

논문 2011-48SP-1-8

# 부화소 움직임 추정을 위한 효과적인 화소 재구성 알고리즘

## (Efficiency Pixel Recomposition Algorithm for Fractional Motion Estimation)

신 왕 호\*, 선우 명 훈\*\*

(Wang Ho Shin and Myung Hoon Sunwoo)

### 요 약

본 논문은 H.264/AVC 비디오 코덱의 부화소 움직임 추정 연산을 효율적으로 줄일 수 있는 화소 재구성 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. 본 논문은 화소 재구성 알고리즘을 이용한 부화소 움직임 추정기를 제안 하였다. 일반적인 부화소 움직임 추정기는 보간 시 같은 화소를 중복하여 보간한다. 이러한 중복 보간은 연산량을 증가 시키는 원인이 된다. 화소 재구성 알고리즘은 화소의 중복 보간을 제거하여 부화소 움직임 추정기의 연산량을 감소시킨다. 제안한 알고리즘은 고속 전역 부화소 탐색 알고리즘과 비교하여 보간 시 연산량이 18.1% 감소하였으며, PSNR 감소는 평균 0.067dB로 화질의 열화는 매우 미비했다. 그러므로 제안한 알고리즘은 저전력의 고효율을 요구하는 모바일 응용에 적합하다.

### 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 a Pixel Re-composition Fractional Motion Estimation (PRFME) algorithm for an H.264/AVC video encoder. Fractional Motion Estimation performs interpolation for the overlapped pixels which increases the computational complexity. PRFME can reduce the computational complexity by eliminating the overlapped pixel interpolation. Compared with the fast full search, the proposed algorithm can reduce 18.1% of computational complexity, meanwhile, the maximum PSNR degradation is less than 0.067dB. Therefore, the proposed PRFME algorithm is quite suitable for mobile applications requiring low power and complexity.

**Keywords :** H.264/AVC, FME, PRFME, Pixel Re-composition Fractional Motion Estimation, 움직임 추정, 보간

## I. 서 론

\* 학생회원, \*\* 정회원, 아주대학교 정보통신대학

전자공학부

(School of Electrical and Computer Engineering, Ajou University)

※ This work was supported by the IT R&D program of MKE/KEIT. [KI002145, High Speed Digital Signal Processing based CMOS Circuit Design for Next-generation Optical Communication].

접수일자: 2010년6월28일, 수정완료일: 2010년10월4일

동영상 압축 표준들은 동영상 데이터에 존재하는 중복성을 제거하여 데이터의 압축률을 향상시킨다. 동일 프레임 내에서 화소들 간의 통계적 발생 확률에 의한 통계적 중복성과 공간적 중복성, 그리고 프레임 사이에 존재하는 시간적 중복성은 동영상 데이터의 대표적인 특징들이다. 이 중에서 동영상의 압축률을 가장 효과적으로 높일 수 있는 시간적 중복성 제거는 이전 영상과

현재 영상의 유사성을 이용하여 움직임 추정 및 보상을 통해 이루어진다. H.264/AVC<sup>[1]</sup>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%<sup>[2]</sup>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 움직임 추정을 수행하면 정화소 움직임 추정만을 수행하였을 때보다 영상의 화질이 PSNR 측면에서 1~3dB 가량 향상된다. 그렇지만 부화소 움직임 추정은 부화소 보간 및 탐색 연산으로 인해 연산량이 크게 증가하게 된다. 일반적으로 부화소 움직임 추정 연산은 1/2화소와 1/4화소 보간으로 인해 정화소 움직임 추정 연산보다 수십배에 달하는 많은 연산량을 필요로 한다. 따라서 부화소 움직임 추정기의 연산량 감소를 위하여 많은 연구가 여러 방면에서 이루어지고 있다. 우선 가장 활발히 이루어진 연구는 탐색지점을 최소 하여 빠르게 매칭 블록을 찾는 고속 탐색 알고리즘이다.<sup>[4]</sup> 논문은 고속 3-스텝 탐색을 제안하였다. 기존의 고속 탐색 알고리즘에 비하여 탐색 포인트를 줄임으로써 연산량을 감소 시켰다.<sup>[5]</sup> 논문은 다이아몬드 탐색을 이용하여 빠르게 최적의 매칭 포인트를 찾아 연산 포인트를 줄임으로써 연산량을 감소시키는 알고리즘이다. 위의 [4~5]과 같은 탐색 포인트를 줄여 연산량을 감소시키는 알고리즘들은 탐색 포인트 감소를 통하여 연산량을 줄일 수는 있으나 고속 전역 탐색에 비하여 정확도가 낮다.<sup>[6]</sup> 논문은 서브 블록 41개를 빠르게 생성하여 최적의 블록 모드를 최단시간 결정하는 알고리즘이다. 이 알고리즘은 부화소 움직임 추정기가 최적의 값을 찾는 시간이 단축 되나, 4x4 블록 SAD 생성기를 21개나 가지고 있어 하드웨어 구현에 적하지 않다.

