대신 평균 38.82% 의 부호화 시간을 감소시켰다.

참고문헌

- [1] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," IEEE Transactions on Circuits and System Videos for Technology, vol. 13, no. 7, pp. 560-576, July 2003.
- [2] I. Richardson, H.264 and MPEG-4 Video Compression. Wiley, 2003.
- [3] JVT reference software version 12.4, available online at: http://iphome.hhi.de/suehring/tml/donload/old_jm/
- [4] G. Bjontegaard. "Calculation of Average PSNR Difference between RD-curve," ITU-T Q6/16, Doc. VCEG-M33, April 2001.
- [5] G. Y. Kim, B. Y. Yoon, and Y. S. Ho, "A Fast Inter Mode Decision Algorithm in H.264/AVC for IPTV Broadcasting Services." Proc. VCIP, pp. 65081L:1~10, Jan. 2007.

조기 종료를 이용한 H.264/AVC 고속 움직임 예측

이교 θ^1 , 호요성 2

Fast motion estimation using early termination in H.264/AVC

Gyo-yoon Lee¹, Yo-Sung Ho²

요 약

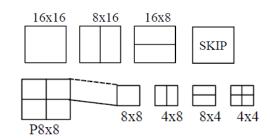
H.264/AVC는 비디오 압축 표준으로 블록 단위로 최적의 부호화 모드를 결정하기 위해 비트율-왜곡 최적화 기법(Rate-distortion optimization)을 사용하지만 복잡도가 높다는 단점이 있다. 특히, 움직임 예측은 H.264/AVC 부호기 복잡도의 50~60%를 차지한다. 그러므로 H.264/AVC의 부호화 시간을 단축하기 위해 고속으로 움직임 벡터를 예측하는 방법이 필요하다. H.264/AVC에서 사용하는 전 영역 고속 움직임 예측 방법은 탐색 영역 내의 모든 정수 화소에 대해서 움직임 벡터를 계산한다. 불필요한 영역까지 움직임 벡터를 예측하므로 계산량이 증가한다. 본 논문에서는 움직임 벡터 예측 과정에서 비트율-왜곡값의 특성과 움직임 벡터의 특징을 이용하여 조기 종료 조건을 가변적으로 결정한다. 설정된 조기 종료 조건을 통하여 불필요한 영역에 대한 예측 과정을 조기에 종료하여움직임 예측에 소요되는 시간이 40% 감소되는 것을 실험을 통해 확인하였다.

1. 서론

H.264/AVC는 비디오 압축 성능을 높이기위해 다양한 크기의 블록 단위 움직임 보상, 1/4화소 단위의 움직임 예측, 문맥 기반의 가변 길이 부호화(Context-based Adaptive Variable Lengh Coding) 및 문맥 기반의 이진 산술 부호화(Context-based Adaptive Binary Arithmetic Coding)등으로 높은 압축 효율을 보인다.[1]

H.264/AVC는 시간적 상관도를 이용한 인터 모드 예측과 공간적 상관도를 이용한 인트라 예측 방식을 이용한다. 인터 모드에서는 5개의 모드(SKIP, 16×16, 16×8, 8×16, P8×8)를 가지며, P8×8 모드는 다시 각각의 8×8 블록 안에서 8×8, 8×4, 4×8, 4×4 중의하나로 나뉜다. (그림 1)은 인터 모드에서 사용되는 블록을 나타낸 것이다.

H.264/AVC는 압축 효율을 높이기 위해 인 터 모드에서 각 블록의 크기에 맞는 최적의 움



(그림 1) 인터모드의 블록

직임 벡터를 선택한다. 최적의 움직임 벡터 선택은 비트율-왜곡 최적화 기법을 사용하여 수행한다.

H.264/AVC는 최적의 비트율-왜곡 지점을 찾기 위하여 참조 영상의 일정한 탐색 범위 내에서의 모든 점에 대하여 비트율-왜곡을 계산한다. 이 과정에서 매우 많은 계산적인 복잡도가 요구된다.

