

논문 2008-45SP-3-9

움직임 예측을 이용한 고속 부화소 움직임 추정기

(Motion-based Fast Fractional Motion Estimation Scheme for H.264/AVC)

이 광 우*, 선우 명 훈**

(Kwang Woo Lee and Myung Hoon Sunwoo)

요 약

본 논문은 H.264/AVC 비디오 코덱의 부화소 움직임 추정 연산을 효율적으로 줄일 수 있는 고속 부화소 움직임 추정 알고리즘을 제안한다. 부화소 움직임 추정 연산은 보다 정확한 움직임 벡터를 찾을 수 있어 비디오 코덱에 널리 사용되지만, 추가적인 보간 및 탐색 연산으로 인해 부호화기의 연산량을 증가시키는 문제점이 있다. 제안하는 고속 부화소 움직임 추정 알고리즘은 SASR (Simplified Adaptive Search Range)을 이용하여 부화소 움직임 추정 연산을 선택적으로 수행하며 MSDSP (Mixed Small Diamond Search Pattern)을 이용하여 부화소 탐색 지점을 감소시켰다. 제안한 알고리즘은 전역 부화소 탐색 알고리즘과 비교하여 탐색 지점이 최대 93.2% 감소하였으며, PDFPS (Prediction-based directional fractional pixel search) 알고리즘보다 탐색 지점이 최대 81% 감소하며 PSNR 감소는 최대 0.04dB로 화질의 열화는 매우 미비했다.

Abstract

In an H.264/AVC video encoder, the motion estimation at fractional pixel accuracy improves a coding efficiency and image quality. However, it requires additional computation overheads for fractional search and interpolation, and thus, reducing the computation complexity of fractional search becomes more important. This paper proposes fast fractional search algorithms by combining the SASR (Simplified Adaptive Search Range) and the MSDSP (Mixed Small Diamond Search Pattern) with the predicted fractional motion vector. Compared with the full search and the prediction-based directional fractional pixel search, the proposed algorithms can reduce up to 93.2% and 81% of fractional search points, respectively with the maximum PSNR lost less than 0.04dB. Therefore, the proposed fast search algorithms are quite suitable for mobile applications requiring low power and complexity.

Keywords : H.264/AVC, Motion Estimation, Fractional Search, Video Coding

I. 서 론

H.264/AVC^[1]의 움직임 추정 연산은 이전 비디오 압축 표준보다 높은 압축률을 얻기 위해 다중 프레임 참조, 가변 블록 움직임 추정 및 보상 그리고 높은 움직임

벡터 정확도 등을 다양한 압축 방법을 지원한다. 그 결과 움직임 추정 연산이 전체 부호화기에서 차지하는 연산 비율이 60%^[2]가 넘어 하드웨어 및 실시간 구현에 가장 큰 걸림돌이 되고 있다.

움직임 추정 연산은 일반적으로 두 단계로 이루어진다. 먼저 정화소 단위에서 움직임 추정을 수행하여 블록 매칭 오차가 최소가 되는 지점을 찾는다. 그 후 최소 블록 매칭 오차 지점을 중심으로 부화소 움직임 추정을 수행하여 최소 블록 매칭 오차 지점을 찾는다. 부화소 움직임 추정은 MPEG-1, 2 와 H.263에서는 1/2화소 지점까지 MPEG-4와 H.264/AVC에서는 1/4화소 지점까지 탐색을 수행한다. 1/4화소까지 부화소 움직임을 수행하면 정화소 움직임 추정만을 수행하였을 때보

* 학생회원, ** 정회원, 아주대학교 전자공학부
(Department of Electrical Engineering, Ajou University)

※ 본 논문은 2008년 2단계 두뇌한국21, 지식경제부가 지원하는 국가 반도체 연구개발사업인 “시스템집적 반도체기반기술개발사업(시스템 IC 2010)”과 IDEC (IC Design Education Center)을 통해 개발된 결과임을 밝힙니다.

