다시점 비디오 부호화를 위한 저 복잡도 움직임 추정 탐색 기법

윤효순[†], 김미영^{††}

요 약

움직임 추정은 비디오 영상 압축에 있어서 중요한 역할을 하지만 최적의 움직임 벡터를 추정하기 위해 많은 계산량을 요구한다. 다시점 비디오는 하나의 3차원 장면을 여러 시점에서 다수의 카메라로 촬영한 동영상으로 다시점 비디오 부호화를 위한 움직임 추정의 계산량은 카메라 수에 비례하여 증가한다. 본 논문에서는 다시점 비디오 부호화를 위한 움직임 추정의 계산량을 줄이면서 영상 화질을 유지하는 저 복잡도 움직임 추정 기법을 제안하였다. 제안한 기법은 계층적인 탐색 기법으로 4-그리드 다이아몬드 탐색 패턴, 2-그리드 다이아몬드 탐색 패턴 그리고 TZ 2 point 탐색 패턴으로 구성되었다. 이들 탐색 패턴들은 움직임 벡터의 분포 특성을 이용하여 탐색 점들을 배치함으로써 적은 계산량으로 움직임 벡터를 추정할 수 있다. 제안한 기법의 성능을 JMVC의 고속 움직임 추정 기법인 TZ 탐색 기법의 성능과 비교하였을 경우, 영상 화질 면에서 약 0.01~0.24(dB) 화질 저하를 보였지만 움직임 추정의 계산량을 약 42%~80% 줄임으로서 약 1.8~4.5배 속도 향상을 보였다.

Low Complexity Motion Estimation Search Method for Multi-view Video Coding

Hyo-sun Yoon[†], Mi-Young Kim^{††}

ABSTRACT

Although Motion estimation (ME) plays an important role in digital video compression, it requires a complicated search procedure to find an optimal motion vector. Multi-view video is obtained by capturing one three-dimensional scene with many cameras at different positions. The computational complexity of motion estimation for Multi-view video coding increases in proportion to the number of cameras. To reduce computational complexity and maintain the image quality, a low complexity motion estimation search method is proposed in this paper. The proposed search method consists of four-grid diamond search patten, two-grid diamond search pattern and TZ 2 Point search pattern. These search patterns exploit the characteristics of the distribution of motion vectors to place the search points. Experiment results show that the speedup improvement of the proposed method over TZ search method (JMVC) can be up to 1.8 ~4.5 times faster by reducing the computational complexity and the image quality degradation is about to 0.01~0.24 (dB)

Key words: motion estimation(다시점 비디오), video compression(비디오 압축), motion vector(움직임 추정), multi-view video coding(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 1월 8일, 수정일: 2013년 2월 14일
완료일: 2013년 2월 21일

[†] 정회원, 전남대학교 전자컴퓨터공학부
(E-mail: estheryoona@hotmail.com)

^{††} 정회원, 전남도립대학교 보건의료과

※ 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 연구임(No. 2012-0007893).

1. 서 론

최근 비약적인 정보 통신의 발달로 영상처리와 관련된 많은 기술들이 발전하면서 이를 기반으로 하는 다양한 고화질의 멀티미디어 콘텐츠를 사용자에게 제공 할 수 있게 되었다. 그리고 디지털 TV, 자유시점 TV(free viewpoint television), 3차원 TV와 같은 새로운 멀티미디어 서비스가 시작되었다.

실감 미디어 콘텐츠에 대한 수요가 증가함에 따라 다시점 비디오는 3차원 영상에 대한 다양한 사용자의 요구를 충족시킬 수 있는 대안으로 주목 받고 있다. 다시점 비디오는 여러 개의 카메라들을 사용하여 동시에 하나의 3차원 장면을 촬영한 동영상의 집합으로 사용자에게 임의의 시점을 제공하며 여러 시점의 영상을 합성하여 보다 넓은 화면을 제공할 수 있다. 그러나 다시점 비디오는 카메라의 수에 비례하여 데이터의 양이 늘어나기 때문에 다시점 비디오를 효율적으로 부호화하는 기술이 필요하다[1,2].

