

# 다시점 비디오 부호화를 위한 효과적인 초기 종료 움직임 추정 기법

윤효순<sup>†</sup>, 김미영<sup>\*\*</sup>

## 요 약

다시점 비디오는 하나의 3차원 장면을 여러 시점에서 다수의 카메라로 촬영한 동영상으로 다시점 비디오 부호화는 카메라의 수에 비례하여 데이터의 양이 증가하기 때문에 계산량을 줄일 수 있는 다시점 비디오 부호화 기술이 필요하다. 본 논문에서 다시점 비디오의 부호화를 위한 움직임 추정의 계산량을 줄이면서 화질을 유지하는 움직임 추정 기법을 제안한다. 제안한 움직임 추정 기법은 움직임 벡터의 분포 특성을 이용한 효과적인 초기 종료 움직임 추정 기법이다. 제안한 움직임 추정 기법은 계층적 움직임 추정 기법으로 다중 사각형 탐색 패턴, 래스터 탐색 패턴, 수정된 다이아몬드 탐색 패턴 그리고 작은 다이아몬드 탐색으로 이루어져있다. 제안한 기법의 성능은 JMVC의 고속 움직임 추정 기법인 TZ 탐색 기법과 전역 탐색 기법(FS)의 성능과 비교한 경우, 영상 화질과 비트량을 비슷하지만 움직임 추정 속도를 각각 약 1.7~4.5배, 90배 향상시킨다.

## An Effective Early Termination Motion Estimation Method for Multi-view Video Coding

Hyo-sun Yoon<sup>†</sup>, Mi-Young Kim<sup>\*\*</sup>

## ABSTRACT

Multi-view video is obtained by capturing one three-dimensional scene with many cameras at different positions. Multi-view video coding requires high computational complexity. To reduce computational complexity and maintain the image quality, an effective early termination motion estimation method is proposed in this paper. The proposed method exploiting the characteristic of motion vector distribution uses a hierarchical search strategy. This strategy method consists of multi-grid square search pattern, modified diamond search pattern, small diamond search pattern and raster search pattern. Experiment results show that the speedup improvement of the proposed method over TZ search method and FS(Full Search) method JMVC (Joint Multiview Video Coding) can be up to 1.7 ~4.5 times and 90 times faster respectively while maintaining similar video quality and bit rates.

**Key words:** multi-view video(다시점 비디오), temporal correlation(비디오 압축), motion estimation(움직임 추정), motion vector(움직임 벡터)

※ 교신저자(Corresponding Author): 김미영, 주소: 전라남도 담양군 담양읍 향교리 262번지(517-802), 전화: 061) 380-8651, FAX: 061) 380-8628, E-mail: kimmee@dorip.ac.kr  
접수일: 2013년 9월 27일, 수정일: 2014년 1월 20일  
완료일: 2014년 2월 20일

<sup>†</sup> 정회원, 전남대학교 전자컴퓨터공학부  
(E-mail: estheryoon@hotmail.com)

<sup>\*\*</sup> 정회원, 전남도립대학교 보건의료과

※ 이 논문은 2013년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업을 지원을 받아 수행된 연구임 (No. 2010-0024120)

## 1. 서 론

비약적인 멀티미디어 시스템의 발달로 실감 미디어 콘텐츠에 대한 수요가 급증하고 있고 다양한 멀티미디어 콘텐츠를 사용자에게 제공할 수 있게 되었다 [1]. 이에 따라 3차원 비디오의 중요성이 커지고 있으며 다시점 비디오는 3차원 영상에 대한 다양한 사용자의 요구를 충족시킬 수 있는 대안으로 주목받고 있다.

다시점 비디오는 여러 개의 카메라로 동시에 하나의 3차원 장면을 촬영한 동영상의 집합으로 다시점 비디오는 사용자에게 임의 시점을 제공하며 여러 시점의 영상들을 합성하여 보다 넓은 화면을 제공할 수 있다. 그러나 다시점 비디오는 카메라의 수에 비례하여 데이터의 양이 증가하므로 다시점 비디오를 효율적으로 부호화하는 기술이 필요하다[2].

다시점 비디오 부호화는 영상들 사이에 존재하는 시간적 상관성과 공간적 상관성을 이용하여 영상들 사이에 존재하는 중복된 데이터를 제거함으로써 데이터를 압축한다. 중복된 데이터들을 제거하기 위하여 움직임 추정을 수행한다.