접수일자: 2007년12월17일, 수정완료일: 2008년4월15일

다 영상의 화질이 PSNR 측면에서 1~3dB 가량 향상된다. 그렇지만 부화소 움직임 추정은 부화소 보간 및 탐색 연산으로 인해 연산량이 크게 증가하게 된다.

일반적으로 정화소 움직임 추정 연산이 넓은 탐색 영역으로 인해 부화소 움직임 추정 연산보다 수십 배에 달하는 많은 연산량을 필요로 하였다. 따라서 움직임 추정기의 연산량 감소 연구의 대부분이 정화소 움직임 추정기에 대해서 이루어지고 있으며 그 결과 5개 이하의 정화소 탐색 지점만으로 정화소 움직임 추정이 가능한 고속 탐색 알고리즘^[3~4]들도 제안되고 있다. 그로 인해 16개의 부화소 지점을 탐색해야하는 부화소 움직임 추정기의 연산량이 상대적으로 커지게 되었다. 따라서 부화소 움직임 추정의 연산량을 감소시킬 수 있는 고속 부화소 움직임 추정 알고리즘이 필요하게 되었다.

본 논문은 부화소 탐색 과정의 특성을 이용해서 고속 부화소 움직임 추정 알고리즘을 제안하고 있다. 부화소 탐색 연산의 특성은 부화소 움직임 추정 과정을 모든 블록에 대해서 수행할 필요가 없으며, 부화소 움직임 벡터도 정화소 움직임 벡터와 유사하게 주변 블록의 부화소 움직임 벡터와 비슷한 방향성을 가지고 있다는 것이다. 이러한 부화소 연산 특성을 이용하여, 제안한 고속 탐색 알고리즘은 두 단계로 이루어져있다. 첫 번째 과정에서는 부화소 움직임 추정 과정을 수행할지 아니면 생략할 지를 선택하게 된다. 부화소 움직임 추정 과정을 생략할 수 있다면 부화소 생성을 위한 보간 연산과 부화소 탐색 과정을 모두 제거할 수 있으므로 연산량 감소에 매우 효과적이다. 그리고 두 번째 단계는 MSDSP (Mixed Small Diamond Search Pattern) 를 적용한 부화소 움직임 추정 과정이다. 제안한 고속 부화소 움직임 추정 알고리즘은 효과적으로 부화소 탐색 지점을 감소시킬 수 있으며 PSNR 열화는 매우 적었다.

본 논문의 구성은 다음과 같다. II장에서는 부화소 움직임 추정 연산과정의 특징을 살펴보고 III장에서는 새로운 고속 부화소 움직임 추정 알고리즘을 제안한다. IV장에서는 기존의 고속 부화소 움직임 추정 알고리즘과 성능을 비교한다. 마지막으로 V장에서 결론을 맺는다.

II. 부화소 움직임 추정 연산과정 특성

1. 부화소 움직임 벡터

부화소 움직임 추정 연산은 정화소 움직임 추정 연산을 수행하였을 때보다 정확한 블록의 움직임을 찾게 되

표 1. 정화소 움직임 벡터와 부화소 움직임 벡터 비율

Table 1. Ratio of between integer and fractional MV.

영상 시퀀스	정화소 움직임 벡터	부화소 움직임 벡터
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 %
Football	47.94 %	52.06 %

어 영상의 화질이 향상된다. 그렇지만 추가적인 부화소 보간 연산과 탐색 과정으로 인해 연산량이 증가하는 문제가 있다. 표1은 6개 영상을 부화소 움직임 추정 연산 후 결정된 최종 움직임 벡터 중 정화소 움직임 벡터와 부화소 움직임 벡터의 비율을 보여준다.