본 논문의 구성은 다음과 같다. II장에서는 JM 참조 모델에서 사용되고 있는 H.264/AVC의 블록매칭 방법

과 부화소 움직임 벡터에 대하여 설명한다. III장에서는 화소 재구성 알고리즘을 이용한 부화소 움직임 추정에 대하여 설명한다. 그리고 IV장에서는 실험 결과와 기존의 방법과의 비교 결과를 분석하고, 이를 토대로 V장에서 결론을 맺는다.

## II. 움직임 추정 알고리즘

### 1. 블록 매칭

움직임 추정에서 블록 매칭은 동영상 데이터 압축에서 가장 중요한 역할을 한다. 가장 좋은 압축 성능을 보이는 블록 매칭 방법은 FS(Full Search)이다. 그러나 이 방법은 너무 많은 연산량을 필요로 하기 때문에 이를 극복하기 위한 고속 블록 매칭 방법들이 제안되었다.

그 중에서 하드웨어 구현에 많이 사용되는 고속 블록 매칭 방법은 TSS(Three Step Search)와 NTSS 이다.

표 1과 같이 NTSS는 다른 탐색 알고리즘과 비교하여 FS와 가장 근접한 성능을 보이고 있다. 다른 고속 탐색 알고리즘들은 탐색영역이 좁은 경우 좋은 성능을 보이지만 NTSS는 탐색영역이 좁거나 넓은 경우에 모두 우수한 성능을 보인다.

표 1. 블록 매칭 알고리즘 비교  
Table 1. Comparison of block matching algorithms.

	블록매칭 방법			
	FS	TSS	Hexagon Search	NTSS
평균 PSNR(dB)	32.32	32.30	32.28	32.32
평균 SAD	1011.36	1013.43	1016.90	1011.18

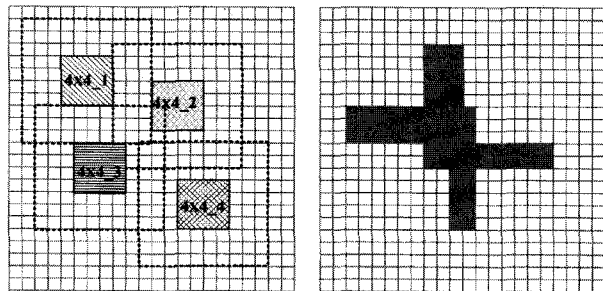
### 2. 부화소 움직임 벡터

부화소 움직임 추정 연산은 정화소 움직임 추정 연산을 수행하였을 때보다 정확한 블록의 움직임을 찾게 되어 영상의 화질이 향상된다. 그렇지만 추가적인 부화소 보간 연산과 탐색 과정으로 인해 연산량이 증가하는 문제가 있다. 표 2는 6개 영상을 부화소 움직임 추정 연산 후 결정된 최종 움직임 벡터 중 정화소 움직임 벡터와 부화소 움직임 벡터의 비율을 보여준다. 표 2에서 보인 것처럼 움직임이 적은 Akiyo, Container 등의 영상은 정화소 움직임 벡터의 비율이 더 높다. 그러나 움직임이 많은 영상일 경우 부화소 움직임 추정이 선택되는

표 2. 정화소 움직임 벡터와 부화소 움직임 벡터 비율

Table 2. Ratio of between integer and fractional MV.

영 상	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 %



(a) 4x4 블록 화소 보간 (b) 중복화소 보간

그림 1. 부화소 움직임 추정의 화소 보간  
Fig. 1. FME interpolation.

비율이 높다. 부화소 움직임 추정은 움직이 많은 영상에서 정화소 움직임 추정에 비하여 더욱 정확한 움직임 예측을 할 수 있다.

그러나 부화소 움직임 추정의 단점인 많은 연산량은 영상을 실시간 처리하기 위해서 많은 부담이 된다.

부화소 움직임 추정에서 연산량이 가장 큰 이유는 똑같은 화소를 중복하여 보간을 수행하기 때문이다. 그림 1의 (a)와 같이 4x4블럭 4개를 연산하기 위해서는 각각의 블록의 화소값과 보간에 필요한 상하좌우 3화소를 가져와 보간을 수행한다. 이렇게 보간을 수행하면 그림 1의 (b)와 같이 중복되는 화소를 여러번 반복하여 보간을 수행하게 된다. 이렇게 중복되는 보간으로 인하여 부화소 움직임 추정은 많은 연산을 수행하게 된다.