본 논문에서는 각 참조 영상의 탐색 범위 내에서 불필요한 지점의 왜곡 계산을 생략하여, 움직임 예측 및 모드 결정 과정을 빠르게 수행

¹ 광주과학기술원 정보통신공학과 영상통신연구실(gyoyoon@gist.ac.kr),

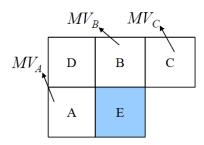
² 광주과학기술원 정보통신공학과 교수(hoyo@gist.ac.kr)

^{**} 본 연구는 지식경제부 및 정보통신산업진홍원의 대학 IT 연구 센터 지원 사업의 연구 결과로 수행되었음 (NIPA-2010-(C1090-1011-0003))

하는 새로운 고속 모드 결정 방법을 제안한다.

2. H.264 움직임 예측 방법

H.264/AVC 참조 소프트웨어에서는 움직임 벡터탐색 과정을 수행하기 이전에 움직임 벡터 예측 과정을 수행한다[2]. (그림 2)는 움직임 벡터 예측 방법을 나타내는 그림이다.



(그림 2) 움직임 벡터 예측

현재 부호화해야 할 E블록의 움직임 벡터 부 호화 효율을 높이기 위해서 움직임 벡터 계산 이전에 인접한 A, B, C블록의 움직임 벡터 중 간값을 E블록의 움직임 벡터 예측값으로 사용 한다. A. B. C블록 중 1개의 블록만 사용할 수 있을 경우에는 사용 가능한 하나의 블록의 움직임 벡터를 예측값으로 사용하며, 주위에 2 개의 블록만 이용할 수 있을 경우에는 움직임 벡터의 평균값을 이용한다. 움직임 벡터 예측 방법은 실제 움직임 벡터와 비슷한 값을 예측 하여 높은 압축 효율을 보인다. 또한 움직임 벡 터 예측의 정확도가 높다면, 예측된 지점 근처 에서 현재 부호화할 블록과 가장 유사한 블록 이 나올 확률이 높다. H.264는 이런 특성을 이용하여 확률이 높은 지점부터 나선형으로 움 직임 벡터 탐색 과정을 수행한다. (그림 3)은 나선형 탐색 순서를 나타낸다.

22	18	12	20	23
14	6	4	8	15
10	2	1	3	11
16	9	5	7	17
24	19	13	21	25

(그림 3) 나선형 탐색순서

H.264/AVC 참조 소프트웨어에서 최적의 움직임 벡터를 예측하기 위해서는 탐색 영역 내에서 나선형 방향으로 탐색 위치를 바꿔가며. 각 위치에서 원 영역과 참조 영역과의 차분값의 절대합인 SAD(Sum of Absolute Distortion)를 구한다. 계산된 SAD는 (수식 1)에 따라, 참조 영상과 움직임 벡터의 비트율과 더해져 움직임 예측의 선택 기준인 J_{motion} 이된다.

$$\begin{split} J_{motion}(MV, REF \mid & \lambda_{motion}) = \\ SAD(s, r(MV, REF)) + & \lambda_{motion} \bullet R(MV, REF) \end{split}$$
 (수식 1)

여기서 λ_{motion} 는 Lagrangian 계수이고, $0.85 \times 2^{Q/3}$ 의 제곱근 값을 갖는다. Q는 양자화계수값이다. R(s,r(MV,REF))는 움직임 벡터와참조 영상을 부호화하는데 필요한 비트량을 나타낸다. 참조 영역내의 각 지점에서 계산된 비트율-왜곡 (J_{motion}) 값이 가장 작은 지점의 참조 영상과 움직임 벡터가 현재 블록의 움직임 정보로 선정된다.

3. 탐색 종료 조건을 이용한 고속 모드 결정

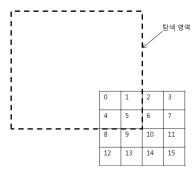
본 논문에서는 탐색 영역 내의 비트율-왜곡 값을 계산하는 과정에서 조건을 이용하여 계산 을 조기에 종료하여 움직임 벡터의 탐색 시간 을 단축하는 방법을 제안한다.