표 1에서 보인 것처럼 부화소 움직임 추정을 하여도 최종 움직임 벡터가 부화소 단위가 선택되는 것은 아니라는 것을 알 수 있다. 오히려 Akiyo, Container 등의 영상은 정화소 움직임 벡터의 비율이 더 높다는 것을 알 수 있다. 최종 움직임 벡터로 정화소 움직임 벡터가 선택될 경우는 부화소 연산 결과가 필요 없어지게 되므로 부화소 연산 과정을 생략할 수 있다. 따라서 정화소 움직임 벡터를 선택하게 되는 블록을 예측할 수 있다면 불필요한 부화소 연산과정을 생략하게 되어 최대 87% 가량에 달하는 부화소 움직임 추정 연산을 감소시킬 수 있다.

2. 부화소 움직임 벡터 예측

많은 고속 탐색 알고리즘은 초기 탐색 지점으로 MVp (predicted motion vector) 를 사용한다. MVp는 주변 블록의 움직임 벡터를 이용하는 미디언 예측기와 상위 블록 모드의 움직임 벡터를 이용하는 상위 모드 예측기를 이용해서 계산되어진다. 예측된 MVp는 정화소와 부화소 움직임 벡터가 포함되어 있으므로 식 (1)을 이용하여 부화소 움직임 벡터만을 따로 분리한다.

$$pred_frac_mv = (pred_mv - mv)\% \beta \quad (1)$$

식 (1)에서 $pred_mv$ 는 정화소와 부화소 움직임 벡터가 포함된 MVp이며 mv 는 정화소 움직임 추정을 수행한 후 얻은 정화소 움직임 벡터이다. 그리고 β 는 부화소 움직임 벡터의 정확도로 1/4 부화소 탐색을 하므로 4가 된다. 따라서 예측 MVp에서 정화소 움직임 벡터를 빼고

표 2. 부화소 움직임 벡터 분포도
Table 2. Fractional MV distribution.

영상 시퀀스	일치	[-1,+1]	[-2,+2]
Akiyo	87.1 %	86.7 %	99.27 %
Coastguard	31.31 %	75.86 %	93.84 %
Container	71.8 %	94.94 %	99.53 %
Foreman	26.54 %	71.64 %	91.99 %
News	78.93 %	93.1 %	98.43 %
Silent	71.16 %	90.42 %	97.43 %

4로 나눈 나머지가 부화소 MVp가 되며 탐색 중심점이 된다.

표 2는 부화소 MVp를 탐색 중심으로 하였을 경우 탐색 중심에서의 움직임 벡터 분포도를 정리하였다. 표 2에서 일치하는 부화소 MVp와 실제 부화소 움직임 벡터가 일치하는 경우이고 [-1,+1]은 실제 부화소 움직임 벡터가 부화소 MVp에서 상하좌우로 1/4화소 하나만큼 떨어져 있을 때이고 [-2,+2]는 1/4화소 두 개만큼 떨어져 있을 때이다.

대부분의 영상에서 부화소 MVp와 부화소 움직임 벡터가 일치할 경우가 70%를 넘었지만 Coastguard와 Foreman 같은 영상의 경우는 상대적으로 움직임이 많기 때문에 일치할 확률이 낮았다. 그렇지만 부화소 MVp를 중심으로 두 화소 안에 부화소 움직임 벡터가 있을 경우는 모든 영상에서 90%가 넘었다. 따라서 정화소 움직임 벡터 주변을 탐색하는 것보다 부화소 MVp를 중심으로 탐색을 수행하게 되면 초기에 최소 블록 매칭 오차지점을 찾아 연산을 종료할 수 있다.

III. 고속 부화소 움직임 추정 알고리즘

II장에서의 부화소 움직임 추정 연산의 특성을 바탕으로 고속 부화소 움직임 추정 알고리즘을 제안하였다. 제안한 알고리즘은 두 단계로 이루어져 있으며, 첫 번째는 SASR (Simplified Adaptive Search Range) 를 이용하여 부화소 움직임 추정 연산이 생략 가능한 블록을 예측하는 과정이고 두 번째는 부화소 탐색 과정에서의 탐색 지점을 감소시킬 수 있는 MSDSP를 적용한 부화소 탐색 과정이다.

