

움직임 벡터 분포 특성과 블록 움직임의 특성을 이용한 대칭형 움직임 추정 기법

A Symmetric Motion Estimation Method by using the Properties of the Distribution of Motion Vectors

윤효순*, 김미영**

전남대학교 전자컴퓨터공학부*, 전남도립대학교 보건의료학과**

Hyo-Sun Yoon(estheryoonyoon@hotmail.com)*, Mi-Young Kim(kimmee@dorip.ac.kr)**

요약

비디오 압축에서 움직임 추정 기법은 영상 화질과 발생 비트량에 있어서 중요한 역할을 하지만 많은 계산 복잡도를 요구한다. 다수의 카메라로 촬영한 동영상인 다시점 비디오는 카메라의 수에 비례하여 데이터의 양이 기하급수적으로 증가하기 때문에 움직임 추정에 많은 계산량을 필요하다. 본 논문에서 다시점 비디오의 부호화를 위한 움직임 추정의 계산량을 줄이면서 화질과 비트량을 유지하는 움직임 추정 기법을 제안한다. 제안한 움직임 추정 기법은 움직임 벡터들의 분포 특성과 영상 블록들의 움직임 특성을 이용한다. 제안한 움직임 추정 기법은 계층적 움직임 추정 기법으로 대칭형 멀티 마름모 패턴, 대각선 패턴, 사각형 패턴 그리고 정교한 패턴으로 구성되어 있다. 제안한 움직임 추정 기법은 움직임 벡터들과 블록 움직임의 특성들을 이용하여 패턴들의 탐색 점들을 탐색 영역 내에 대칭적으로 배치하고 블록 움직임 크기에 따라 적응적으로 탐색 패턴을 선택하여 움직임 벡터를 추정한다. 제안한 기법의 성능은 JMVC의 고속 움직임 추정 기법인 TZ 탐색 기법과 전역 탐색 기법인 PBS (Pel Block Search)의 성능과 비교한 경우, 영상 화질 면에서와 발생 비트량면에서 비슷하지만 움직임 추정에 필요한 계산량을 각각 약 40~75%, 98% 감소시킨다.

■ 중심어 : | 움직임 추정 | 다시점 비디오 | 다시점 비디오 부호화 | 움직임 벡터 |

Abstract

In video compression, Motion Estimation(ME) limits the performance of image quality and generated bit rates. However, it requires much complexity in the encoder part. Multi-view video uses many cameras at different positions. Multi-view video coding needs huge computational complexity in proportion to the number of the cameras. To reduce computational complexity and maintain the image quality, an effective motion estimation method is proposed in this paper. The proposed method exploiting the characteristics of motion vector distribution and the motion of video. The proposed is a kind of a hierarchical search strategy. This strategy consists of multi-grid rhombus pattern, diagonal pattern, rectangle pattern, and refinement pattern. Experiment results show that the complexity reduction of the proposed method over TZ search method and PBS (Pel Block Search) on JMVC (Joint Multiview Video Coding) can be up to 40~75% and 98% respectively while maintaining similar video image quality and generated bit rates.

■ keyword : | Motion Estimation | Multi-view Video | Multi-view Video Coding | Motion Vector |

*This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No.2010-0024120)

접수일자 : 2016년 11월 15일

심사완료일 : 2017년 02월 14일

수정일자 : 2017년 02월 02일

교신저자 : 김미영, e-mail : kimmee@dorip.ac.kr

I. 서론

멀티미디어 시스템과 네트워크 통신의 발달로 멀티미디어 콘텐츠에 대한 수요가 급증하고 있고 다양한 멀티미디어 콘텐츠를 사용자에게 제공할 수 있게 되었다. 실감 미디어에 대한 사용자의 요구가 증가함에 따라 3차원 비디오의 중요성이 커지고 있다[1].

3차원 영상에 대한 다양한 사용자의 요구를 충족시킬 수 있는 대안으로 주목받고 있는 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 기하급수적으로 증가하므로 다시점 비디오를 효율적으로 부호화하는 기술이 필요하다[2].