영상 화질과 인코더의 속도에 영향을 주는 움직임 추정은 동영상 압축에서 중요한 역할을 하지만 전체 인코더에서 많은 계산량을 요구한다. 움직임 추정에 소요되는 계산량을 줄이기 위하여 다음과 같은 고속 움직임 추정 기법들이 제안되었다. 대표적인 고속 움직임 추정 기법에는 3단계 탐색(Three Step Search: TSS)[3], 새로운 3단계 탐색(New Three Step Search: NTSS)[4], 2차원 로그형 탐색(2 Dimension LOGarithmic search:2DLOG)[5], 4단계 탐색(Four Step Search:4SS)[6], 다이아몬드 탐색(Diamond Search: DS) [7,8], 2단계 탐색(2 Step Search:2SS) [9,10], 움직임 벡터를 추정하는 움직임 벡터 필드 적응적 탐색 기법(MVFAST)[11], 육각형 탐색 기법(Hexagonal Search:HS)[12] 그리고 예측된 움직임 벡터 필드 적응적 탐색기법(PMVFAST)[13] 등이 있다.

고속 움직임 추정 기법[3-13]들은 움직임이 작은 영상이나 영상 크기가 작은 경우에 움직임 벡터 추정을 잘 수행하지만 영상의 움직임이 크거나 영상의 크기가 큰 경우에 움직임 추정 시 국부적 최소화 문제에 빠지게 되므로 정확한 움직임 벡터를 추정할 수 없으므로 영상 화질 저하를 가져온다. 영상의 움

직임이 크거나 영상 크기가 상대적으로 큰 영상에서 움직임 벡터 추정을 잘 수행할 수 있는 비대칭 다중 육각형 탐색 기법[14]을 사용하여 화질 저하를 개선하였다.

위에서 나열한 기법[3-14]들은 일반적으로 단일 시점 영상에서 사용되는 움직임 추정 기법이다. 다시점 비디오를 압축할 때 사용되는 고속 움직임 추정 기법에는 TZ 기법[15] 그리고 TZH 기법[16][18] 등이 있다. TZ 기법은 초기에 탐색 영역 전역에 탐색 점들을 배치해 움직임이 작은 블록에 대해 불필요한 탐색 점들을 검사함으로써 많은 계산량을 요구한다. TZH 기법은 TZ 기법의 계산량을 줄이기 위하여 TZ 기법에서 사용하고 있는 다이아몬드 그리드 패턴 대신 육각형 그리드 탐색 패턴을 이용하였다.

본 논문에서는 다시점 비디오 부호화를 위한 움직임 추정의 계산량을 줄이면서 화질을 유지하는 움직임 추정 기법을 제안한다. 제안한 움직임 추정 기법은 움직임 벡터의 분포 특성을 이용한 계층적 움직임 추정 기법으로 다중 사각형 탐색 패턴, 래스터 탐색 패턴, 수정된 다이아몬드 탐색 패턴[17] 그리고 작은 다이아몬드 탐색[7,8]으로 이루어져있다. 다중 사각형 탐색 패턴은 움직임 벡터가 탐색 영역의 중심부분에 많이 분포하는 사실을 이용하여 탐색 점들을 배치한다. 본 논문에서 제안된 움직임 추정 기법은 국부적 최소화 문제를 해결하기 위하여 탐색 영역 내에 탐색 점들을 규칙적, 대칭적으로 배치한다. 그리고 블록 움직임의 크기에 따라 적절한 탐색 패턴을 사용함으로써 움직임 추정 계산량을 줄였다. 특히, 움직임이 작은 블록에 대하여 불필요한 탐색 점들을 줄임으로써 초기에 움직임 추정을 마친다.

본 논문의 구성은 다음과 같다. 2장에서는 JMVC의 고속 움직임 추정 기법인 TZ 기법에 대하여 설명하고, 3장에서는 본 논문에서 제안한 움직임 추정 기법을 기술한다. 그리고 4장에서는 TZ 기법, TZH 기법 그리고 FS 기법과 성능을 비교한 후, 마지막으로 5장에서는 결론을 맺는다.