1. SASR을 이용한 부화소 움직임 추정 연산 생략

일반적인 전역 부화소 움직임 추정 연산은 모든 블록에 대해 정화소 움직임 추정에서 찾은 최소 블록 매칭 지점을 중심으로 부화소 탐색을 수행한다. 그렇지만 표

1에서와 같이 부화소 움직임 추정 연산을 하여도 정화소 움직임 벡터가 선택되는 비율이 높았다. 그러므로 블록을 부화소 움직임 추정 연산이 필요한 블록과 그렇지 않은 블록으로 구분하여 부화소 움직임 추정 연산이 필요한 블록에 대해서만 부화소 탐색을 수행한다. 이와 같이 블록을 구분하기 위해서 ASR (Adaptive Search Range) 알고리즘^[4]을 이용하였다.

ASR은 정화소 탐색 중심의 블록 매칭 오차값을 이용하여 정화소 움직임 추정 연산의 탐색 영역을 결정하는 알고리즘이다^[4]. 만약 ASR의 계산값이 0이라면 탐색 영역은 0이 되고 이 경우는 블록이 움직임이 없다고 판단되어 움직임 벡터는 (0,0)이 된다. 이와 같이 블록의 움직임이 없다고 판단된 블록에 대해서는 부화소 움직임 추정을 하여도 정화소 움직임 추정보다 좋은 결과를 얻지 못하므로 부화소 움직임 추정을 수행하지 않는다. 따라서 블록의 움직임 여부에 대해서만 판단하면 되므로 ASR을 수정한 SASR을 제안했다.

$$SASR_{n,q} = \left\lfloor \frac{SAD_{n,q,0,0}}{Q} \times \frac{1}{4} \right\rfloor \quad (2)$$

식 (2)에서 $SAD_{n,q,0,0}$ 은 탐색 중심점의 블록 매칭 오차 값이며 Q 는 블록에 포함되어 있는 픽셀의 개수가 된다. 예를 현재 블록 크기가 4 x 4이면 Q 는 16이 되고, 8 x 8이라면 Q 는 64가 된다. 따라서 SASR의 계산값이 0이 되면 부화소 움직임 추정 연산을 수행하지 않고 생략 할 수 있어 부화소 연산량 감소에 효과적이다. 4로 나누었기 때문에, 부화소 단위의 움직임을 추정할 수 있어 생략이 수월해 진다. 표 3은 SASR을 적용한 결과 부화소 탐색이 생략 되는 블록 비율이다.

표 3의 결과에서 볼 수 있듯이 SASR을 적용하였을 때 부화소 연산이 생략되는 블록의 비율이 표 1에서 정화소 움직임 벡터를 선택하는 블록의 비율과 비슷하다는 것을 알 수 있다. SASR은 블록의 움직임을 이용하여 부화소

표 3. 부화소 움직임 추정 생략 비율
Table 3. Fractional motion estimation skip ratio.

영상 시퀀스	부화소 연산 생략 블록
Akiyo	89.62 %
Coastguard	25.63 %
Container	84.06 %
Foreman	16.47 %
News	82.72 %
Silent	74.85 %

움직임 추정을 생략하므로 움직임이 적은 영상은 생략되는 부화소 움직임 추정의 비율이 높았다. 그렇지만 움직임이 많은 영상은 생략되는 비율이 움직임이 적은 영상에 비해 상대적으로 낮아 부화소 탐색 연산량 감소에 효과적이지 못했다. 따라서 부화소 움직임 추정을 수행할 시에 부화소 탐색 지점을 감소시킬 수 있는 새로운 탐색 패턴을 제안했다.