일반적으로 동영상 비디오 부호화는 영상들 사이에 존재하는 공간적, 시간적, 시점간의 중복된 데이터를 제거함으로써 데이터를 압축한다. 이러한 중복된 데이터들을 제거하기 위하여 움직임 추정을 수행한다.

영상 화질과 발생 비트량 그리고 인코더의 속도에 영향을 주는 움직임 추정은 동영상 압축에서 중요한 역할을 하지만 전체 인코더에서 많은 계산량을 요구한다. 움직임 추정을 수행하기 위하여 일반적으로 블록 정합 기법을 사용하는데, 블록 정합 기법에는 전역 탐색 기법과 전역 탐색 기법의 계산량을 줄이기 위해 제안된 많은 고속 움직임 추정 기법들이 있다. 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 벡터 필드 적응적 탐색 기법(MVFAST)[11], 육단계 탐색 기법(Hexagonal Search:HS)[12], 예측된 움직임 벡터 필드 적응적 탐색 기법(PMVFAST)[13] 그리고 비대칭 다중 육각형 탐색 기법[14] 등이 있다.

위에서 나열한 기법[3-14]들은 일반적으로 단일 시점

영상에서 사용되는 움직임 추정 기법으로, [3-13]의 기법들 영상 크기가 작거나 규칙적인 영상에서 사용한 기법이고, [14]는 영상 크기가 크거나 불규칙한 영상에서 사용되는 계층적 움직임 추정 기법이다.

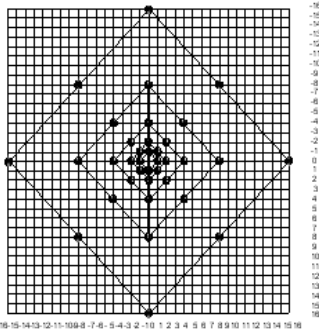
다시점 비디오를 압축할 때 사용되는 고속 움직임 추정 기법에는 TZ 기법[15] 그리고 TZH 기법[16] 등이 있다. TZ 기법은 움직임 추정 초기에 탐색 영역 전역에 탐색 점들을 배치하는데, 움직임이 작은 블록에서 불필요한 탐색 점들을 검사함으로써 많은 계산량을 요구한다. TZH 기법은 TZ 기법의 계산량을 줄이기 위하여 TZ 기법에서 사용하고 있는 다이아몬드 그리드 패턴 대신 육각형 그리드 탐색 패턴을 이용하였다.

본 논문에서는 다시점 비디오 부호화를 위한 움직임 추정의 계산량을 줄이면서 영상 화질을 유지하는 움직임 추정 기법을 제안한다. 제안한 움직임 추정 기법은 움직임 벡터들의 분포 특성과 블록의 움직임 특성을 이용한 계층적 움직임 추정 기법으로 멀티 그리드 마름모 패턴, 대각선 패턴, 사각형 패턴 그리고 정교한 패턴으로 이루어져 있다. 멀티 그리드 마름모 패턴은 움직임 벡터가 탐색 영역의 중심 부분에 많이 분포하는 사실과 블록의 움직임이 수평, 수직 방향으로 많이 움직인다는 특성을 이용하여 탐색 점들을 탐색 영역에 배치한다 [17]. 영상 움직임이 크거나 규칙적인 영상을 위하여 사각형 패턴을 사용한다. 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.

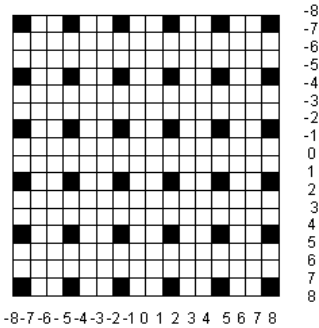
본 논문의 구성은 다음과 같다. II에서는 JMVC의 고속 움직임 추정 기법인 TZ 기법에 대하여 설명하고, III에서는 본 논문에서 제안한 움직임 추정 기법을 기술한다. 그리고 IV에서는 TZ 기법, TZH 기법 그리고 FS 기법과 성능을 비교한 후, 마지막으로 V에서는 결론을 맺는다.

