

논문 2009-46SD-2-12

고속 부화소 움직임 추정을 위한 중심 지향적 십자 다이아몬드 탐색 알고리즘

(A Center Biased Cross-Diamond Search Algorithm for Fast
Fractional-pel Motion Estimation)

조성현*, 이종화**

(Seong Hyeon Jo and Jong Hwa Lee)

요약

본 논문에서는 H.264/AVC 표준 중에서 움직임 추정기의 고속 움직임 예측을 위한 알고리즘을 제안하였다. 움직임 추정은 H.264의 비디오 코딩 과정에서 가장 많은 연산량을 차지하는 중요한 처리과정이다. 정화소 단위에서의 탐색에 비해서 부화소 단위까지의 움직임 추정은 영상의 압축률은 높일 수 있지만 계산의 복잡도가 늘어나는 문제점이 있다. 제안하는 알고리즘은 정화소 움직임 벡터가 최종 움직임 벡터인 경우에 부화소 탐색을 생략하게 되며 그렇지 않은 경우 CBCDS (Center Biased Cross-Diamond Search)를 적용하여 부화소 움직임 탐색을 하여 기존의 전역 탐색 알고리즘에 비해 비교적 화질에 변화가 없고 탐색 지점이 81.4% 감소하는 결과를 보였다

Abstract

In general video coding systems, motion estimation (ME) is regarded as a vital component in a video coder as it consumes a large amount of computation resources. Fractional pixel motion estimation can improve the video compression rate at the cost of higher computational complexity. It is based on the experimental results that the sum of absolute differences (SAD) shows parabolic shape and thus can be approximated by using interpolation technique. In this paper, we propose a fast fractional pixel search algorithm by combining SASR (Simplified Adaptive Search Range) and the CBCDS (Center Biased Cross-Diamond Search) pattern with the predicted motion vector. Compare with the fractional pel full search and the CBFPS, the proposed CBCDS algorithms can reduce fractional pel search points up to 81.4%, respectively with the PSNR lost about 0.05dB.

Keywords : H.264, Fractional-pel search, Center Biased Cross Diamond Search(CBCDS)

I. 서론

디지털 영상신호를 포함한 멀티미디어 데이터를 실제 응용분야에 적용하기 위해서는 먼저 주어진 데이터

를 효율적으로 압축하는 일이 중요하다. H.264/AVC는 최근 여러 가지 멀티미디어 서비스 및 DMB 규격에 H.264/AVC 코덱이 포함됨으로써 산업계 및 시장의 수요가 급증하고 있다.^[1] H.264/AVC에서는 프레임의 상관관계를 이용하여, 프레임 사이의 시간적 중복성을 제거하기 위해서 움직임 예측 (Motion Estimation)과 움직임 보상 (Motion Compensation)을 사용한다. 이 과정은 동영상 압축에서 일반적으로 연산량의 60% 이상을 차지하고 있는 중요한 처리 과정이다.^[2]

* 학생회원, ** 정회원, 울산대학교 전기전자정보시스템 공학부

(School of Electrical Engineering, University of Ulsan)

※ 이 논문 또는 저서는 2007년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2007-511-D00195).

접수일자: 2008년9월25일, 수정완료일: 2009년2월2일

동영상에서 물체의 움직임은 샘플링에 사용되는 격자 단위의 정수배 단위로만 일어나지는 않기 때문에 정수

격자 단위에서의 움직임 벡터가 실제의 움직임 벡터가 아닐 수도 있다. 이로 인해 H.264/AVC에서의 움직임 추정은 정화소 단위에서 움직임 추정을 수행하여 블록 매칭 오차가 최소가 되는 지점을 중심으로 부화소 움직임 추정을 수행하여 최소 블록 매칭 오차 지점을 찾는다. H.264/AVC의 JM (Joint Model) 에서는 정화소 움직임 예측을 위해서는 UMHexagonS (unsymmetrical multi-hexagon search)가 사용되며, 부화소 움직임 예측을 위해서는 CBFPS (Center Biased Fractional Pel Search)가 사용된다. 부화소 움직임 추정은 1/4화소 지점까지 탐색을 수행하면 정화소 움직임 추정만을 수행하였을 때보다 영상의 화질이 PSNR 기준으로 1~3dB 가량의 향상을 나타낸다. 그렇지만 부화소 움직임 추정은 부화소 보간 및 탐색 연산으로 인해 연산량이 크게 증가하게 된다.