2. Mixed Small Diamond Search Pattern

부화소 MVp를 초기 탐색점으로 부화소 움직임 추정을 수행하면 탐색 중심점 근처에서 움직임 벡터가 발견될 확률이 높아 고속 탐색 알고리즘에 적합하다. 표 2에서와 같이 부화소 움직임 벡터는 부화소 MVp 주변 $[-2,+2]$ 안에 대부분 존재하였지만 $[-2,+2]$ 안의 모든 부화소 지점을 탐색하게 되면 총 25개의 부화소 지점을 탐색하여야 하므로 오히려 전역 탐색보다 탐색점이 많아진다. 따라서 부화소 MVp 주변 $[-2,+2]$ 영역을 효과적으로 탐색하는 탐색패턴이 필요하다.

여러 고속 부화소 탐색 알고리즘에서 적은 탐색지점으로 좋은 탐색 성능을 보여주는 SDSP (Small Diamond Search Pattern)^[6]가 사용되고 있다. SDSP는 각 단계마다 탐색 중심점과 주변 상하좌우 4지점을 탐색하고 최소 블록 매칭 오차가 탐색 중심점에서 발견되면 탐색을 종료하고 그렇지 않으면 최소점을 중심으로 다시 탐색을 수행한다. 그렇지만 SDSP는 상하좌우만을 탐색하고 대각선 지점은 탐색을 하지 않으므로 잘못된 최소 블록 매칭 오차 지점을 찾게 될 수 있다.

제안한 MSDSP 탐색 패턴은 SDSP 탐색 단계에 대각선 탐색 지점을 추가하여 잘못된 최소 지점을 선택하여 다음 탐색 단계의 중심점이 되는 것을 막았다. 대부분의 움직임 벡터가 탐색 중심점(0,0)과 수직 또는 수평 방향에 집중되어 있는 특성^[6]을 가지므로 우선 SDSP 탐색을 적용하여 상하좌우 지점을 탐색하고 그 후에 대각선 지점을 추가로 탐색한다. 그 후 다시 SDSP를 수행한다. 따라서 MSDSP는 두 단계의 SDSP와 한 단계의 대각선 탐색으로 구성되어있으며 SDSP를 적용하였을 경우보다 정확한 최소 블록 매칭 오차지점을 찾을 확률을 증가시킬 수 있다.

그림 1에 MSDSP 탐색 방법을 도시하였으며 탐색 절차는 다음과 같다.