다시점 비디오뿐만 아니라 모든 영상들은 저장, 전송 또는 다양한 분야에서 사용되기 위하여 반드시 압축되어야 하는데, 이 때 움직임 추정 기법을 이용하여 압축한다.

영상 화질과 인코더의 속도에 영향을 주는 움직임 추정은 동영상 압축에서 중요한 역할을 하지만 전체 인코더의 계산량의 60~80%를 차지한다. 그래서 움직임 추정에 소요되는 계산량을 줄이기 위하여 다음과 같은 움직임 추정 기법들이 제안되었다.

대표적인 고속 움직임 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], 육각형 탐색기법(Hexagonal Search:HS)[11], 움직임 벡터 필드 적응적 탐색 기법(MVFAST)[12] 그리고 예측된 움직임 벡터 필드 적응적 탐색기법(PMFAST)[13]등이 있다. 고속 움직임 추정 기법[3-13]들은 상대적으로 움직임이 작은 영상이나 영상 크기가 작은 경우에 움직임 벡터 추정을 잘 수행한다. 그러나 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.

비대칭 다중 육각형 탐색 기법[14]은 계층적인 움직임 추정 기법으로 영상의 움직임이 크거나 영상 크기가 상대적으로 큰 영상에서 움직임 벡터 추정을 잘 수행하여 화질 저하를 개선하였다. 그러나 비대칭 다중 육각형 탐색 기법은 탐색 영역 내에 배치하는데 있어서 비대칭적, 불규칙적으로 배치함으로서 블록 움직임이 수직으로 크거나 블록 움직임이 크면서 규칙적인 영상에서 국부적 최소화에 빠질 수 있다.

위에서 나열한 고속 움직임 추정 기법들은 일반적으로 단일 시점 영상에서 사용되는 움직임 추정 기법들이다. 다시점 비디오를 압축하기 위하여 JMVC에서 사용되는 고속 움직임 추정 기법은 TZ 탐색 기법[15]으로 TZ 탐색 기법은 초기에 블록의 움직임에 무관하게 탐색 영역 전역에 탐색 점들을 배치한다. 이로 인하여 움직임이 작은 블록에서 불필요한 탐색 점들을 검사함으로서 움직임 벡터 추정에 많은 계산량을 필요로 한다. 그리고 움직임이 큰 블록에서 움직임 벡터 추정할 때 비슷한 탐색 영역에 탐색 점들을 배치함으로서 불필요한 계산량을 사용하는 단점을 가지고 있다.

본 논문에서는 다시점 비디오 부호화를 위한 움직임 추정의 계산량을 줄이면서 영상 화질과 발생 비트량을 유지하는 저 복잡도 움직임 추정 기법을 제안하였다. 제안한 움직임 추정 기법은 계층적인 탐색 기법으로 4-그리드 다이아몬드 탐색 패턴, 2-그리드 다이아몬드 탐색 패턴 그리고 TZ 2 point 탐색 패턴으로 구성되었다. 이들 탐색 패턴들은 움직임 벡터의 분포 특성을 이용하여 탐색 점들을 탐색 영역에 배치함으로써 적은 계산량으로 움직임 벡터를 추정할 수 있다.

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

2. TZ 탐색 기법

JMVC의 고속 움직임 추정 기법은 TZ 탐색 기법으로 TZ 탐색 기법의 알고리즘 순서도는 그림 1에

그림 1. TZ 탐색 기법의 알고리즘 순서도

나타냈다. TZ 탐색 기법에서 이용되는 탐색 패턴들은 그림 2에 나타났다. TZ 탐색 기법은 다음과 같은 알고리즘을 수행하여 움직임 벡터를 추정한다.

[단계1]: 움직임 벡터 예측

메디안 예측, 상위 모드를 이용한 예측, 이전 화면

의 대응 블록을 이용한 예측, 이웃한 참조 화면을 이용한 예측들을 사용하여 초기 탐색 시작점을 결정한다.