일반적으로 정화소 움직임 추정 연산이 부화소 움직임 추정연산에 비해서 상대적으로 넓은 탐색 영역을 가지다 보니 연산량 역시 많아지게 된다. 따라서 대부분의 연구가 정화소 움직임 추정 연산에 집중이 되어 이루어져 왔으며 여러 가지 고속 탐색 알고리즘들^[3~5]이 제안되고 개발되었다. 그러나 부화소 움직임 추정 연산은 탐색 영역이 넓지 않다보니 연구가 이루어지지 않아 아직까지도 전역 탐색 (Full Search)이 가장 많이 사용되어지고 그 결과로써 정화소 움직임 추정 연산은 넓은 탐색 영역을 가지지만 연산량은 부화소 움직임 추정 연산보다 적어지게 되는 결과를 낳게 되었다. 상대적으로 정화소 움직임 추정 연산의 연산량이 감소하게 되면서 부화소 움직임 추정 연산의 연산량이 전체 부호화기 내에서 차지하는 부분이 커지면서 최근 부화소 고속 탐색 알고리즘이 연구되어 지고 있다.

본 논문은 부화소 움직임 추정 연산에서의 고속 탐색 알고리즘을 제안하고자 한다. 그러기 위해서 먼저 부화소 움직임 탐색의 특징을 정확히 알아야 한다. 우선 정화소 움직임 벡터가 최종 움직임 벡터가 되는 경우가 많기 때문에 항상 부화소 움직임 추정 연산이 필요한 것은 아니다. 그리고 부화소 움직임 추정에서는 정화소 움직임 추정에서와 같이 주변 블록에서의 움직임 벡터와 유사한 방향성을 가지게 되는 특성을 가지므로 정화소 움직임 벡터와 유사한 과정을 통해 예측 부화소 움직임 벡터를 구할 수 있다.

본 논문의 구성은 II장에서는 부화소 움직임 추정 연산의 특징과 기존의 알고리즘을 살펴보고 III장에서는

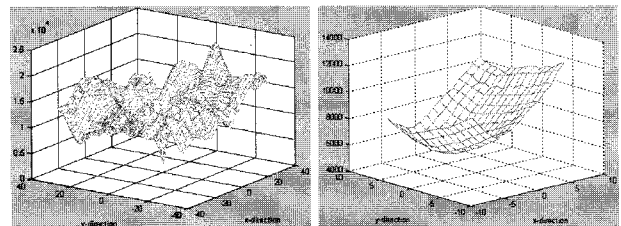
새로운 고속 부화소 움직임 추정 알고리즘을 제안한다. 그리고 IV장에서는 제안한 알고리즘과 기존의 알고리즘과의 비교를 통하여 제안한 알고리즘의 성능을 비교하게 되며 마지막으로 V장에서는 결론을 맺는다.

II. 부화소 움직임 추정 특성과 기존 부화소 움직임 추정 알고리즘

1. 정화소 움직임 벡터와 부화소 움직임 벡터

여기에 내용을 입력하세요.정화소 움직임 추정 연산과 부화소 움직임 추정 연산 사이에는 약간의 있다.

우선 위의 그림 1 (a)에서 보는 것과 같이 정화소 단위에서의 에러 표면을 보게 되면 하나의 Global minimum 뿐만 아니라 많은 Local minimum으로 인하여 픽셀간의 블록 오차값이 거리에 선형적으로 비례한다고 할 수가 없다. 그리하여 정화소 움직임 추정 연산에서는 Local minimum이 중요한 문제점이 되었었다. 하지만 그림 1 (b)에서 보는 것과 같이 부화소 단위에서의 에러 표면에는 하나의 Global minimum을 중심으로 하여 거리에 비례하여 오차가 비례하여 증가하는 것



(a) 정화소 에러 표면 (b) 부화소 에러 표면

그림 1. 움직임 추정의 에러 표면
Fig. 1. Error Surface of Motion Estimation.