1. 식 (1)을 이용해 부화소 MVp를 계산하고 예측점을 초기 탐색 중심점으로 정한다.
2. MSDSP 탐색 중심점과 상하좌우 4지점에 대해서

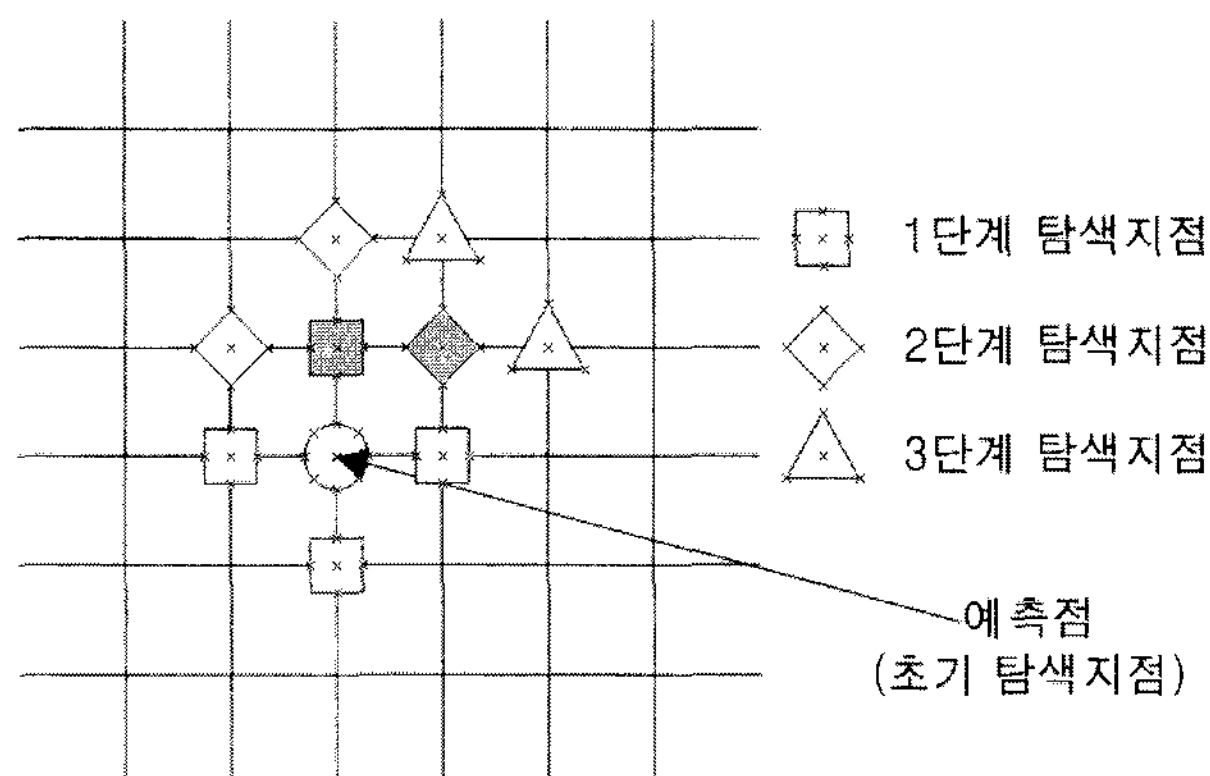


그림 1. MSDSP 탐색 방법
Fig. 1. MSDSP search.

블록 매칭 오차를 계산한다.

3. 최소 블록 매칭 오차점이 탐색 중심이라면 부화소 움직임 추정 연산을 종료하고 부화소 움직임 벡터가 결정된다. 만약 최소점이 탐색 중심이 아니라 주변에서 발견되었다면 다음 단계를 진행한다.

4. 탐색 중심점 주변 대각선 4지점에 대해서 탐색을 수행하여 최소 블록 매칭 오차 지점을 찾은 후 이전 단계에서 찾은 최소점과 비교하여 블록 매칭 오차 값이 더 작은 지점을 다음 탐색 단계의 중심으로 설정한다.

5. 이전 단계에서 결정된 최소점을 중심으로 상하좌우 4지점에 대해서 블록 매칭 오차값을 계산하여 최소점을 찾는다.

6. 최소 블록 매칭 오차점을 부화소 움직임 벡터로 결정하고 탐색을 종료한다.

IV. 성능 비교

제안한 고속 부화소 움직임 추정 알고리즘을 QCIF(176 x 144) 크기의 영상 시퀀스(Akiyo, News, Silent, Coastguard, Foreman, Container)에 테스트 하였다. 부화소 움직임 추정 연산 성능 비교를 위해 정화소 움직임 추정은 전역 탐색 방법을 이용하였으며 탐색 범위는 16이며 참조 프레임은 한 장으로 설정하였다. 그리고 첫 번째 프레임을 제외하고는 모든 프레임이 P-프레임으로 복호화를 진행하였다.