[단계2]: 다중 그리드 다이아몬드 탐색

초기 탐색 시작점을 탐색의 원점으로 다이아몬드 탐색 패턴을 사용한다. 다이아몬드 탐색 패턴 stride length의 범위는 1, 2, 4, 8, 16, 32, 64이다. 그림 2 (a)는 stride length가 8인 다이아몬드 탐색 패턴으로 그림 2 (a)와 같은 방법으로 탐색 영역에 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

그림 2 (b)는 raster length가 3인 래스터 탐색 패턴으로 그림 2 (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

이전 단계에서 구해진 최적 탐색 점을 탐색 시작점으로 변경한 후 [단계2]의 초기 그리드 탐색을 수행한다. 탐색 점들 중 최소 비용함수 값을 가지는 점

그림 2. TZ 탐색 기법의 탐색 패턴들

이 [단계5]의 최적 탐색점이다. 그리고 변경된 탐색 시작점과 [단계5]의 최적 탐색점 사이의 거리(ui-BestDistance)를 구한다.

uiBestDistance가 0이면 [단계5]의 최적 탐색점을 움직임 벡터로 추정하고 움직임 추정을 마친다. 만약 uiBestDistance가 1이면 2개의 점을 추가 검색 한 후, 최소 비용 함수 값을 가지는 점을 움직임 벡터로 추정하고 탐색을 마치고, 그렇지 않으면 [단계5]를 반복 수행한다.

3. 제안한 움직임 추정 기법-저 복잡도 움직임 추정 탐색 기법

일반적으로 고속 움직임 추정 기법[3-14]들에서 움직임 추정에 사용되는 탐색 패턴의 변화는 추정된 움직임 벡터의 정확도와 움직임 추정 속도에 영향을 준다. 움직임 벡터들의 분포 특성, 블록의 움직임 방향 또는 영상 크기를 고려한 탐색 패턴들은 고려하지 않는 패턴들 보다 영상 화질과 움직임 벡터 추정에 소요되는 계산 량에서 좋은 성능을 보였다.

예를 들면, 4단계 탐색(FSS), 다이아몬드 탐색(DS) 중 움직임 벡터의 분포 특성을 고려한 DS와 고려하지 않는 FSS의 움직임 벡터 추정 결과를 표 1과 표 2에 나타냈다. 실험 영상으로 Akiyo, Claire, Carphone, Foreman, Mother and Daughter (M & D), Salesman and Stefan과 같은 단일 영상들을 사용하였다.

표 1에 각 영상들의 PSNR을 나타냈고 표 2에 각 영상에서 움직임 벡터 추정 시, 검사하는 평균 탐색 점 수를 나타내었다. 표 1과 표 2에서 움직임 벡터의 분포 특성을 고려한 DS가 FSS에 비해 움직임 벡터 추정에 사용하는 탐색 점들 수는 적으면서 정확한 움직임 벡터를 추정하였다. DS와 FSS의 성능을 비

표 2. 영상의 평균 탐색점 수

영 상	FS	FSS	DS
Akiyo	961	17	13.02
Claire	961	17.08	13.1
Carphone	961	17.8	14.4
Foreman	961	18.6	15.4
M & D	961	17.1	13.2
Salesman	961	17.0	13
Stefan	961	18.9	16.2

교하였을 경우, DS가 정확한 움직임 벡터를 추정함으로써 영상 화질이 좋을 뿐만 아니라 움직임 벡터 추정에 소요되는 계산량을 줄였다. 즉, 움직임 벡터 추정에 사용되는 탐색 패턴의 변화는 추정된 움직임 벡터의 정확도와 움직임 벡터 추정 속도에 영향을 준다. 그래서 본 논문에서는 TZ 탐색 기법에서 사용하는 탐색 패턴들을 변화시킴으로써 영상 화질은 유지하면서 움직임 추정 계산량을 줄이는 기법을 제안하였다.