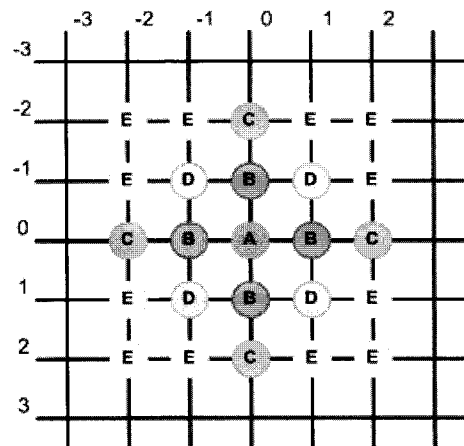


그림 2. 정화소 MV의 분포
Fig. 2. Distrubution of Integer-pel MV.

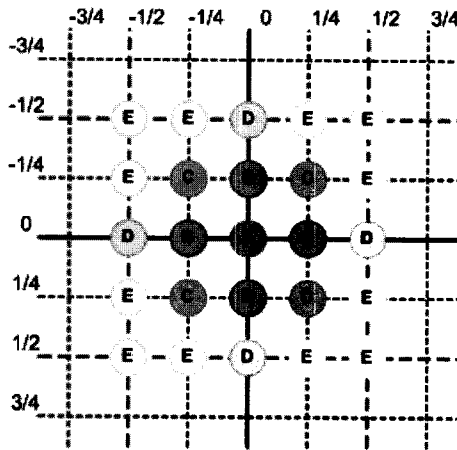


그림 3. 부화소 MV의 분포
Fig. 3. Distribution of Fractional-pel MV.

을 볼 수가 있는데 이것은 부화소 단위의 값들이 6-tap filter와 선형 보간법을 통해서 화소값들을 구하기 때문이라고 할 수 있다. 그림 2와 3은 정화소와 부화소 단위에서의 움직임 벡터 분포를 나타내는데 정화소 단위에서는 ± 1 픽셀의 대각선에 위치한 지점보다 ± 2 픽셀의 수직과 수평 방향에 움직임 벡터가 존재할 확률이 상대적으로 더 높았다. 그러나 그림 3의 부화소 단위에서의 움직임 벡터의 분포를 보면 ± 1 픽셀 이내의 지점이 ± 2 픽셀이내의 점보다는 움직임 벡터가 존재할 확률이 상대적으로 높아지게 된다. 이러한 점을 바탕으로 하여 부화소 움직임 추정 연산 알고리즘을 제안하게 되면 보다 효과적이 될 것이다.^[6]

2. 부화소 움직임 추정의 특성

부화소 움직임 추정 연산은 정화소 움직임 추정 연산만을 진행하였을 때보다 좋은 화질의 영상을 얻을 수 있다는 장점으로 인해 초기 멀티미디어 압축 표준에서부터 적용된 기술이다. 그렇지만 부화소 보간 연산 및 탐색 연산으로 인해 부호화기의 연산량이 증가하는 단점이 있으며, 또한 부화소 움직임 추정이 정화소 움직임 추정보다 항상 더 정확한 움직임을 찾을 수 있다고 말할 수 없다.

표 1에서 볼 수 있는 것과 같이 정화소 움직임 벡터가 최종 움직임 벡터가 될 확률이 Coastguard와 Foreman 같이 움직임이 빠른 경우에는 상대적으로 낮게 나타나지만 그 이외의 영상 시퀀스에서는 70% 이상이 정화소 움직임 벡터가 최종 움직임 벡터가 되어 부화소 움직임 추정 연산이 불필요한 경우가 존재한다. 그러므로 이러한 경우에 대해 탐색이 이루어지기 전에

표 1. 정화소 MV와 부화소 MV비율
Table 1. Ratio of between integer and fractional MV.