## 2. TZ기법

JMVC의 고속 움직임 추정 기법은 TZ 기법이다. TZ 기법의 탐색 패턴들은 그림 1과 같다. TZ 기법은 다음과 같은 알고리즘을 수행하여 움직임 벡터를 추

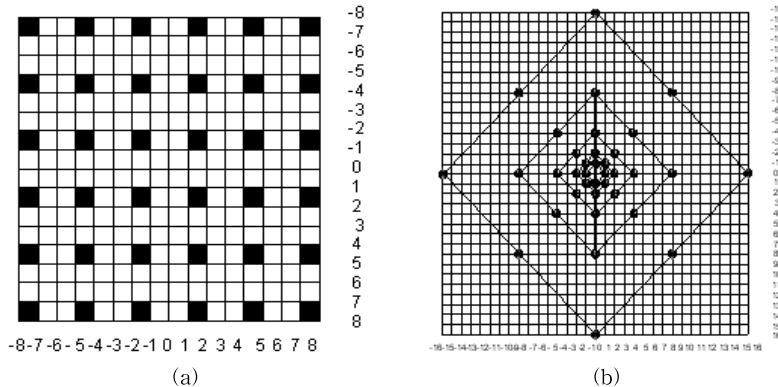


그림 1. TZ 기법의 탐색 패턴들. (a) Initial grid search pattern (stride length 16), (b) Raster search pattern (raster length=3)

정한다.

**[단계1]:** 움직임 벡터 예측(Motion Vector Prediction) 메디안 예측, 상위 모드를 이용한 예측, 이전 화면의 대응 블록을 이용한 예측, 이웃한 참조 화면을 이용한 예측들을 사용하여 초기 탐색 시작점을 결정한다.

**[단계2]:** 초기 그리드 탐색 (Initial Grid search) 초기 탐색 시작점을 탐색의 원점으로 초기 그리드 탐색 패턴을 사용한다. 초기 그리드 탐색 패턴 stride length의 범위는 1, 2, 4, 8, 16, 32, 64이다. 그림 1(a)는 stride length가 16인 초기 그리드 탐색 패턴이다. 그림 1(a)와 같은 방법으로 탐색 범위에 탐색 점들을 배치한다. 탐색 점들 중 최소 비용함수 값을 가지는 점이 [단계2]의 최적 탐색점이다.

**[단계3]:** 초기 탐색 시작점과 [단계 2]의 최적 탐색점 사이의 거리(uiBestDistance)를 구한다.

uiBestDistance가 0이면 초기 탐색 시작점을 움직임 벡터로 추정하고 움직임 추정을 마친다. 만약 uiBestDistance가 래스터 탐색의 iRaster (stride length) 보다 크면 [단계4]를 수행하고 그렇지 않으면 [단계5]를 수행한다.

**[단계4]:** 래스터 탐색 (Raster search)

그림 1(b)는 raster length가 3인 래스터 탐색 패턴이다. 그림 1(B)와 같은 방법으로 탐색 영역 내에 탐색 점들을 배치한다. 탐색 점들 중 최소 비용 함수 값을 가지는 점이 [단계4]의 최적 탐색점이다.

**[단계5]:** 정교한 탐색 (Refinement search)

탐색 시작점을 변경한 후 [단계2]의 초기 그리드 탐색을 수행한다. 탐색 점들 중 최소 비용함수 값을

가지는 점이 [단계5]의 최적 탐색점이다. 그리고 변경된 탐색 시작점과 [단계 5]의 최적 탐색점 사이의 거리(uiBestDistance)를 구한다. uiBestDistance가 0 이면 [단계5]의 최적 탐색점을 움직임 벡터로 추정하고 움직임 추정을 마친다. 그렇지 않으면 [단계5]를 반복 수행한다.

### 3. 제안한 예측구조

TZ 기법에서 사용하는 초기 그리드 탐색은 탐색 영역 전역에 탐색 점들을 배치한 후 최적의 탐색 점을 찾고 래스터 탐색 역시 탐색 영역 전역에 탐색 점들을 수직, 수평 방향으로 배치하여 최적의 탐색 점을 찾는다. 이것은 초기 그리드 탐색에서 iRaster (래스터 탐색의 탐색 길이: 3) 보다 큰 탐색 영역에 있는 탐색 점들을 검사할 필요가 없다는 것을 의미한다. 즉, 탐색 시작점과 최적의 탐색점 사이의 거리(uiBestDistance)가 iRaster보다 큰 경우, 래스터 탐색을 수행할 때 초기 그리드 탐색 점들이 배치된 영역에 탐색 점들을 배치하여 검사하기 때문이다.