### III. 화소 재구성 알고리즘을 이용한 부화소 움직임 추정

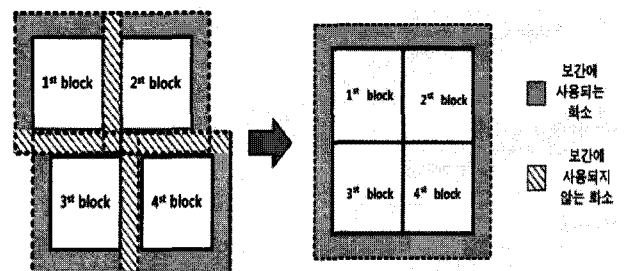
II절에서의 부화소 움직임 추정 연산의 특성을 바탕

으로 화소 재구성을 이용한 부화소 움직임 추정 알고리즘을 제안한다. 제안한 알고리즘은 두 단계로 이루어져 있다. 첫 번째는 보간에 필요한 화소를 재구성 하는 단계이며, 두 번째 단계는 화소 보간 및 부화소 움직임 추정을 수행하는 단계이다.

#### 1. 화소 재구성

일반적으로 부화소 움직임 추정을 하기 위해서는 각각 블록의 화소값과 상하좌우 3 화소를 가져와 보간을 수행한다. 그러나 이러한 보간의 수행은 그림 1의 (b)와 같이 중복되는 화소의 보간을 수행함으로써 같은 연산을 여러번 반복하여 불필요한 연산량을 늘리는 결과를 가져온다. 이러한 중복 화소 보간을 없애기 위하여 그림 2와 같은 화소 재구성을 제안하였다. 그림 2의 (a)와 같은 블록의 화소들과 상하좌우 3 화소들을 이용하여 보간 수행 시 중복되는 화소들을 그림 2의 (b)와 같이 중복되는 화소들을 제거하고 원래의 블록 화소와 외곽의 상하좌우 3화소를 가져와 보간을 수행한다. 이렇게 보간을 수행하면 보간 수행 시 중복되는 화소를 보간하지 않기 때문에 부화소 움직임 추정 수행시 많은 연산량을 줄일 수 있다.

제안한 화소 재구성 부화소 움직임 추정 알고리즘은 중복 보간을 생략하여 연산량을 감소시킨다. 이는 보간 시 필요한 6 탭 필터의 사용이 감소하고 메모리에서 화소를 읽어오는 횟수가 감소함으로써 전체적으로 연산량과 에너지 소비의 감소를 가져오게 된다. 화소 보간 시 화소 재구성 알고리즘을 사용 할 경우 연산량의 감소는 표2와 같다. 표3는 영상 한 프레임에 보간 할 때의 연산량을 보여준다. 고속 전역탐색 부화소 알고리즘에 비하여 연산량이 평균 18.1% 줄어드는 것을 알 수 있다. 제안하는 화소 재구성 알고리즘은 화소의 보간 전에 보



(a) 화소 재구성 전 (b) 화소 재구성 후

그림 2. 부화소 움직임 추정의 화소 재구성  
Fig. 2. Sub-block combination.

표 3. 부화소 움직임 추정 보간 연산량 비교

Table 3. Interpolation comparison of fractional motion estimation

영 상	고속 부화소 탐색	제한한 알고리즘	감소율(%)
Akiyo	19232	16346	16%
Coastguard	28216	23426	16.9%
Container	25486	20134	21%
Foreman	22096	18432	16.6%
News	27992	21468	23.4%
Silent	26776	21542	19.6%

간 시 중복하여 보간하는 화소를 미리 제거한다. 그러므로써 매크로 블록을 보간하는데 필요한 화소의 수가 일반적인 부화소 움직임 추정의 화소 보간보다 적게 들어간다. 따라서 화소 보간을 위한 메모리 액세스가 줄어들며 보간의 연산량 감소 효과를 볼수 있다.

예를 들어 그림 1과 같이 4x4 블록 4개를 보간하는데 일반적인 부화소 움직임 추정기는 10x10 블록 4개가 필요하다. 총 400개의 화소가 있어야 4x4 4개 블록을 보간 할 수 있다. 그러나 제안한 화소 재구성 알고리즘을 이용하여 화소 중복을 제거하면 14x14 블록 하나 만을 이용하여 보간을 수행 할수 있다. 화소 보간에 필요한 픽셀은 총 196 화소이다. 일반적인 부화소 움직임 추정기와 비교하여 51%의 보간에 필요한 화소를 줄일 수 있다. 그로인하여 메모리 액세스 감소 효과를 볼 수 있다.