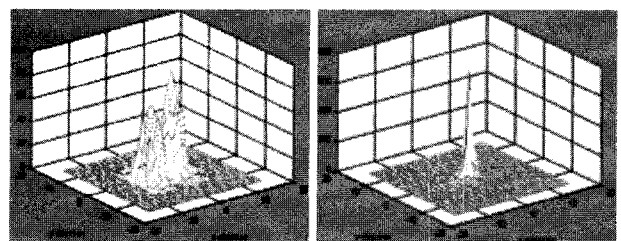
| Sequence | Integer-pel MV | Fractional-pel MV |
|------------|----------------|-------------------|
| Akiyo | 87.26 % | 12.74 % |
| Coastguard | 25.63 % | 74.37 % |
| Container | 74.02 % | 25.98 % |
| Foreman | 26.45 % | 73.55 % |
| News | 80.01 % | 19.99 % |
| Silent | 72.08 % | 27.92 % |

예측할 수 있다면 상당히 많은 연산과정을 생략할 수가 있게 되어 부화소 움직임 추정 연산과 부호화기 전체 연산에도 큰 영향을 미치게 될 것이다.

3. 부화소 움직임 벡터 예측

많은 고속 탐색 알고리즘이 초기 탐색 지점으로 예측 움직임 벡터(predicted motion vector)를 사용하고 있다. 예측 움직임 벡터는 현재 블록이 주변 블록들과 유사한 방향성을 가지고 움직일 것이라는 가정 하에 주변 블록의 움직임 벡터를 이용해 초기 탐색 지점을 예측하는 것이다. 그림 4는 예측 움직임 벡터를 사용했을 때와 사용하지 않았을 때의 탐색 중심에서부터 움직임 벡터 분포도를 보여준다.^[7~8]

그림 4 (a)는 예측 움직임 벡터를 사용하지 않고 참조 프레임의 대응 블록 위치를 탐색 중심으로 설정하고 움직임 추정을 하였을 때의 움직임 벡터의 분포도이고, 그림 4 (b)는 예측 움직임 벡터를 중심으로 설정하였을 때의 움직임 벡터 분포도이다. 그림에서 보듯이 움직임 벡터를 탐색 중심점으로 설정하였을 경우 거의 대부분의 움직임 벡터가 탐색 중심점으로 설정하였을 경우 거의 대부분의 움직임 벡터가 탐색 중심에 집중되어 있는 것을 확인할 수 있다. 따라서 초기 탐색 지점으로 예측 움직임 벡터를 이용하는 것이 탐색 연산 감소에 효과적이라는 것을 알 수 있다.



(a) (0, 0) 중심 분포 (b) MVp 중심 분포

그림 4. 움직임 벡터 분포도
Fig. 4. Distribution of Motion Vector.

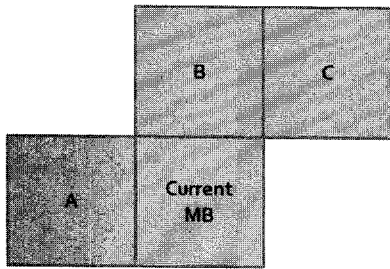


그림 5. 움직임 벡터 예측을 위한 참조 블록들
Fig. 5. Reference blocks for predicted motion vector.

움직임 벡터의 예측은 현재 블록 크기 및 인접 블록의 움직임 벡터 사용가능 여부에 따라 결정되며, 예측 방법으로는 미디언 예측기 (median predictor) 와 상위 블록 모드 예측기 (up-layer predictor) 그리고 이전 프레임의 대응 블록을 이용한 예측기 (corresponding-block predictor) 등이 있다.

$$pred_mv = median(mv_A, mv_B, mv_C) \quad (1)$$

식 (1) 에서 mv_A, mv_B, mv_C는 그림 5의 A, B, C 블록의 움직임 추정 후 결정된 움직임 벡터이고, pred_mv는 미디언 필터를 통해 예측된 움직임 벡터이다.