본 논문에서 제안한 다중 사각형 탐색 패턴은 움직임 추정 초기에 탐색 영역의 원점을 중심으로 4픽셀 이내에 탐색 점들을 배치하여 움직임 벡터를 추정하는데 이는 다음과 같은 실험 결과를 이용하였다. 여러 영상에서 움직임 벡터들을 조사한 결과, 움직임 벡터가 탐색 영역의 원점을 중심으로 반경 1화소 이내, 2화소 이내, 3화소 이내 그리고 4화소 이내에 분포할 확률을 계산한 결과는 표 1와 같다. 그림 2(a)의 다중 사각형 탐색 패턴은 표 1에서와 같이 움직임

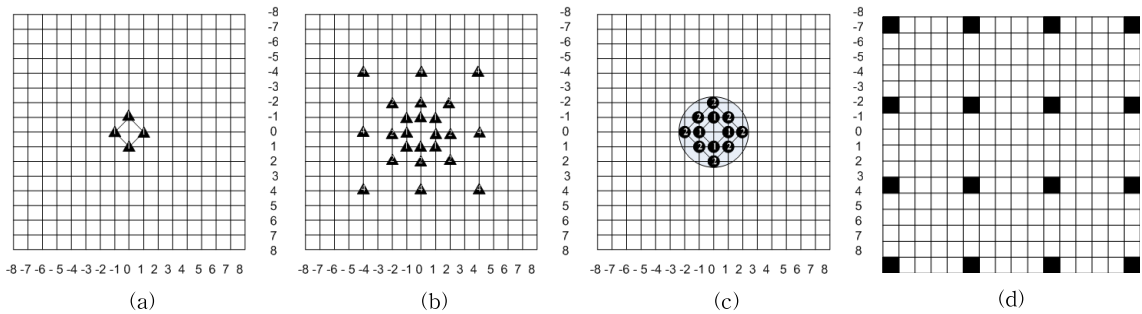


그림 2. 제안한 기법의 탐색 패턴들. (a) Multi-grid Square search pattern, (b) Small Diamond search pattern, (c) Modified Diamond search pattern, (d) Raster search.

표 1. 움직임 벡터 분포

	1 pixel 이내	2 pixel 이내	3 pixel 이내	4 pixel 이내
akiyo	98.53%	98.65	98.69	98.70
carphone	56.68%	60.40	61.62	62.31
claire	91.81%	91.97	92.04	92.12
foreman	36.90%	44.88	49.35	52.53
mother	87.83%	88.75	89.29	89.44
salesman	95.27%	95.44	95.52	95.54
stefan	40.19%	44.85	48.14	50.72
suzie	61.71%	66.27	68.45	69.57

벡터가 탐색 영역의 중심부분에 많이 분포한다는 사실을 이용하여 움직임 추정 초기에 탐색 영역의 중심부분에 탐색 점들을 배치하여 움직임 벡터를 추정함으로써 움직임이 작은 블록에서 빠르게 움직임 벡터를 추정할 수 있다. 즉 제안한 다중 사각형 탐색 패턴은 움직임 추정 초기에 탐색 영역의 원점을 중심으로 4픽셀 이내에 탐색 점들을 배치하여 움직임 벡터를 추정함으로써 움직임이 작은 블록에서 초기에 움직임 추정을 종료할 수 있고 iRaster+1인 탐색 영역 내에만 탐색 점들을 배치함으로써 중복된 탐색 점들의 수를 줄임으로 계산량을 줄였다.

움직임이 큰 영상에서는 다중 사각형 탐색 패턴을 수행한 후 래스터 탐색 패턴을 사용하여 움직임 추정을 수행하는데, 제안한 움직임 추정 기법에서 사용하는 그림 2(d)의 래스터 탐색 패턴은 기존의 래스터 탐색 패턴의 iRaster(래스터 탐색의 탐색 길이: 3)를 iRaster+2로 증가시켰다. iRaster를 5로 증가시키므로 움직임 추정 시, 적은 탐색 점들을 사용하여 움직임 벡터가 존재할 가능성이 높은 위치를

추정한다.

그리고 TZ 탐색 기법의 정교한 탐색은 uiBest-Distance가 0이 될 때까지 초기 그리드 탐색 패턴을 반복 수행하는데, 이 때 불필요한 탐색 점들을 배치하고 검사한다. 이러한 문제점을 해결하기 위하여 제안한 움직임 추정 기법은 그림 2(b)와 같은 작은 다이아몬드 탐색 패턴 또는 그림 2(c)와 같이 탐색 영역의 원점을 중심으로 반경 2화소 이내에 움직임 벡터가 분포할 확률이 약 56%(블록의 움직임이 큰 영상)~98%(블록의 움직임이 작은 영상)[7,17]이라는 사실을 이용한 수정된 다이아몬드 탐색 패턴을 사용하였다.