제안한 화소 재구성 알고리즘을 이용하여 부화소 움직임 추정의 화소 보간 수행 시 원래의 필요한 화소가 아닌 주위의 다른 화소를 이용하여 보간을 수행하는 경우가 발생한다. 그러나 영상에서 하나의 픽셀과 그 주위의 픽셀 하나의 블록과 그 주위의 블록들은 유사한 값을 가지고 있다. 그래서 화소 재구성 알고리즘을 적용하여도 부화소 움직임 추정의 성능을 크게 떨어 들이지 않는다. 이는 IV성능비교를 통하여 증명하였다.

2. 화소 보간 및 부화소 움직임 추정

화소 재구성 부화소 움직임 추정기는 그림 3과 같다. 화소 재구성이 완료되고 나면 22x1의 1/2 보간 필터를 이용하여 보간을 수행한다. 여기서 22x1의 보간을 한번에 수행하는 이유는 보간에 필요한 화소를 한 번에 가

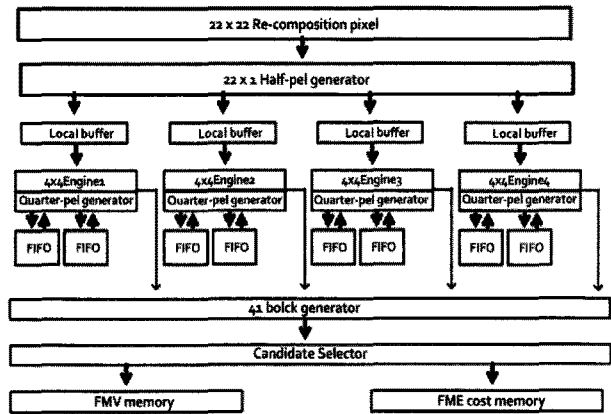


그림 3. 화소 재구성 움직임 추정기  
Fig. 3. SCFME hardware design.

져와 같은 보간을 수행함으로써 같은 화소를 반복하여 가져와 중복 보간을 수행하는 것을 방지하기 위함이다. 1/2 보간이 수행된 화소를 4x4 단위로 나누어 1/4 보간을 수행한다. 앞에서 22x1 화소를 한번에 처리하였으므로 4x4 블록 4개를 동시에 1/4 보간을 수행할 수 있다. 1/2 보간과 1/4 보간이 완료된 4x4 블록들은 다시 41 블록 생성기에 모이게 된다. 앞 단계에서 4x4 블록 단위로 쪼개진 블록들을 이용하여 상위 블록인 4x8, 8x4, 8x8, 16x8, 8x16, 16x16을 만들어 낸다. 이와 같이 모든 과정이 수행되면 7가지 모드 41개 블록이 모두 생성된다. 이렇게 생성된 41개 블록을 이용하여 현재의 영상과 가장 적합한 블록을 찾는다.

이때 단순히 화소 값만을 이용하여 비교를 할 경우 모든 매크로 블록에서 최상위 블록인 16x16을 선택하게 된다. 단순히 화소값을 덧셈 트리를 이용하여 각각의 블록을 만들고 비교하게 되면 화소의 합 값은 모두 동일하다. 이러한 문제를 해결하기 위하여 블록 비교 시 블록들이 가지고 있는 움직임 벡터 값을 더하여 비교함으로써 움직임이 적으면서 현재 매크로 블록과 가장 유사한 화소 값을 가지는 블록을 결정 하도록 제안 하였다.

$$SAD_{8x4\_block\_1} = SAD_{4x4\_block\_1} + SAD_{4x4\_block\_2} + MV_{4x4\_block\_1} + MV_{4x4\_block\_2} \tag{1}$$

식 (1)은 4x4 블록을 이용하여 한 단계 위의 8x4 블록을 생성하는 하는 식이다. 블록의 화소 값이 SAD<sub>4x4\_block</sub>와 블록의 움직임 벡터인 MV<sub>4x4\_block</sub>를 이용하여 상위 블록의 화소값을 생성한다. 이로 인해 블록 생성 시 화소값과 움직임 벡터값을 이용하여 비교함

으로써 보다 정확한 부화소 움직임 예측이 가능하다.