II. TZ 기법

JMVC의 고속 움직임 추정 기법은 TZ 기법은 PBS의 계산량을 줄이기 위하여 사용되는 기법으로 TZ 기법에서 사용되는 탐색 패턴들은 [그림 1]과 같다. 그리고 TZ 기법은 다음과 같은 알고리즘을 수행하여 움직임 벡터를 추정한다.



(a) Initial grid search pattern (stride length :16)



(b) Raster search pattern (raster length :3)

그림 1. TZ기법의 탐색 패턴들

[단계1] : 움직임 벡터 예측 - 현재 블록의 탐색 시작점을 결정하기 위하여 현재 블록과 상관성을 가진 블록들의 움직임 벡터들을 이용한다. 즉, 메디안 예측, 상위 모드를 이용한 예측, 이전 화면의 대응 블록을 이용한 예측, 이웃한 참조 화면을 이용한 예측들을 사용하여 현재블록의 탐색 시작점을 결정한다.

[단계2] : 초기 그리드 탐색 (Initial Grid search)

단계1의 탐색 시작점을 탐색의 원점으로 초기 그리드 탐색 패턴을 사용하여 탐색 점들을 탐색 영역에 배치한다.

다. 초기 그리드 탐색 패턴에서 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 같이 탐색 영역에 탐색 점들을 배치하고 각 점에서의 비용함수를 계산한다. 이들 탐색 점들 중 최소 비용함수 값을 가지는 점이 [단계2]의 최적 탐색점이다.

[단계3] : [단계 1]의 초기 탐색 시작점과 [단계 2]의 최적 탐색점 사이의 거리를 구하여 uiBestDistance에 저장한다. 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 전역에 탐색 점들을 배치하고, 탐색 점들 중 최소 비용 함수 값을 가지는 점을 [단계4]의 최적 탐색 점으로 결정한다.

[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 거리를 구한다. uiBestDistance가 0이면 [단계5]의 최적 탐색 점을 움직임 벡터로 추정하고 움직임 추정을 마친다. 그렇지 않으면 [단계5]를 반복 수행한다.

III. 제안한 예측 구조

TZ 기법의 초기 그리드 탐색은 탐색 영역 전역에 탐색 점들을 배치하고 비용 함수를 계산하여 최적의 탐색 점을 찾은 후, 래스터 탐색을 수행한다. 래스터 탐색 또한 탐색 영역 전역에 탐색 점들을 배치하여 최적의 탐색 점을 찾는다. 초기 그리드 탐색과 래스터 탐색 모두 탐색 영역 전역에 탐색 점들을 배치하므로 초기 그리드 탐색에서 iRaster(래스터 탐색의 탐색 길이:3) 보다 큰 탐색 영역에 있는 탐색 점들을 검사할 필요가 없다는

것을 의미한다.

본 논문에서 제안한 [그림 2(a)]의 멀티-그리드 마름모 패턴의 초기 탐색 점들 배치는 탐색 영역의 원점을 중심으로 4픽셀 이내에 탐색 점들을 배치하여 움직임 벡터를 추정하는데 이는 다음과 같은 실험 결과를 이용하였다. 여러 영상들의 움직임 벡터들을 분석한 결과, 움직임 벡터가 탐색 영역의 원점을 중심으로 반경 1화소이내, 2화소이내, 3화소 이내 그리고 4화소 이내에 분포 확률을 계산한 결과는 [표 1]와 같다[17]. 멀티-그리드 마름모 패턴의 탐색 점들의 배치는 [표 1]에서와 같이 움직임 벡터가 탐색 영역의 중심부분에 많이 분포한다는 사실과 영상의 움직임이 대각선 방향 보다는 수평, 수직 방향으로 더 많이 움직임을 보인다는 사실들을 이용하여 움직임 추정 초기에 탐색 영역의 중심 부분에 수평, 수직 방향으로 탐색 점들을 배치함으로써 움직임이 작은 블록이나영상의 움직임이 규칙적인 블록에서 빠르게 움직임 벡터를 추정하여 초기에 움직임을 추정할 수 있다.