예측된 움직임 벡터 pred_mv 에는 정화소 예측된 움직임 벡터와 부화소 예측된 움직임 벡터가 같이 포함되어 있다. 따라서 부화소 탐색을 위한 부화소 움직임 벡터 예측값을 구하기 위해서 식 (2) 을 이용한다.

$$pred_frac_mv = (pred_mv - mv) \% \beta \quad (2)$$

식 (2) 에서 pred_frac_mv는 예측된 부화소 움직임 벡터이며, pred_mv는 미디언 예측기와 상위 블록 예측기에서 얻은 예측 움직임 벡터 중에서 블록 매칭 오차 값이 작은 예측점이다. 그리고 % 연산자는 나머지 연산자이며, β 는 부화소 단위로 MPEG -1, 2는 1/2 화소 수준까지 움직임 탐색을 수행하므로 4가 된다. 따라서 예측된 움직임 벡터를 부화소 단위인 4로 나누어 주게 되면 예측 부화소 움직임 벡터를 얻게 된다.

이렇게 예측 부화소 움직임 벡터를 바탕으로 하여 부화소 움직임 추정 연산을 수행하게 되면 ± 1 픽셀 내에서 대략 90% 이상 ± 2 픽셀 내에서는 대략 98% 이상의 부화소 움직임 벡터를 찾게 된다. 그러므로 부화소 움직임 추정 연산에 사용되어 지는 탐색 패턴의 경우에는 ± 1 픽셀 이내에서 지점을 보다 빠르게 찾을 수 있는 탐색 패턴이 보다 적합하다고 하겠다. 그러나 ± 2 픽셀을 벗어나서 부화소 움직임 벡터가 존재하는 경우는 예측 부

화소 움직임 벡터가 올바르게 선택되지 못했을 확률이 높다 이런 경우는 주로 움직임이 빠른 경우이다. 이러한 경우는 상대적으로 확률이 낮으므로 확률이 높은 경우에 적합한 탐색 방법을 택하는 것이 필요하다.

III. 제안하는 고속 부화소 움직임 추정 알고리즘

II 장의 부화소 움직임 추정의 특성을 바탕으로 고속 부화소 탐색 알고리즘을 제안한다. 제안한 고속 부화소 움직임 추정 알고리즘은 SASR (Simplified Adaptive Search Range) 값에 따라 블록의 부화소 움직임 추정 연산을 수행하지 않고 생략할 수 있다. 이와 같은 선택적 부화소 움직임 추정 연산은 효과적으로 연산량을 감소시킬 수 있다. 또한 추가적인 부화소 움직임 추정기의 연산량 감소를 위해 예측된 부화소 움직임 벡터를 탐색 중심으로 한 MSDSP (Mixed Small Diamond Search Pattern) 을 제안하였다. 그림 6은 제안한 고속 부화소 움직임 추정 알고리즘의 전체 연산 흐름도이다.

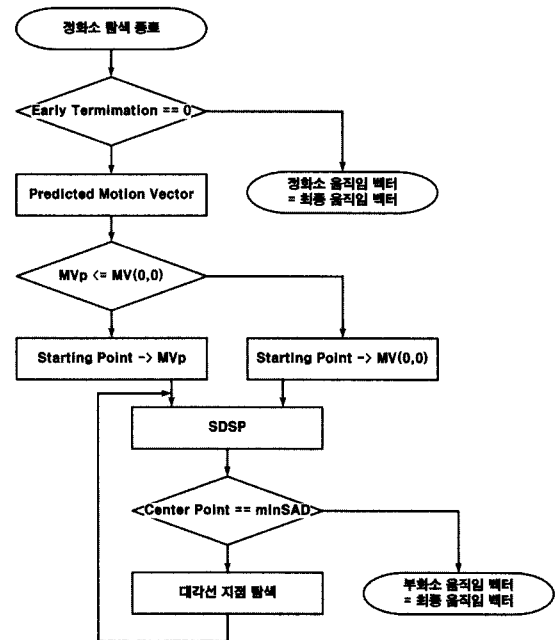


그림 6. 고속 부화소 움직임 추정 연산 흐름도
Fig. 6. Fast Fractional-pel ME Flow.