제안한 움직임 추정 기법은 블록 움직임의 크기에 따라 적응적으로 탐색 패턴을 사용함으로써 움직임 추정 계산량을 줄였다. 제안한 움직임 추정 기법은 그림 2(a)의 다중 사각형탐색 패턴, 그림 2(b)의 작은 다이아몬드 탐색 패턴, 그림 2(c)의 수정된 다이아몬드 탐색 패턴 그리고 그림 2(d)의 iRaster가 5인 래스터 탐색 패턴으로 구성된다. 제안한 움직임 추정기법의 알고리즘 순서도는 그림 3와 같고 다음과 같은 알고리즘을 수행하여 움직임 벡터를 추정한다.

**[단계1]** 움직임 벡터 예측(Motion Vector Prediction)

메디안 예측, 상위 모드를 이용한 예측, 이전 화면의 대응 블록을 이용한 예측, 이웃한 참조 화면을 이용한 예측들을 사용하여 초기 탐색 시작점을 결정한다.

**[단계2]** 다중 사각형 탐색 패턴(Multi-grid Square Search)

그림 2(a)의 다중 사각형 탐색 패턴을 이용하여 초기 탐색 시작점 주위에 탐색 점들을 배치한다. 탐색 점들 중 최소 비용함수 값을 가지는 점이 [단계2]의 최적 탐색점이다.

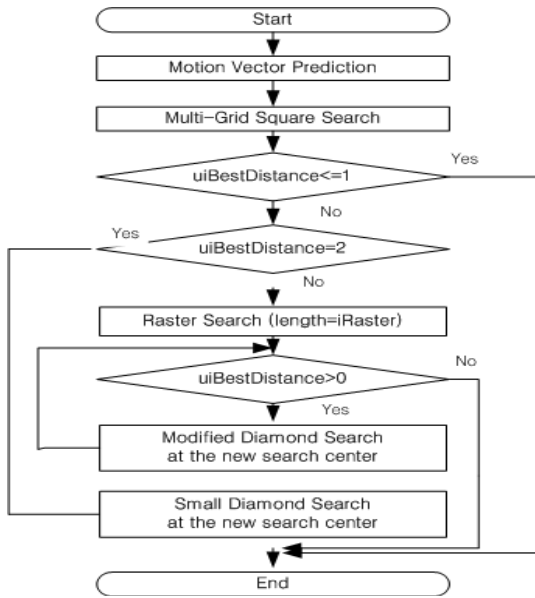


그림 3. 제안한 기법의 순서도

**[단계3]**

초기 탐색 시작점과 [단계 2]의 최적 탐색점 사이의 거리(uiBestDistance)를 구한다. uiBestDistance가 1이하인 경우 [단계2]의 최적 탐색 점을 움직임 벡터로 추정하고 움직임 추정을 마친다. 만약 uiBestDistance가 래스터 탐색의 iRaster (stride length) 보다 크면 [단계5]를 수행하고 그렇지 않으면 [단계4]를 수행한다.

**[단계4] 작은 다이아몬드 탐색(Small Diamond search)**

탐색 시작점을 변경한 후 작은 다이아몬드 탐색을

수행한다. 탐색 점들 중 최소 비용함수 값을 가지는 점을 움직임 벡터로 추정하고 움직임 추정을 마친다.

**[단계5] 래스터 탐색(Raster search)**

iRaster의 크기를 iRaster+2로 변경한 후, 그림 2(d)와 같은 방법으로 탐색 영역 내에 탐색 점들을 배치하여 래스터 탐색을 수행한다. 탐색 점들 중 최소 비용함수 값을 가지는 점을 [단계5]의 최적 탐색점이라 한다.

**[단계6] 수정된 다이아몬드 탐색**

탐색 시작점을 변경한 후 수정된 다이아몬드 탐색을 수행한다. 탐색 점들 중 최소 비용함수 값을 가지는 점이 [단계6]의 최적 탐색점이다. 그리고 변경된 탐색 시작점과 [단계 6]의 최적 탐색점 사이의 거리(uiBestDistance)를 구한다. uiBestDistance가 0이면 [단계6]의 최적 탐색점 움직임 벡터로 추정하고 움직임 추정을 마친다. 그렇지 않으면 [단계6]를 최대 2회까지 반복 수행한다.

**4. 실험 결과**

제안하는 예측 구조의 성능을 확인하기 위하여 제안하는 예측 구조를 JMVM 6.0에서 구현하였다. 실험 영상으로 640×480크기의 Exit, Ballroom, Race1, Flamenco2와 1024×768크기의 Uli를 사용하였다. Exit, Ballroom, Race1, Uli는 1차원 배열의 영상이고 Flamenco2는 2차원 배열 영상이다. 실험조건은 표 2에 나타내었다.