TZ 탐색 기법은 초기에 블록의 움직임 크기에 무관하게 탐색 영역 전역에 탐색 점들을 배치한다. 이로 인하여 움직임이 작은 블록에서 불필요한 탐색 점들을 검사함으로써 움직임 벡터 추정에 많은 계산량을 필요로 한다. 즉, TZ 탐색 기법의 초기에 사용하는 다중 그리드 다이아몬드 탐색 패턴은 블록의 움직임 크기에 무관하게 탐색 영역 전역에 탐색 점들을 배치하므로 블록의 움직임이 작을 때 불필요한 탐색 점들을 조사함으로써 계산량을 증가시킨다.

그리고 움직임이 큰 블록에서 움직임 벡터 추정할 때 비슷한 탐색 영역에 탐색 점들을 배치함으로써 불필요한 계산량을 사용하는 단점을 가지고 있다. 즉, TZ 탐색 기법은 블록의 움직임이 큰 경우 다중 그리드 다이아몬드 탐색 후 래스터 탐색을 사용하는 데, 래스터 탐색 역시 탐색 영역 전역에 탐색 점들을 수직, 수평 방향으로 배치하여 최적의 탐색 점을 찾는다. 이것은 다중 그리드 다이아몬드 탐색에서 iRaster(래스터 탐색의 탐색 길이:3) 보다 큰 탐색 영역에 있는 탐색 점들을 검사할 필요가 없다는 것을 의미한다.

그래서 제안한 기법에서는 TZ 탐색 기법의 불필요한 계산량을 줄이기 위하여 움직임 벡터의 분포 특성을 이용한 탐색 패턴들을 제안하였다. 제안한 기법의 초기 탐색 패턴인 그림 3 (a)의 4-그리드 다이

표 1. 영상의 PSNR

영 상	FS	FSS	DS
Akiyo	34.5	34.33	34.39
Claire	35.05	34.74	34.85
Carphone	30.88	30.15	30.48
Foreman	29.54	28.22	28.66
M & D	31.52	31.34	31.42
Salesman	32.7	32.53	32.62
Stefan	23.88	22.62	22.77

아몬드 탐색 패턴과 그림 3 (b)의 2-그리드 다이아몬드 탐색 패턴은 탐색 영역의 원점을 중심으로 반경 2화소 이내에 움직임 벡터가 분포할 확률이 약 56% (블록의 움직임이 큰 영상)~98%(블록의 움직임이 작은 영상)[8,16]이라는 사실을 이용한 탐색 패턴들이다. 특히, 4-그리드 다이아몬드 탐색 패턴은 블록의 움직임이 큰 영상에서의 보다 정확한 움직임 벡터 추정 확률을 높이기 위해 탐색 영역의 원점을 중심으로 반경 4화소 이내에 탐색 점들을 배치하였다.

TZ 탐색 기법의 정교한 탐색은 $uiBestDistance$ 가 0이 될 때까지 다중 그리드 다이아몬드 탐색 패턴을 반복 수행하는데, 이 때 많은 탐색 점들을 조사함으로써 계산량을 증가시킨다. 이러한 문제점을 해결하기

위하여, 제안한 고속 움직임 추정 기법은 그림 3(b)의 2-그리드 다이아몬드 탐색 패턴을 사용한다. 그리고 2-그리드 다이아몬드 탐색에서 $uiBestDistance$ 가 0이면 움직임 벡터 추정을 마치고 $uiBestDistance$ 가 1이면 그림 3(c)의 TZ 2 point 탐색 패턴을 사용하여 두 개의 점을 추가하여 조사한 후 최적의 움직임 벡터를 추정한다. 예를 들면 TZ 2 point 탐색 패턴은 2-그리드 다이아몬드 탐색 패턴의 탐색 점들 중 최소 비용함수 값을 가지는 점의 위치가 탐색 시작점을 중심으로 (0,-1)이면 (-1,-2)와 (1,-2) 위치에 있는 두 개의 점을 추가하고, (-1,0)이면 (-2,-1)와 (-2,1) 위치에 두 개의 점을 추가하고 (0,1)이면 (-1,2)와 (1,2) 위치에 있는 두 개의 점을 추가하고 (1,0)이면 (2,-1)

그림 3. 제안한 움직임 추정 기법의 탐색 패턴들

와 (2,1)에 추가한다.