1. 부화소 탐색 생략 알고리즘

표 1을 보면 대부분 영상에서 부화소 움직임 벡터보다 정화소 움직임 벡터가 더 많이 선택되는 것을 알 수 있다. 움직임 벡터로 정화소 움직임 벡터가 선택될 경우에는 부화소 움직임 추정 연산 결과가 의미가 없어지

게 되므로 이와 같은 경우에는 부화소 움직임 추정 연산이 생략 가능하게 된다. 특히 이전 프레임의 블록과 현재 프레임 블록 사이에 움직임이 없다면 보간 연산 및 탐색 연산이 불필요하므로 움직임 추정 연산을 생략할 수 있다. 실제로 대부분의 움직임 벡터가 탐색 중심점에 집중되어 있으므로^[5], 이전 프레임 블록과 비교하여 움직임이 있는 블록은 부화소 움직임 추정을 수행하며, 움직임이 없는 블록은 부화소 움직임 추정을 생략한다.

$$SASR_{n,q} = \left[\frac{SAD_{n,q;0,0}}{Q} \times \frac{1}{4} \right] \quad (3)$$

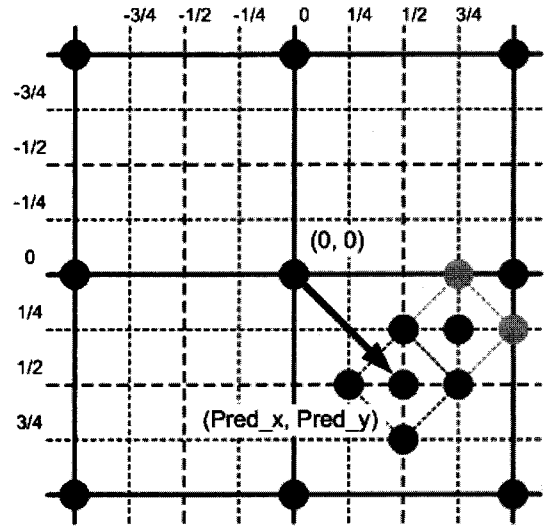
식 (3)에서 SAD_{n,q} 는 정화소 탐색 중심점의 블록 매칭 에러 값이며 Q는 블록 모드에 따라서 가변적으로 변경되며, 4 x 4일 경우는 16을 사용하며, 16 x 16일 경우는 256이 된다. 부화소 움직임 추정은 제안한 SASR의 값에 따라서 연산하지 생략할지가 결정된다. 또한 SASR의 값을 계산하기 위한 부가적인 연산도 단지 쉬프트 연산으로 가능하므로 추가적인 연산에 대한 부담이 매우 적다고 할 수 있다.

2. Center Biased Cross-Diamond Search

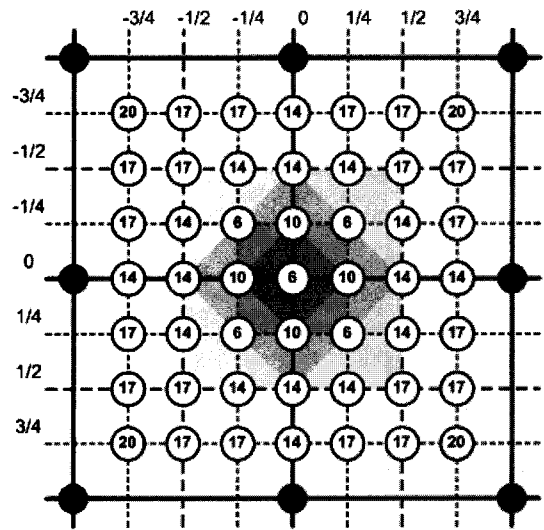
제안한 고속 부화소 움직임 추정 알고리즘은 초기 탐색 지점으로 예측 움직임 벡터를 사용한다. 예측 움직임 벡터 주변에 움직임 벡터가 있을 확률이 상당히 높으므로 예측 움직임 벡터를 중심으로 탐색하면 초기에 움직임 벡터를 찾아 추정 연산을 종료할 수 있으므로, 고속 탐색 알고리즘에 적합하다.