제안한 기법, TZ 탐색 기법 그리고 TZH[16]의 평균 화질과 비트량을 표 3에 나타내었고 BD-PSNR과



그림 4. 실험 영상(위에서부터 Uli, Ballroom, Race1, Exit, Flamenco2)

표 2. 실험 조건

Sequence	Quantization Parameter (QP)	Search area	The number of frames
Exit	22, 27, 32, 37	±96	100
Ballroom	22, 27, 32, 37	±96	100
Uli	22, 27, 32, 37	±96	100
Flamenco2	22, 27, 32, 37	±96	100
Race1	22, 27, 32, 37	±96	100

표 3. 실험결과 - 화질과 비트량

Sequence	QP	TZh		TZ		Proposed method	
		Bitrates (kbps)	PSNR_Y (dB)	Bitrates (kbps)	PSNR_Y (dB)	Bitrates (kbps)	PSNR_Y (dB)
Exit	22	797.09	39.84	794.72	39.85	838.78	39.83
	27	340.29	38.28	337.60	38.30	372.25	38.24
	32	179.06	36.39	177.44	36.44	201.44	36.34
	37	107.05	34.11	105.36	34.17	122.95	34.00
Uli	22	4154.02	38.98	4148.45	38.99	4185.33	38.99
	27	2202.02	36.96	2197.35	36.97	2223.23	36.96
	32	1225.13	34.42	1215.34	34.40	1236.36	34.43
	37	675.33	31.60	667.83	31.62	683.47	31.63
Ballroom	22	1504.70	39.05	1494.35	39.04	1652.13	39.05
	27	758.73	36.80	751.27	36.82	804.72	36.79
	32	407.38	34.25	402.10	34.28	440.53	34.23
	37	230.66	31.56	226.70	31.56	254.09	31.51
Race1	22	1230.89	39.90	1204.80	39.92	1398.69	39.70
	27	603.72	37.24	582.93	37.30	705.56	36.91
	32	314.07	34.56	298.58	34.63	367.85	34.31
	37	187.63	31.81	174.83	31.95	224.82	31.30
Flamenco2	22	1773.05	41.25	1759.47	41.26	1802.53	41.25
	27	973.98	38.39	962.37	38.42	998.04	38.39
	32	528.56	35.37	521.85	35.39	544.89	35.37
	37	284.36	32.36	278.78	32.38	296.97	32.38

BD-BitRate를 표 4에 제시하였다. 제안한 기법, TZ 그리고 TZH 탐색 기법의 움직임 추정 시간을 표 5에 나타내었다. Ballroom과 Exit에서 제안한 기법은 TZ 탐색 기법보다 움직임 추정의 계산량을 약 45% 감소시킨다. 영상 화질은 각각 0.372(dB), 0.24(dB) 저하를 보이고 발생 비트량은 각각 10.02Kbps, 11.33 Kbps 증가한다. Uli에서 제안한 기법은 TZ 탐색 기법보다 영상 화질은 약 0.05(dB) 저하, 비트량은 1.25Kbps 증가하지만 움직임 추정 계산량은 약 70%

감소시킴으로 약 3.3배 빠르게 움직임 벡터를 추정할 수 있었다. Flamenco2에서 제안한 기법은 TZ 탐색 기법보다 영상 화질은 약 0.2(dB) 저하, 비트량은 4.6Kbps 증가하지만 약 2.5배 정도 빠르게 움직임 벡터를 추정한다. 그리고 Race1에서 제안한 기법은 TZ 탐색 기법보다 영상 화질은 약 1.2(dB) 저하, 비트량은 30.57Kbps 증가하지만 약 4.5배 정도 빠르게 움직임 벡터를 추정한다.

Exit, Ballroom, Flamenco2, Race1 그리고 Uli에

표 4. BDSNR과 BDBitrate

Sequence	BDPSNR(dB)		BDBitrate(kbps)	
	TZ vs. 제안기법	TZH vs. 제안기법	TZ vs. 제안기법	TZH vs. 제안기법
Exit	-0.24	-0.343	11.339	12.723
Uli	-0.052	-0.031	1.256	0.752
Ballroom	-0.372	-0.314	10.029	8.408
Flamenco2	-0.219	-0.133	4.605	2.758
Race1	-1.207	-0.986	30.570	23.001

서 제안한 기법은 움직임 추정 계산량을 50% 감소시킴으로 TZH탐색 기법보다 약 2배 빠르게 움직임 벡터를 추정한다. Exit, Ballroom, Flamenco2, Race1 그리고 Uli에서 제안한 기법은 TZH 탐색 기법보다 각각 0.3(dB), 0.3(dB), 0.1(dB), 0.9(dB), 0.03(dB) 화질 저하를 보이고 비트량은 각각 12.72Kbps, 8.4 Kbps, 2.7Kbps, 23.0Kbps, 2.7Kbps 증가한다.