그림 3 (a)의 4-그리드 다이아몬드 탐색 패턴, 그림 3 (b)의 2-그리드 다이아몬드 탐색 패턴, 그리고 그림 3 (c)의 TZ 2 point 탐색 패턴으로 구성된 제안한 움직임 추정 기법의 알고리즘 순서도는 그림 4에 제시되었고 다음과 같은 알고리즘을 수행하여 움직임 벡터를 추정한다.

[단계1]: 움직임 벡터 예측

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

[단계2]: 4-그리드 다이아몬드 탐색

그림 3(a)의 탐색 패턴을 이용하여 초기 탐색 시작점 주위에 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 1이면 [단계5]를 수행하고 그렇지 않으면 [단계4]를 수행한다.

[단계4]: 2-그리드 다이아몬드 탐색

탐색 시작점을 변경한 후 2-그리드 다이아몬드 탐색을 수행한다. 탐색 점들 중 최소 비용함수 값을 가지는 점이 [단계4]의 최적 탐색점이다. 그리고 변경된 탐색 시작점과 [단계 4]의 최적 탐색점 사이의 거리(uiBestDistance)를 구한다. uiBestDistance가 0이면 초기 탐색 시작점을 움직임 벡터로 추정하고 움직임 추정을 마친다. 만약 uiBestDistance가 1이면 [단계5]를 수행하고 그렇지 않으면 [단계4]를 반복 수행한다.

[단계5]: TZ 2 point 탐색

2-그리드 다이아몬드 탐색 패턴의 탐색 점들 중 최소 비용함수 값을 가지는 점의 위치가 탐색 시작점을 중심으로 (0,-1)이면 (-1,-2)와 (1,-2) 위치에 있는 두 개의 점을 추가하고, (-1,0)이면 (-2,-1)와 (-2,1) 위치에 두 개의 점을 추가하고 (0,1)이면 (-1,2)와 (1,2) 위치에 있는 두 개의 점을 추가하고 그리고 (1,0)이면 (2,-1)와 (2,1)에 추가한다. 이 탐색 점들 중 최소 비용함수 값을 가지는 점을 움직임 벡터로 추정하고 움직임 추정을 마친다.

그림 4. 제안 기법의 알고리즘 순서도

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 Uli,를 사용하였다. 실험 영상들은 그림 5에 나타내었다. Exit, Ballroom, Race1, Uli는 1차원 배열의 영상이고 Flamenco2는 2차원 배열 영상이다. 실험조건은 표 3에 나타내었다.

일반적으로 동영상 압축 분야에서 기술 비교의 핵심은 “같은 용량일 때 원본 영상으로부터 손실을 얼마나 줄이느냐? 또는 같은 화질에서 용량을 얼마나 줄이느냐?”이기 때문에 압축 용량 대비 화질을 평가 척도로 사용한다. 그래서 일반적으로 4개 정도의 지점을 적당히 골라 용량 대비 화질을 RD curve로 비교함으로써 성능의 우위를 판단한다. 실험 영상들의 RD curve는 그림 6에 나타내었다. RD curve에서 두 곡선이 교차하거나 일치하게 되면 어떤 방법의 성능이 좋은지가 애매해 질 수 있는데, 이 때 수치상으로

그림 5. 실험 영상(위에서부터 Uli, Ballroom, Race1, Exit, Flamenco2)

표 3. 실험 조건

영상	기본 QP	탐색범위	프레임 수
Exit	22, 27, 32, 37	± 96	100
Ballroom			
Uli			
Race1			
Flamenco2			

성능을 비교하기 위해 BD-PSNR과 BD-BitRate가 사용된다. 제안 기법과 TZ 탐색 기법의 BD-PSNR과 BD-BitRate는 표 4에 제시하였고 그리고 움직임 추정 시간은 표 5에 나타내었다.