H.264에서는 탐색 영역의 크기를 사용자의설정에 따라 조절할 수 있다. 일반적으로 많이사용하는 탐색 영역 크기인 32를 사용할 경우,움직임 벡터 예측된 지점으로부터 오른쪽으로 32, 왼쪽 32, 위쪽 32, 아래쪽 32의 블록 내에서 탐색을 한다. 즉 가로 65, 세로 65의 블록 내에서 탐색을 수행하므로 총 4225개의 위치에서 16×16블록의 SAD를 계산하게 되어그 복잡도가 굉장히 높아지며, 탐색 시간 또한길어지게 된다. 하지만, 일반적으로 탐색 영역내에서도 대부분의 위치는 움직임 예측 과정에서 불필요한 지점이다. 따라서, 특정 조건을 이용하여 불필요한 지점들의 SAD 계산을 종료하여, 탐색 시간을 단축시킨다.

H.264/AVC의 고속 전 영역 탐색 방법은 (그림 4)와 같이 탐색 영역내의 모든 점에서 4×4 블록 단위로 16×16 블록의 SAD를 계산한다. 구해진 $0\sim15$ 블록의 SAD는 각 모드의 J_{motion} 계산 과정에서 모드 별 블록 크기에 맞게 더하여 계산하고, 각 모드별로 SAD 계산을

수행하지 않고도 이미 계산된 16×16블록의 SAD 결과값을 이용하여 연산을 가능하게 하여 연산 속도를 빠르게한다.

16×16블록의 SAD 연산은 수평 방향으로 순차적으로 계산된다. 제안된 방법에서는 한 라인의 SAD 값이 더해지면 더해진 SAD값과 임계값의 크기를 비교하여 SAD 값이 임계값보다크면, SAD 계산을 종료하고 다음 탐색 위치로이동한다.



(그림 4) 탐색 영역내의 SAD 연산

제안된 방법을 수행하기 위해서는 임계값을 정하는 과정이 중요하다. 임계값은 현재 탐색 위치까지 계산한 16×16블록 크기의 SAD값들 중 최소값과 움직임 벡터의 비트율 성분의 합 (J_{motion}) 이다. 이 방법은 16×16 블록 모드에서 는 정확하지만. 16×8. 8×16 혹은 P8×8 모드 의 움직임 벡터를 찾는 과정에서는 약간의 오 차가 발생할 수도 있다. 하지만 일반적으로 16×16블록의 크기에서 가장 유사한 블록이 그 보다 작은 블록에서도 가장 유사한 블록일 확 률이 높다. 임계값은 그 보다 작은 16×16블록 의 SAD값이 나올 경우 최소값으로 다시 재정 의 한다. 일반적으로 정확한 움직임 예측과 나 선형 탐색 방식을 이용할 경우, 최적의 위치는 탐색 과정의 초기에 나타날 확률이 높기 때문 에 효율은 증대된다.

4. 실험 결과 및 분석

본 논문에서는 제안한 전 영역 고속 모드 결정 방법을 H.264/AVC 참조 소프트웨어 JM 12.4에 구현하였다[3]. 해상도 352×288 크기의 움직임이 다양한 각기 다른 5개의 비디오시퀀스fmf 대상으로 IPPP 구조의 베이스라인

(Baseline) 프로파일로 부호화했다. 움직임 예측에서 참조 영상은 5개를 사용했으며, 움직임 예측을 위한 탐색 범위는 32이다. 비트율-왜곡곡선을 유도하기 위한 양자화 계수는 27을 사용하였다. [표 1]은 실험에 사용된 부호화 계수를 요약한 표이다.

[표 1] 부호화 변수

실험변수	값
ProfileIDC	77
LevelIDC	51
Search Range	32
Reference	5
QP	27

제안한 움직임 예측 방법과 H.264 참조 소 프트 웨어의 고속 모드 결정 방법의 성능 비교 를 위해 PSNR과 비트율을 사용했다[4].

H.264의 고속 모드 결정 방법과 제안한 고속 모드 결정 방법의 수행 시간 비교를 위해 $\triangle T$ 를 다음과 같이 정의한다[5].

$$\Delta T = \frac{Time(H.264$$
고속모드) $- Time($ 제안한방법)}{Time(H.264고속모드)} $\times 100[\%]$ (수식 2)

[표 2]는 H.264/AVC와 제안한 알고리즘의 부호화 시간을 비교한 것이다.