상위 블록 모드 예측기와 미디언 예측기를 이용하여 초기 탐색 지점을 선택한다. 부화소 움직임 추정은 정화소 탐색에서 얻은 정화소 주변을 탐색하는 것이므로 예측 부화소 움직임 벡터 역시 탐색하는 범위 안을 가리키고 있어야 한다. 만약 탐색 범위 밖을 가리키게 된다면 예측 움직임 벡터는 무의미하게 된다.

CBFPS 알고리즘에서 사용되어지는 작은 다이아몬드 탐색 패턴 (Small Diamond Search Pattern) 은 각 탐색 단계마다 중심과 상하좌우 총 5지점을 탐색한다. 만약 블록 매칭 오차가 최소가 되는 지점이 중심이면 탐색을 멈추고 그렇지 않을 경우는 다음 단계로 탐색을 진행하며 중심이 최소점이 될 때까지 탐색을 반복한다. 작은 다이아몬드 탐색 패턴은 일반적으로 적은 탐색 지점으로 우수한 탐색 성능을 가지고 있다.



(a) CBCDS 움직임 추정 패턴



(b) 각 픽셀의 최소 탐색점의 수

그림 7. CBCDS 탐색 알고리즘

Fig. 7. CBCDS Algorithm.

여러 가지 부화소 움직임 벡터의 특성을 고려하여 그림 7과 같은 CBCDS (Center Biased Cross-Diamond Search) 알고리즘을 제안하였다. 이 알고리즘은 기존의 JM에 표준화된 CBFPS (Center Biased Fractional pel search) 알고리즘을 개선한 알고리즘으로 탐색에 사용되는 탐색 패턴을 Cross search 와 Diamond search 패턴 모두를 사용하면서 대각선 지점의 탐색을 보다 빠르게 하는 탐색 패턴을 적용한 알고리즘이다.

IV. Simulation 및 설계 결과

본 논문에서 제안한 고속 부화소 탐색 알고리즘을 검

증하기 위해 Akiyo, News, Silent, Coastguard, Foreman 과 Container 총 6개의 영상 시퀀스가 사용되었으며, 움직임이 적은 영상과 움직임이 많은 영상으로 구성되어있다. 영상 시퀀스는 QCIF (Quarter Common Intermediate Format) 로 한 화면에 176 x 144 의 화소로 구성되어있다. 움직임 탐색은 탐색 영역[-16, +16] 범위 안에서 이루어졌으며, 참조 영상의 개수는 1장으로 설정하였다. 영상에서 첫 번째 프레임은 제외한 모든 프레임에서 움직임 추정을 수행하였다.

제안한 고속 부화소 탐색 알고리즘의 성능 비교를 위해서 표 2와 같이 Fractional Full search 와 CBFPS 에 대해 비교하였다. 그 결과 제안한 알고리즘은 FS와 비교하여 PSNR 차이는 최대 0.05dB 탐색 지점의 수는 81.4% 이상의 감소를 CBFPS와 비교하여서는 PSNR은 0.01dB 탐색 지점의 수는 73.2% 이상의 감소를 나타내었다.

표 2. 기존 고속 부화소 탐색 알고리즘과의 성능 비교

Table 2. Comparisons with existing fast fractional-pel search.

| Sequence | | Fraction-pel ME Algorithms | | |
|------------|--------|----------------------------|-------|----------|
| | | FS | CBFPS | Proposed |
| Akiyo | PSNR | - | 0 | -0.01 |
| | Points | 16 | 10.48 | 1.53 |
| Coastguard | PSNR | - | 0.01 | 0.02 |
| | Points | 16 | 10.86 | 2.91 |
| Container | PSNR | - | 0.01 | 0.02 |
| | Points | 16 | 10.36 | 1.55 |
| Foreman | PSNR | - | -0.04 | -0.05 |
| | Points | 16 | 11.04 | 2.97 |
| News | PSNR | - | 0.03 | 0.03 |
| | Points | 16 | 10.58 | 2.06 |
| Silent | PSNR | - | -0.02 | -0.01 |
| | Points | 16 | 10.8 | 2.88 |