Exit, Ballroom, Flamenco2, Race1 그리고 Uli 영상에서 제안한 기법은 TZ 탐색 기법과 TZH 탐색 기법보다 약 2~3.3배 빠르게 움직임 벡터를 추정하고 비슷한 영상 화질과 발생 비트량을 보인다.

QP 37일 때 Ballroom 영상에서 제안한 기법과 FS 기법의 화질, 비트량 그리고 움직임 추정 시간을 표 6에 나타내었다. 제안한 기법은 FS 기법보다 영상 화질은 약 0.13(dB) 저하, 비트량은 약 28.84Kbps 증가하지만 움직임 추정 계산량을 약 98% 감소시킴으로 약 90배 정도 빠르게 움직임 벡터를 추정한다.

5. 결론

본 논문에서는 다시점 비디오의 움직임 추정에 소요되는 계산량을 줄이면서 화질과 발생하는 비트량을 유지하기 위한 효과적인 초기 종료 움직임 추정 기법을 제안하였다. 제안한 기법은 움직임 벡터의 분포 특성을 이용한 계층적인 탐색 기법으로 다중 사각

표 5. Comparison values of total motion estimation time

Sequence	QP	TZH	Proposed method	$\Delta T$	TZ	Proposed method	$\Delta T$
		(sec)	(sec)		(sec)	(sec)	
Exit	22	82615	42676	-0.48	77122	42676	-0.45
	27	75471	38846	-0.49	69999	38846	-0.45
	32	70484	36667	-0.48	64453	36667	-0.43
	37	64855	35111	-0.46	60665	35111	-0.42
Uli	22	204897	114850	-0.44	386955	114850	-0.70
	27	188061	107416	-0.43	355760	107416	-0.70
	32	177220	101272	-0.43	324375	101272	-0.69
	37	160546	96382	-0.40	298839	96382	-0.68
Ballroom	22	88437	46297	-0.48	83309	46297	-0.44
	27	82561	42901	-0.48	78003	42901	-0.45
	32	78475	40491	-0.48	72337	40491	-0.44
	37	73980	37978	-0.49	67215	37978	-0.43
Race1	22	98219	47606	-0.52	215806	47606	-0.78
	27	89408	44019	-0.51	199540	44019	-0.78
	32	78869	40716	-0.48	176361	40716	-0.77
	37	67291	38502	-0.43	154004	38502	-0.75
Flamenco2	22	54598	27472	-0.50	69597	27472	-0.61
	27	52245	26334	-0.50	65944	26334	-0.60
	32	49548	24961	-0.50	61253	24961	-0.59
	37	47157	23518	-0.50	60091	23518	-0.61

표 6. Ballroom 실험결과 (QP:37)

Ballroom	FS(Full Search)			Proposed method		
	PSNR(dB)	BitRate (Kbps)	encoding time(s)	PSNR(dB)	BitRate (Kbps)	encoding time(s)
0 view	32.11	295.15	417191	32.05	305.51	4523
1 view	31.24	156.54	434363	31.07	188.27	4883
2 view	32.18	245.80	427006	32.12	275.06	4601
3 view	31.24	153.23	436203	31.02	188.80	4896
4 view	31.54	268.89	428528	31.46	299.80	4622
5 view	31.76	171.95	434723	31.49	206.66	4914
6 view	31.89	265.47	428527	31.83	291.74	4674
7 view	31.17	244.94	430623	31.08	276.83	4865

형 탐색 패턴, 래스터 탐색 패턴, 수정된 다이아몬드 탐색 패턴 그리고 작은 다이아몬드 탐색으로 이루어져 있다.

본 논문에서 제안한 기법의 성능을 TZ 기법, TZH 기법 그리고 FS 기법의 성능과 비교하였을 때, 영상 화질과 비트량을 비슷하지만 움직임 추정 계산량을 줄임으로서 움직임 추정 속도를 각각 약 1.7~4.5배, 약 2배 그리고 약 90배 향상시킨다.