제안한 알고리즘을 전역 부화소 움직임 추정과 CBFPS (Center Biased Fractional Pixel Search)^[7] 그리고 PDFPS (Prediction-based Directional Fractional Pixel Search)^[8]와 비교하였다. CBFPS는 H.264/AVC joint model에 포함되어있는 고속 탐색 알고리즘으로 초기 탐색지점으로 미디언 예측기를 사용하며 SDSP

탐색 패턴을 이용하여 부화소 탐색을 수행한다. PDFPS는 초기 탐색지점 설정을 위해 미디언 예측기와 상위모드 예측기를 이용하며 방향성 탐색과 3단계의 SDSP 탐색을 수행한다. 또한 threshold를 두어서 블록 매칭 오차 값이 threshold보다 작으면 탐색을 종료하는 early termination 방법이 적용되어 있다. 표 4에 제안한 고속 탐색 알고리즘을 전역 탐색, CBFPS, PDFPS 방법과 탐색 지점과 PSNR 측면에서 성능을 비교하였다.

표 4에서 보인 것처럼 제안한 고속 알고리즘은 다른 고속 알고리즘보다 적은 탐색 지점을 찾으면서도 PSNR의 감소는 작은 것을 알 수 있다. 전역 부화소 탐색에 비해서는 탐색 지점과 PSNR 감소는 평균 83%와 0.03dB였으며, CBFPS에 비해서는 탐색지점은 73% 감소하였고 PSNR은 0.28dB 감소하였다. 그리고 PDFPS 알고리즘보다는 26% 가량 부화소 탐색 지점이 감소하였으며 PSNR

표 4. 기존 고속 부화소 탐색 알고리즘과의 성능 비교

Table 4. Comparisons with existing fast fractional search.

영상 시퀀스	탐색 방법	탐색지점	PSNR 변화량
Akiyo	전역 탐색	16	-
	CBFPS	10.48	0
	PDFPS	2.76	-0.01
	Proposed	1.09	-0.02
Coastguard	전역 탐색	16	-
	CBFPS	10.86	0.01
	PDFPS	4.54	0.02
	Proposed	4.53	-0.01
Container	전역 탐색	16	-
	CBFPS	10.36	0.01
	PDFPS	2.8	0.03
	Proposed	1.66	-0.01
Foreman	전역 탐색	16	-
	CBFPS	11.04	-0.04
	PDFPS	5.59	-0.08
	Proposed	4.23	-0.06
News	전역 탐색	16	-
	CBFPS	10.58	0.03
	PDFPS	3.2	0.04
	Proposed	2.14	-0.02
Silent	전역 탐색	16	-
	CBFPS	10.8	-0.02
	PDFPS	3.86	0
	Proposed	2.95	0
Football	전역 탐색	16	-
	CBFPS	11.45	-0.03
	PDFPS	6.23	-0.03
	Proposed	5.74	-0.02

열화는 0.03dB가 발생하였다. 위의 시뮬레이션 결과 제안한 고속 탐색 알고리즘이 다른 알고리즘에 비해 PSNR 감소는 미비하면서 효과적으로 탐색 지점을 감소시켰음을 알 수 있다.

특히 움직임이 적은 Akiyo, Container, News와 같은 영상에서는 SASR 알고리즘으로 인해 다른 고속 탐색 알고리즘보다 탐색 지점의 감소가 컸다. 그리고 Foreman, Coastguard와 같이 움직임이 많은 영상의 경우에는 SASR 알고리즘을 통한 부화소 탐색 생략 비율이 적어 연산량 감소가 크지 않았지만 PDFPS방법과 비슷한 탐색 지점을 가지면서 보다 좋은 PSNR 성능을 가짐을 알 수 있다. Table 4.를 보면 PSNR의 수치가 전역 부화소 탐색에 비하여 증가하는 경우를 볼 수 있다. 이는 부화소 전역 탐색이라 해서 integer full search와는 탐색방식이 다르다. 우선, 1/2 지점에서 전역탐색한 후 그 주변 1/4 지점을 전역탐색 하기 때문에 모든 1/4 화소가 탐색되는 것이 아니다. 이러한 이유로 PSNR이 증가할 가능성이 있다.