V. 결 론

본 논문에서는 H.264/AVC 표준 중에서 가장 연산량이 많으면서 성능에 가장 큰 영향을 끼치게 되는 움직임 추정기의 고속 움직임 예측을 위한 움직임 추정 알고리즘을 제안하였다. 그 중에서 많은 연구가 진행되어진 정화소 단위에서의 움직임 탐색이 아닌 부화소 단위에서의 움직임 탐색 알고리즘을 통해서 연산량과 화질 이 두가지 모두에서 이전의 알고리즘 보다 향상되어진 결과를 얻었다. 부화소 움직임 추정에서의 결과가 정화소 움직

임 벡터와 동일하게 나타나게 되는 경우가 상당히 높기 때문에 이러한 경우에는 부화소 움직임 추정을 하지 않고 움직임 추정을 마칠 수 있도록하여 연산량을 감소시켰으며 그렇지 않을 경우에는 예측된 부화소 MVP 지점을 중심으로 하여 CBCDS 알고리즘을 이용하여 이전의 알고리즘들 보다 정확하면서도 빠르게 최종 움직임 벡터를 탐색할 수 있다. 제안한 알고리즘을 시뮬레이션을 통해서 전역 탐색 알고리즘과 비교한 결과 연산량은 87.9% 감소하였으며 PSNR의 감소는 0.04dB 이하로 미비하였다.

참 고 문 헌

- [1] Joint Video Team (JVT) of ISO/IEC & ITU-T VCEG, "Draft ITU-T Recommendation an Final Draft International Standard of Joint Video Specification". ITU-R Rec. H.264|ISO/IEC 14496-10 AVC, 2003.
- [2] J. Ostermann, T. Wedi, et al., "Video Coding with H.264/AVC : tools, performance, amd complexity", IEEE Circuits and System Magazine, vol. 4, pp. 7-28, 2004
- [3] 최용일, 전병우, "H.264 표준의 가변 움직임 블록을 위한 고속 블록을 위한 고속 움직임 탐색 기법", 전자공학회 논문지 제41권 SP편 제6호, pp. 209-220, 2004년 11월.
- [4] A. M. Tourapis, O. C. Au, and M. L. Liou, "Predictive Motion Vector Field Adaptive Search Technique (PMVFAST)-Enhancing Block Based Motion Estimation", Proc. SPIE Visual Commun. Image Process., San Jose, CA, Jan. 2001.
- [5] Z. B. Chen, P. Zhou, and Y. He, "Fast Integer pel and Fractional pel Motion Estimation for JVT", JVT-F017, 6th meeting : Awaji, Japan, Dec. 2002
- [6] 라병두, 엄민영, 최윤식, "H.264에서 주변 움직임 벡터를 이용한 고속 부 화소 탐색 제어 기법", 전자공학회 논문지 제44권 SP편 제3호, pp. 16-22, 2007년 5월.
- [7] X. Yi and N. Ling, "Improved Partial Distortion Search Algorithm for Rapid Block Motion Estimation via Dual-halfway-stop", IEEE Int. Conf. Acoustics, Speech and Signal Processing, vol. 2, pp. 917-920, March 2005.
- [8] C. Du, Y. He and J. Zhengm "PPHS : A Parabolic Prediction-Baesd, Fast Half-Pixel Search Algorithm for Very Low Bit-Rate

Moving-Picture Coding”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13 No. 6, June 2003.

저 자 소 개



조 성 현(학생회원)
 2007년 울산대학교 전기전자정보
 시스템공학부 학사 졸업.
 2009년 울산대학교 전자공학정보
 시스템공학부 석사 졸업.
 <주관심분야 : 멀티미디어 시스
 템, VLSI 설계>



이 종 화(정회원)
 1974년 울산대학교 전기공학과
 학사 졸업.
 1976년 한국과학기술원
 재료공학과 석사 졸업.
 1986년 Brunel University
 물리학과 이학박사 졸업.
 1972년~현재 울산대학교 전기전자정보시스템
 공학부 교수
 1998년~1999년 Novosibirsk State Technical
 Visiting Scholar
 <주관심분야 : 반도체소자, VLSI 설계>