### 참 고 문 헌

[ 1 ] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. kauff, P. Eisert, and T. Wiegand, "3D Video and Free Viewpoint Video-Technologies, Applications and MPEG Standards," *Proc. the IEEE International Conference on Multimedia and Exposition*, 2006. pp. 2161-2164.

[ 2 ] *Vision on 3D Video*, ISO/IEC JTC1/SC29/WG11 N10357, 2009.

[ 3 ] T. Koga, K. Iinuma, A. Hirano, and Y. Ishiguro, "Motion Compensated Interframe Coding for Video Conference," *Proc. the NTC81*, pp. G5.3.1-5.3.5, 1981.

[ 4 ] R. Li, B. Zeng, and M.L. Liou, "A New Three Step Search Algorithm for Block Motion Estimation," *IEEE Trans. on Circuits and System for Video Technology*, Vol. 4, No. 4, pp. 438-441, 1994.

[ 5 ] J.R. Jain and A.K. Jain, "Displacement Measurement and Its Application in Inter-

frame Image Coding," *IEEE Trans. on Communications*, Vol. 29, No. 12, pp. 1779-1808, 1981.

[ 6 ] L.M. Po and W.C. Ma, "A Novel Four Search Algorithm for Block Motion Estimation," *IEEE Transactions on Circuit and Systems for Video Technology*, Vol. 6, Issue 3, pp. 313-317, 1996.

[ 7 ] J.Y. Tham, S. Ranganath, and A.A. Kassim, "A Novel Unrestricted Center-Biased Diamond Search Algorithm for Block Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, Issue 4, pp. 369-377, 1998.

[ 8 ] S. Zhu and K.K. Ma "A New Diamond Search Algorithm for Fast Block Matching Motion," *IEEE Transaction on Image Processing*, Vol. 9, No. 2, pp. 287-290, 2000.

[ 9 ] Yuk Ying Chung, Neil W. Bergmann, "Fast Search Block Matching Motion Estimation Algorithm using FPGA," *Proc. the SPIE, Visual Communications and Image Processing*, Vol. 4067, pp. 913-921, 2000.

[10] Danian Gong and Yun He, "Fast Motion Estimation Algorithm using Horizontal and Multi-grid Search Strategy," *Picture Coding Symposium*, pp. 362-365, 2001.

[11] Prabhudev Irappa Howur and K.K. Ma, *Report on Performance of Fast Motion*



*Estimation using Motion Vector Field Adaptive Search Technique*, ISO/IEC/JCTI/SC29/WG11 M5453, 1999.

[12] Ce Zhu, Xion Lin, and Lau-Pui Chau, "Hexagon Based Search Pattern for Fast Block Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No.5, pp. 349-355, 2002.

[13] A.M. Tourapis, O.C. Au, and M.L. Liou, *Fast Block Matching Motion Estimation using Predictive Motion Vector Field Adaptive Search Technique*, ISO/IEC/JCTI/SC29/ WG11 MPEG2000/M5866, 2000.

[14] Z. Chen, P. Zhou, and Y. He, "Fast Integer Pel and Fractional Pel motion Estimation in for JVT" in JVT of ISO/ICE/ MPEG and ITU-T VCEG, 6th Meeting, 2002.

[15] Tang Xiu-li, Dai Sheng-kui, Cai Can-hui, "An Analysis of TZSearch Algorithm in JMVC," International Conference on Green Circuits and System, pp. 516-519, 2010.

[16] N. Purnachand, Luis Nero Alves, and Antonio Navarro, "Improvements to TZ Search Motion Estimation Algorithm for Multiview Video Coding," *Proc. the IWSSIP*, pp. 388-391, 2012.

[17] H.S Yoon and G.S Lee, "Adaptive Motion Estimation Method using Temporal Correlation," *The KIPS Transactions: Part B*, Vol. 9, No. 2, pp. 199-204, 2004.

[18] 윤효순, 김미영 "다시점 비디오 부호화를 위한 저 복잡도 움직임 추정 탐색 기법", 한국 멀티미디어학회논문지, Vol. 16, No. 5, pp.539-548



김 미 영

1994년 3월 ~ 1997년 8월 전남대학교 전산학과 이학 박사  
 1998년 3월 ~ 현재 전남도립대학교 보건의료과 교수  
 관심분야 : 영상처리, 영상부호화, 멀티미디어 시스템



윤 효 순

2002년 3월 ~ 2005년 2월 전남대학교 전산학과 이학 박사  
 관심분야 : 영상처리, 영상부호화, 멀티미디어 시스템