실험 영상의 RD curve에서 Exit, Uli, Ballroom, Flamenco2의 경우 TZ 탐색 기법과 제안 기법의 곡선은 거의 일치한다. Exit과 Ballroom에서 제안 기법은 TZ 탐색 기법보다 움직임 추정 계산량은 약 45% 감소시킴으로 약 1.5배 빠르게 움직임 벡터를 추정하였지만 영상 화질은 각각 0.098(dB), 0.24(dB) 저하를 보였고 발생 비트량도 각각 5.632Kbps, 6.2190 Kbps 증가시켰다. Uli에서 제안 기법은 TZ 탐색 기법보다 영상 화질 면에서 약 0.037(dB) 저하, 발생 비트량에서 0.894Kbps 증가했지만 움직임 추정 계산량을 약 70% 감소시킴으로 약 2.5배 빠르게 움직임 벡터를 추정할 수 있었다. Flamenco2에서 제안

그림 6. 실험 영상의 RD curves (QP: 22, 27, 32, 37)

표 4. BD-PSNR와 BD-BitRate

영상	QP	TZ 탐색 기법		제안 기법		BDPSNR	BDBitrate
Exit	22	794.72	39.85	824.30	39.83	-0.098	5.632
	27	337.60	38.03	356.81	38.25		
	32	177.44	36.44	190.27	36.36		
	37	105.36	34.17	115.81	34.06		
영상	QP	TZ 탐색 기법		제안 기법		BDPSNR	BDBitrate
Uli	22	4148.454	38.986	4170.989	38.984	-0.037	0.894
	27	2197.355	36.968	2210.951	36.957		
	32	1215.340	34.397	1230.358	34.424		
	37	667.828	31.619	679.667	31.613		

영상	QP	TZ 탐색 기법		제안 기법		BDPSNR	BDBitrate
Ballroom	22	1494.352	39.041	1541.684	39.045	-0.2405	6.2190
	27	751.270	36.815	784.892	36.791		
	32	402.099	34.278	426.587	34.232		
	37	226.697	31.559	244.496	31.517		
영상	QP	TZ 탐색 기법		제안 기법		BDPSNR	BDBitrate
Race1	22	1204.804	39.924	1277.280	39.769	-0.5158	12.9688
	27	582.925	37.296	617.218	37.032		
	32	298.576	34.628	316.434	34.353		
	37	174.825	31.946	185.308	31.468		
영상	QP	TZ 탐색 기법		제안 기법		BDPSNR	BDBitrate
Flamenco2	22	1759.468	41.259	1771.337	41.253	-0.0958	1.9987
	27	962.367	38.417	975.926	38.402		
	32	521.854	35.394	531.525	35.378		
	37	278.780	32.375	287.001	32.354		

표 5. 움직임 추정시간 비교

영상	QP	TZ 탐색기법	제안 기법	ΔT
Exit	22	77122	42935	-0.44
	27	69999	38901	-0.44
	32	64453	36849	-0.43
	37	60665	35182	-0.42
영상	QP	TZ 탐색기법	제안 기법	ΔT
Uli	22	386955	116024	-0.70
	27	355760	108137	-0.70
	32	324375	102968	-0.68
	37	298839	97360	-0.67
영상	QP	TZ 탐색기법	제안 기법	ΔT
Ballroom	22	83309	46057	-0.45
	27	78003	43085	-0.45
	32	72337	40205	-0.44
	37	67215	38197	-0.43
영상	QP	TZ 탐색기법	제안 기법	ΔT
Race1	22	215806	47961	-0.78
	27	199540	44608	-0.78
	32	176361	41285	-0.77
	37	154004	38647	-0.75
영상	QP	TZ 탐색기법	제안 기법	ΔT
Flamenco2	22	69597	27462	-0.61
	27	65944	26267	-0.60
	32	61253	24788	-0.60
	37	61291	23630	-0.61

기법은 TZ 탐색 기법보다 영상 화질 면에서 0.0958 (dB) 저하, 발생 비트량에서 1.9987Kbps 증가하였지만 움직임 추정 계산량을 약 60% 감소시킴으로 약 2.5배 정도 빠르게 움직임 벡터를 추정하였다. Race1 에서 제안 기법은 TZ 탐색 기법보다 영상 화질 면에서 0.5185(dB) 저하, 발생 비트량은 12.998Kbps 증가하였지만 움직임 추정 계산량을 약 78% 감소시킴으로 약 4.5배 이상 빠르게 움직임 벡터를 추정할 수 있었다.