표 1. 움직임 벡터 분포

영상	1 pixel	2 pixel	3 pixel	4 pixel
Akiyo	98.53%	98.65%	98.69%	98.70%
Carphone	56.68%	60.40%	61.62%	62.31%
Claire	91.81%	91.97%	92.40%	92.12%
Foreman	30.69%	44.88%	49.35%	52.53%
Mother	87.83%	88.75%	89.29%	89.44%
Salesman	95.72%	95.44%	95.52%	95.54%
Suzie	61.72%	66.27%	68.45%	69.57%

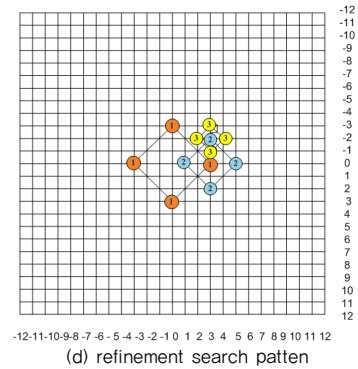
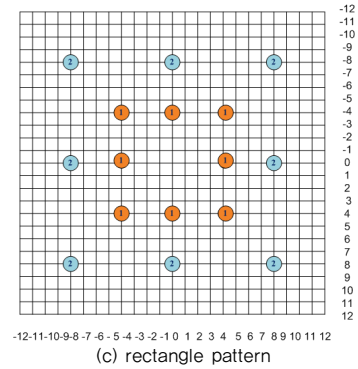
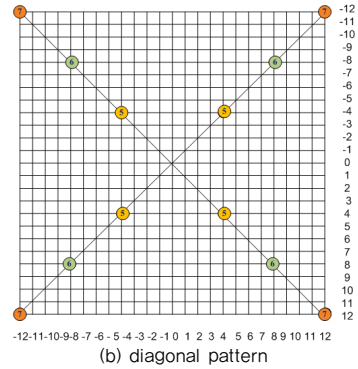
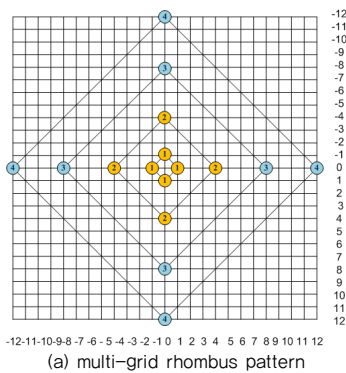


그림 2. 제안 기법의 탐색 패턴들

멀티-그리드 마름모 탐색을 수행한 후, 블록의 움직임이 작거나 수평, 수직 방향이 아닌 경우 [그림 2(b)]의 대각선 탐색 패턴을 이용하여 움직임 벡터를 수행한다. 대각선 탐색 후, 블록의 움직임이 큰 경우 [그림 2(c)]의 사각형 패턴을 이용하여 탐색 영역에 탐색 점들을 규칙적, 대칭적으로 배치한다.

그리고 TZ 탐색 기법의 정교한 탐색은 초기 그리드 탐색 패턴을 이용하여 $uiBestDistance$ 가 0이 될 때까지 반복 수행하는데, 이 때 불필요한 탐색 점들을 배치하고 검사한다. 이러한 문제점을 해결하기 위하여 제안한 움직임 추정 기법에서의 정교한 패턴은 [그림 2(d)]와 같은 탐색 패턴을 이용한다.

제안한 움직임 추정 기법은 블록 움직임의 크기에 따라 적응적으로 탐색 패턴을 사용함으로써 움직임 추정 계산량을 줄인다. 제안한 움직임 추정 기법은 [그림 2(a)]의 멀티-그리드 마름모 패턴, [그림 2(b)]의 대각선 패턴, [그림 2(c)]의 사각형 탐색 패턴 그리고 [그림 2(d)]의 정교한 탐색 패턴으로 구성된다. 제안한 움직임 추정 기법은 다음과 같은 알고리즘을 수행하여 움직임 벡터를 추정한다.