[표 2] 부호화 시간 비교

	Н. 264	Proposed	△Time
Akiyo	746. 415	447. 506	-40.05%
Foreman	760. 720	474.742	-37.59%
Mobile	765.302	480. 513	-37. 21%
News	751. 967	463. 351	-38.38%
Paris	746. 464	441. 249	-40.89%
Average			-38, 82%

[표 2] 실험 결과로부터, 움직임이 적은 단순한 영상에서의 부호화 시간이 더욱 단축되는 성능을 보인다.

[표 3]과 [표 4]는 각각 H.265/AVC와 제 안한 방법의 비트율과 PSNR의 성능 비교를 나타낸다.

제안한 알고리즘은 H.264/AVC 소프트웨어와 비교하여 화질적 성능(PSNR)은 차이가 거의 없었다. 하지만 각 영상의 특성에 따라 부호

화 비트율은 평균 1.28%증가를 보였으나, Akiyo 시퀀스와 같은 움직임이 적은 시퀀스는 비트율 변화가 거의 없었다.

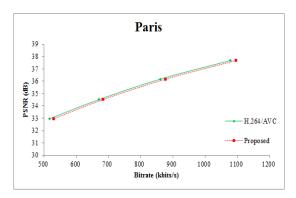
丑	3]	부호화	비트율	(kb/s)	目記
----------	----	-----	-----	--------	----

실험영상	Н. 264	Proposed	△bitrate
Akiyo	116.01	116.06	0%
Foreman	428.62	437.76	2.13%
Mobile	2099.02	2102.64	0.17%
News	231.90	236. 13	1.82%
Paris	723.67	740. 58	2.30%
Average			1.28%

[표 4] 부호화 성능(PSNR)비교

실험영상	Н. 264	Proposed
Akiyo	40.60	40.60
Foreman	37. 93	37. 91
Mobile	36. 31	36. 31
News	39. 17	39. 17
Paris	36. 89	36. 89

(그림 5)는 정확한 실험 결과 측정을 위하여 부호화 비트율이 가장 많이 상승한 Paris 시퀀 스에 대하여, 비트율-왜곡 그래프를 나타낸 것 이다.



(그림 5) 비트율-왜곡 그래프

양자화 계수의 변화에 따른 부호화 시간의 변화는 없다. 제안한 알고리즘의 비트율은 양자 화 계수값의 변화에 관계없이 비슷한 성능을 보이며, H.264 참조소프트웨어와 성능과 큰 차이가 없음을 알 수 있다.

V. 결론

본 논문에서는 움직임 벡터의 선택 기준인 비트율-왜곡 값을 이용하여 움직임 예측 계산 과정의 종료 조건을 설정한다. 설정한 종료 조 건을 이용하여 탐색 영역내의 왜곡 정도가 높 은 불필요한 위치에서의 계산 과정을 조기에 종료하여 움직임 벡터를 결정하는 계산 과정을 빠르게 한다. 이 과정에서 나선형 탐색 방법과 움직임 벡터 예측 과정을 이용하여 탐색 과정 의 초기에 최적의 움직임 벡터를 선정하여 부 호화 시간을 더욱 효율적으로 감소시킨다. 종료 조건은 여러 개의 참조 화면을 이용할 경우 각 참조 화면에 공통의 종료 조건을 적용하여 참 조 영상의 숫자가 늘어나더라도 효율적으로 부 호화를 수행한다. 제안한 방법은 H.264/AVC 고속 전 영역 탐색 과정과 비교하여, 평균 1.28%의 비트율이 증가하는 대신 평균 38.82% 의 부호화 시간을 감소시켰다.

참고문헌

- [1] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," IEEE Transactions on Circuits and System Videos for Technology, vol. 13, no. 7, pp. 560-576, July 2003.
- (2) I. Richardson, H.264 and MPEG-4 Video Compression. Wiley, 2003.
- [3] JVT reference software version 12.4, available online at: http://iphome.hhi.de/suehring/tml/donload/old.jm/
- (4) G. Bjontegaard. "Calculation of Average PSNR Difference between RD-curve," ITU-T Q6/16, Doc. VCEG-M33, April 2001.
- (5) G. Y. Kim, B. Y. Yoon, and Y. S. Ho, "A Fast Inter Mode Decision Algorithm in H.264/AVC for IPTV Broadcasting Services." Proc. VCIP, pp. 65081L:1~10, Jan. 2007.