5. 결 론

본 논문에서는 다시점 비디오 부호화의 움직임 추정에 소요되는 계산량을 줄이면서 화질과 발생되는 비트량을 유지하기 위한 저 복잡도 움직임 추정 탐색 기법을 제안하였다. 제안한 움직임 추정 기법은 계층적인 탐색 기법으로 4-그리드 다이아몬드 탐색 패턴, 2-그리드 다이아몬드 탐색 패턴 그리고 TZ 2 point 탐색 패턴들로 이루어져있다. 이 탐색 패턴들은 움직임 벡터의 분포 특성을 이용하여 탐색 점들을 배치함으로써 움직임 추정에 소요되는 계산량을 줄였다.

본 논문에서 제안한 기법의 성능을 TZ 탐색 기법의 성능과 비교하였을 때 약 0.01~0.24(dB) 화질 저하를 보였지만 움직임 벡터 추정 속도를 약 1.8~4.5 배 향상시켰다.

참 고 문 헌

“멀티미디어학회논문지가 참고문헌상에 없습니다. 본 학회논문지와 학술적 관련성을 위하여 1편이상을 참고문헌에 포함하시고 논문에서 언급하시기 바랍니다.”

- [1] ISO/IEC JTC1/SC29/WG11 N10357, *Vision on 3D Video*, 2009.
- [2] 윤효순, 김미영, “다시점 비디오 부호화를 위한 시간적 예측 구조,” *한국멀티미디어학회논문지*, Vol. 15, No. 9, pp. 1093- 1101. 2012. (논문지명 기입 요함!)
- [3] T.Koga, K.Iinuma, A.Hirano, and Y.Ishiguro, “Motion Compensated Interframe Coding for

- Video Conference,” *Proc. 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 Transactions 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 Interframe Image Coding,” *IEEE Transactions 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, 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, 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 and Neil W.Bergmann, “Fast Search Block Matching Motion Estimation Algorithm using FPGA,” *Proc. 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] 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, pp. 349-355, 2002.
- [12] ISO/IEC/JCTI/SC29/ WG11, “Report on

Performance of Fast Motion Estimation using Motion Vector Field Adaptive Search Technique,” *M5453*, 1999. (표준 및 연구보고서는 본 학회의 규정형식에 따라 수정요함!!!)

- [13] *ISO/IEC/JCTI/SC29/ WG11 MPEG2000*, “Fast Block Matching Motion Estimation using Predictive Motion Vector Field Adaptive Search Technique,” *M5866*, 2000. (표준 및 연구보고서는 본 학회의 규정형식에 따라 수정요함!!!)
- [14] *ISO/ICE/ MPEG and ITU-T VCEG*, “Fast Integer Pel and Fractional Pel Motion Estimation in for JVT,” *6th meeting*, 2002. (표준 및 연구보고서는 본 학회의 규정형식에 따라 수정요함!!!)
- [15] *ISO/IEC MPEG, ITU-T VCEG, MVC Software Reference Manual-Joint Multiview Video Model (JMVM) 8.2*, 2010. (표준 및 연구보고서는 본 학회의 규정형식에 따라 수정요함!!!)
- [16] TANG Xiu-lil, DAI Sheng-kui, and CAI Can-hui, “An Analysis of TZSearch Algorithm in JMVC,” *International Conference on Green Circuits and Systems*. pp. 516-520, 2010. (학회명, 페이지 기입 요함!)

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