[단계1] 움직임 벡터 예측 - 현재 블록의 탐색의 시작점을 결정하기 위하여 메디안 예측, 상위 모드를 이용한 예측, 이전 화면의 대응 블록을 이용한 예측, 이웃한 참조 화면을 이용한 예측들을 사용하여 탐색 시작점을 결정한다.

[단계2] 멀티-그리드 마름모 패턴

[그림 2(a)]의 멀티-그리드 마름모 패턴을 이용하여 초기 탐색 시작점을 중심으로 멀티-그리드 마름모 패턴의 ①,②에 탐색 점들을 배치한다. 탐색 점들 중 최소 비용함수 값을 가지는 점을 [단계2]의 최적 탐색점으로 결정한다. 만약, 최소 비용 함수를 가지는 점이 ①이면 최소 비용 함수를 가지는 점을 움직임 벡터로 추정하고 탐색을 마친다. 그렇지 않으면 멀티-그리드 마름모 패턴의 ③④에 탐색 점들을 배치하고 조사한다. 만약, 최소 비용 함수를 가진 점이 ②와 ③이면, [단계 6]을 수행한다. 최소 비용 함수를 가진 점이 ④이면 [단계 3]을 수행한다.

[단계3] 대각선 패턴

초기 탐색 시작점을 중심으로 [그림 2(b)]의 탐색 패턴에 있는 탐색 점들을 배치하여 조사한다. 만약, 최소 비용 함수를 가진 점이 ④, ⑤, ⑥ 중의 하나이면 [단계 6]을 수행한다. 그렇지 않으면 [단계 4]를 수행한다.

[단계4] 사각형 패턴

탐색 시작점을 이전 단계의 최소 비용함수 값을 가지

는 점으로 변경하고 이 점을 중심으로 [그림 2(c)]의 탐색 패턴의 ①, ②에 탐색 점들을 배치한다. 그리고 최소 비용 함수를 가진 점을 결정한다. 만약, 최소 비용 함수를 가진 점이 ①이면 [단계 5]를 수행하고, 그렇지 않으면 이 단계를 반복 수행한다.

[단계5] 탐색 시작점을 변경한 후, 변경된 탐색 시작점을 중심으로 [그림 2 (d)]정교한 탐색 패턴의 ①과 같이 탐색 점들을 배치하고, 탐색 점들 중 최소 비용함수 값을 가지는 점을 결정한다.

[단계6] 정교한 탐색 - 탐색 시작점을 변경한 후, 변경된 탐색 시작점을 중심으로 정교한 탐색 패턴의 ②과 같이 탐색 점들을 배치하고 각 탐색 점들에서의 비용함수를 계산한다. 최소 비용 함수를 가진 점을 탐색 시작점으로 변경한다. 변경된 탐색 시작점을 중심으로 정교한 탐색 패턴의 ③과 같이 탐색 점들을 배치하고 각 탐색 점들에서의 비용함수를 계산하고 최소 비용 함수를 가진 점을 움직임 벡터로 추정하고 탐색을 마친다.

IV. 실험결과



(Uli, Ballroom, Race1, Exit, Flamenco2)

그림 3. 실험영상

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 2]에 나타내었다.