V. 결 론

본 논문에서는 부화소 움직임 추정의 특성을 이용한 새로운 고속 부화소 움직임 추정 알고리즘을 제안하였다. 모든 블록에 대해서 부화소 움직임 추정 연산을 수행하여도 최종 움직임 벡터가 정화소 움직임 벡터일 경우가 많이 존재한다. 이러한 블록을 SASR을 통해 예측하여 불필요한 부화소 움직임 추정 연산을 생략하였다. 그리고 부화소 움직임 추정의 탐색 지점 감소를 위해 부화소 MVp를 탐색 중심으로 하는 MSDSP 탐색방법을 제안하였다. 다른 부화소 추정기의 경우는 복잡한 연산을 사용해야 하는데, 제안한 알고리즘은 한번의 쉬프트 연산으로 부화소 탐색 여부를 결정해서 연산 속도가 빨라졌다. 게다가 다이아몬드 탐색을 방식을 보완하여 정확성을 높이기도 하였다. 시뮬레이션 결과 전역 부화소 움직임 추정에 비해 최대 93.2%, PDFPS 방법에 비해 81%의 부화소 탐색 지점이 감소하였으며 PSNR 감소는 매우 미비했다.

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] Y. J. Wang, C. E. Cheng, and T. S. Chang, "A fast fractional pel motion estimation algorithm for H.264/MPEG-4 AVC," IEEE Int. Symp. On Circuits and Systems, 2006, pp.3974-3977
- [3] X. Q. Banh and Y. P. Tan, "Adaptive dual-cross search algorithm for block-matching motion estimation," IEEE Trans. On Consumer Electronics, vol. 50, no. 2, pp. 766-775, 2004.
- [4] 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, 2005, pp. 917-920
- [5] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," IEEE trans. Image Proceeding, vol. 11, pp.1442-1449, 2002.
- [6] B. La, M. Eom, Y. Choe, "Fast Sub-pixel Search control by using Neighbor Motion Vector in H.264," IEEE Int. Conf. on Advanced Communication Technology, 2007, pp. 62-65
- [7] Z. B. Chen, P. Zhou, and Y. He, "Fast integer pel and fractional pel motion estimation for JVT," JVT-F017, 6th meeting, 2002.
- [8] L. Yang, K. Yu, J. Li, and S. Li, "Prediction-based directional fractional pixel motion estimation for H.264 video coding," IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, 2005, pp. 901-904

— 저 자 소 개 —



이 광 우(학생회원)
2006년 아주대학교 전자공학부
학사 졸업.
2008년 아주대학교 전자공학부
석사 졸업.
<관심분야 : 멀티미디어 시스템
SoC>



선우 명 훈(정회원)
1980년 서강대학교 전자공학과
학사 졸업.
1982년 한국과학기술원 전자공학
석사 졸업.
1982년~1985년 한국전자통신
연구소 (ETRI) 연구원.
1985년~1990년 Univ. of Texas at Austin
전자공학 박사.
1990년~1992년 Motorola, DSP Chip Division
(미국).
1992년~1996년 아주대학교 전기전자공학부
조교수.
1996년~2001년 아주대학교 전자공학부 부교수.
2001년~현재 아주대학교 전자공학부 교수
300편 이상의 논문, 37개 특허 출원 및
등록, IEEE Workshop on Signal
Processing System (SIPS) 2005 Best
Student Paper Award 포함 20회 이상의
논문상 수상. IEEE SIPS 2003 Technical
Program Chair 역임.
2007년 현재 IEEE SIPS, Cool chips, Design
Automation and Test in Europe
(DATE), IEEE International ASIC/SOC
Conference, Asian-Pacific Conference
on CAS (APC-CAS), Asian-Solid State
Circuits Conference (A-SSCC), Inter-
national SOC Design Conference
(ISOCC), International Symposium on
VLSI Design, Automation and Test
(VLSI-DAT) 등의 국제 학회 committee
위원.
<주관심분야 : VLSI 및 Parallel Architecture, 통
신 멀티미디어용 DSP 칩 및 SOC 설계>