표 2. 실험 조건

sequence	Quantization Parameter (QP)	search area	the number of frames
Exit	22, 27, 32, 37	±96	100
Ballroom			
Uli			
Race1			
Flamenco2			

제안한 기법과 TZ 기법의 화질과 비트량, BD-PSNR 그리고 BD-BitRate를 [표 3]에 제시하였다. 제안한 기법과 TZ의 움직임 추정 시간을 [표 4]에 나타내었다. Uli에서 제안한 기법은 TZ 기법보다 영상 화질은 약 0.01(dB) 저하, 비트량은 0.09Kbps 증가하지만 움직임 추정 계산량은 약 73% 감소시켰다. Uli는 영상 움직임이 거의 없는 영상으로 초기에 움직임 추정을 종료함으로써 움직임 추정 계산량을 73% 감소하였으며 화질과 발생 비트량은 TZ 기법의 성능과 유사하다. Exit과 Ballroom에서 제안한 기법은 TZ 기법보다 움직임 추정의 계산량을 약 48% 감소시킨다. 영상 화질은 각각 0.17(dB), 0.18(dB) 저하를 보이고 발생 비트량은 각각 6.3Kbps, 4.6Kbps 증가한다. Exit과 Ballroom은 영상의 전경 부분과 배경 부분의 움직임의 크기가 다르므로 블록 움직임이 작은 블록에서는 TZ 기법보다 빠르게 움직임 벡터를 추정한 반면에 블록 움직임이 큰 블록에서는 TZ기법과 유사한 계산량을 필요로 한다. Race1에서 영상 화질은 0.86(dB)저하, 발생 비트량은 각각 22.93Kbps 증가하였지만 움직임 추정 계산량을 약 70%감소시켰다. Race1는 영상의 움직임이 빠른 영상으로 움직임 벡터 추정 시 국부적 최소화에 빠져 화질 저하를 가져왔다. Flamenco2에서 영상 화질은 0.10(dB) 저하, 발생 비트량은 2.03Kbps 증가하였지만 움직임 추정 계산량을 각각 약 63% 감소시켰다. Flamenco2는 움직임 벡터 추정을 초기에 종료함으로써 움직임 추정 계산량을 감소하였다.

표 3. 실험 결과

- 화질과 비트량 그리고 BDPSNR과 BDBitrate

Sequence	QP	TZ		Proposed method		BDPSNR	BDBitrate
		(kpbs)	(dB)	(kpbs)	(dB)		
Exit	22	794.72	39.85	814.52	39.83	-0.17(dB)	6.32(kbps)
	27	337.60	38.03	357.21	38.02		
	32	177.44	36.44	187.78	36.40		
	37	105.36	34.17	111.21	34.14		
Uli	22	4,148.45	38.99	4,168.72	38.99	-0.01(dB)	0.09(kbps)
	27	2,197.35	36.97	2,216.53	36.96		
	32	1,215.34	34.40	1,210.27	34.42		
	37	667.83	31.62	675.48	31.62		
Ballroom	22	1,494.35	39.04	1,523.16	39.05	-0.18(dB)	4.62(kbps)
	27	751.27	36.82	779.09	36.81		
	32	402.10	34.28	421.68	34.25		
	37	226.70	31.56	240.23	31.54		
Race1	22	1,204.80	39.92	1,368.59	39.78	-0.86(dB)	22.93(kbps)
	27	582.93	37.30	687.13	37.15		
	32	298.58	34.63	352.94	34.41		
	37	174.83	31.95	210.35	31.70		
Flamenco2	22	1,759.47	41.26	1,768.28	41.24	-0.1(dB)	2.03(kbps)
	27	962.37	38.42	973.63	38.40		
	32	521.85	35.39	537.31	35.38		
	37	278.78	32.38	281.16	32.38		

표 4. 실험 결과- 움직임 추정시간 비교

영상	QP	TZ	제안기법	ΔT (%)
		(sec)	(sec)	
Exit	22	77122	40103	-0.48
	27	69999	36399	-0.48
	32	64453	34160	-0.47
	37	60665	32759	-0.46
Uli	22	386955	104478	-0.73
	27	355760	96055	-0.73
	32	324375	90825	-0.72
	37	298839	86663	-0.71
Ballroom	22	83309	44154	-0.47
	27	78003	41342	-0.47
	32	72337	38339	-0.47
	37	67215	36296	-0.46
Race1	22	215806	43161	-0.80
	27	199540	39908	-0.80
	32	176361	37036	-0.79
	37	154004	32341	-0.79
Flamenco2	22	69597	25751	-0.63
	27	65944	24399	-0.63
	32	61253	22664	-0.63
	37	60091	22234	-0.63

일반적으로 영상의 화질이 저하가 되면 발생 비트량은 증가하게 되고 화질이 향상되면 발생 비트량은 감소하는데 화질과 발생 비트량은 trade-off 관계이다. 이는 움직임 추정시 보다 정확한 움직임 벡터를 찾으면 현재 블록과 참조 블록 사이의 유사도가 증가하므로 현재 블록과 참조 블록사이의 차분 영상이 작아짐에 따라 차분 영상 부호화시 발생하는 비트량이 줄어든다.

표 5. Ballroom 실험결과 (QP: 37)

Ballroom	FS (Full Search)		
	PSNR(dB)	BitRate (Kbps)	encoding time(s)
0 view	32.11	295.15	417191
1 view	31.24	156.54	434363
2 view	32.18	245.80	427006
3 view	31.24	153.23	436203
4 view	31.54	268.89	428528
5 view	31.76	171.95	434723
6 view	31.89	265.47	428527
7 view	31.17	244.94	430623

Ballroom	제안기법		
	PSNR(dB)	BitRate (Kbps)	encoding time(s)
0 view	32.03	309.04	4497
1 view	31.10	195.62	4853
2 view	32.21	308.59	4573
3 view	31.00	200.23	4874
4 view	31.49	319.22	4623
5 view	31.50	219.05	4862
6 view	31.88	326.29	4660
7 view	31.02	297.26	4810

QP 37일 때 Ballroom 영상에서 제안한 기법과 FS 기법의 화질, 비트량 그리고 움직임 추정 시간을 [표 5]에 나타내었다. 제안한 기법은 FS 기법보다 영상 화질은 약 0.12(dB) 저하, 비트량은 약 40Kbps 증가하지만 움직임 추정 계산량을 약 98.9% 감소시킴으로 빠르게 움직임 벡터를 추정한다.

V. 결론

본 논문에서는 다시점 비디오의 움직임 추정에 소요되는 계산량을 줄이면서 화질과 발생하는 비트량을 유지하기 위한 움직임 추정 기법을 제안하였다. 제안한

기법은 움직임 벡터의 분포 특성을 이용한 계층적인 탐색 기법으로 다중 사각형 탐색 패턴, 래스터 탐색 패턴, 수정된 다이아몬드 탐색 패턴 그리고 작은 다이아몬드 탐색으로 이루어져있다.

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.

참고 문헌

- [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," IEEE International Conference on Multimedia and Exposition, 2006(7).
- [2] ISO/IEC JTC1/SC29/WG11 N10357, "Vision on 3D Video," 2009(2).
- [3] T. Koga, K. Iinuma, A. Hirano, and Y. Ishiguro, "Motion compensated interframe coding for video conference," 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(8).
- [5] J. R. Jain and A. K. Jain, "Displacement measurement and its application in Interframe 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, 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," Visual Communications and Image Processing 2000, Proc.SPIE, Vol.4067, pp.913-921.

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

[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).

[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, 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 MPEG 2000/M5866, 2000(3).

[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-lil, OAT Sheng-kui, and CAT Can-hui, "An Analysis of TZSearch Algorithm

in JMVC," pp.516-519, 2010.

[16] N. Purnachand, Luis Nero Alves, and Antonio Navarro, "Improvements to TZ search motion estimation algorithm for multiview video coding," IWSSIP 2012, 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.

저 자 소 개

윤 효 순(Hyo-Sun Yoon)

정희원



- 2005년 : 전남대학교 전산학과 (박사)
- 2005년 ~ 현재 : 전남대학교 전자컴퓨터공학부

<관심분야> : 영상처리, 영상 압축, 영상 부호화

김 미 영(Mi-Young Kim)

정희원



- 1994년 : 전남대학교 전산학과 (박사)
- 1998년 ~ 현재 : 전남도립대학교 보건의료과 교수

<관심분야> : 영상처리, 영상 압축, 영상 부호화