### 3. 제안하는 알고리즘의 적용

인코더에서 제안하는 화소 재구성 부화소 움직임 추정기는 아래 그림 4와 같이 적용한다. 우선 정화소 움직임 추정기를 이용하여 얻어진 41개의 SAD값과 41개의 움직임 벡터 값을 이용하여 움직임 보상기에서 최적의 값을 갖는 매크로 블록을 크기의 서브블록들을 결정한다. 결정된 서브 블록들의 움직임 벡터를 화소 재구성기에서 받아 화소 재구성기는 움직임 벡터에 따라 서브 블록들의 구성 화소와 보간에 필요한 상하좌우 화소 중 보간 시 중복 되지 않는 화소들을 메모리에서 읽어온다.

이렇게 메모리에서 읽어온 화소를 22x22 보간 픽셀로 재구성 한 후에 그림 3의 움직임 추정기를 이용하여 22x1 화소를 매 클럭마다 1/2 보간을 수행한다. 1/2 보간기를 거친 화소는 4x4 단위로 서브 블록을 만들고 에더트리를 이용하여 상위 블록을 만드는 과정을 수행하고, 1/2 부화소 탐색을 수행한다. 탐색을 수행하는 동안 1/2 부화소를 잉요하여 1/4 부화소 보간도 동시에 수행한다. 1/2 부화소 탐색에서 결정된 최적의 탐색지점을 기점으로 다시 1/4 부화소 탐색을 수행하여 부화소 움직임 추정을 완료 한다.

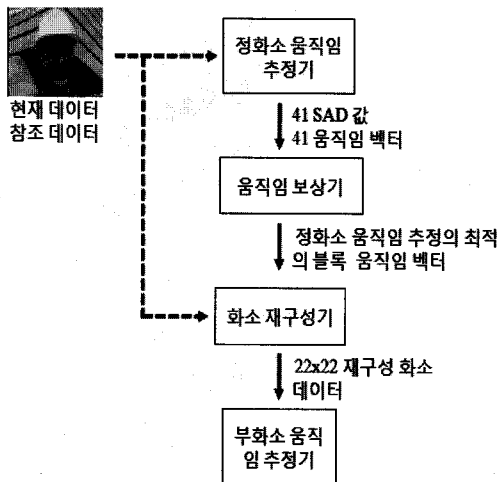


그림 4. 화소 재구성 알고리즘 적용  
Fig. 4. Flow chart of proposed SCFME.

## IV. 실험

제안한 화소 재구성 부화소 움직임 추정 알고리즘을 QCIF(176 x 144) 크기의 영상 시퀀스(Foreman, Akiyo,

표 4. 부화소 움직임 추정 성능 비교

Table 4. Performance comparison of fractional motion estimation.

영 상	정화소 탐색	고속 부화소 탐색	제안한 알고리즘
Container	35.825	36.109	36.088(-0.21)
Mobile	34.29	34.492	34.48(-0.01)
Silent	35.825	36.109	36.071(-0.38)
Foreman	36.4	37.423	37.413 (-0.01)
Akiyo	37.88	38.35	38.339 (-0.011)
Car-phone	37.48	37.72	37.685 (-0.035)
Coastguard	34.37	34.67	34.517 (-0.153)

표 5. 다른 부화소 움직임 추정 알고리즘과 성능 비교

Table 5. Performance comparisons with other algorithm.

PSNR	[7]	[9]	[10]	제안한 알고리즘
Container	*	*	34.77	36.088
Mobile	*	34.257	31.64	34.48
Silent	*	*	34.36	36.071
Foreman	34.28	35.706	34.34	37.413
Akiyo	*	*	36.83	38.339
Car-phone	*	37.189	*	37.685
Coastguard	32.46	*	*	34.517
평 균	33.37	35.717	34.388	36.37

Car-phone, Coastguard, Container, Soccer, Mobile)에 시험 하였다. 부화소 움직임 추정 연산 성능 비교를 위해 정화소 움직임 추정은 전역 탐색 방법을 이용하였으며 탐색 범위는 32이며 참조 프레임은 한 장으로 설정하였다. 그리고 첫 번째 프레임을 제외하고는 모든 프레임 P-프레임으로 복호화를 진행하였다. 제안한 알고리즘을 비교하였다.

표 4에서 보인 것처럼 제안한 고속 알고리즘은 정화소 탐색 알고리즘보다 PSNR의 성능은 높다. 그리고 고속 전역 탐색에 비하여 PSNR의 조금 떨어지지만 표 3에서 알 수 있듯이 연산량은 고속 전역 부화소 탐색에 비해서는 적다는 것을 알 수 있다. PSNR 감소는 고속 전역 탐색에 비하여 평균 0.067dB 감소하였고 연산량은

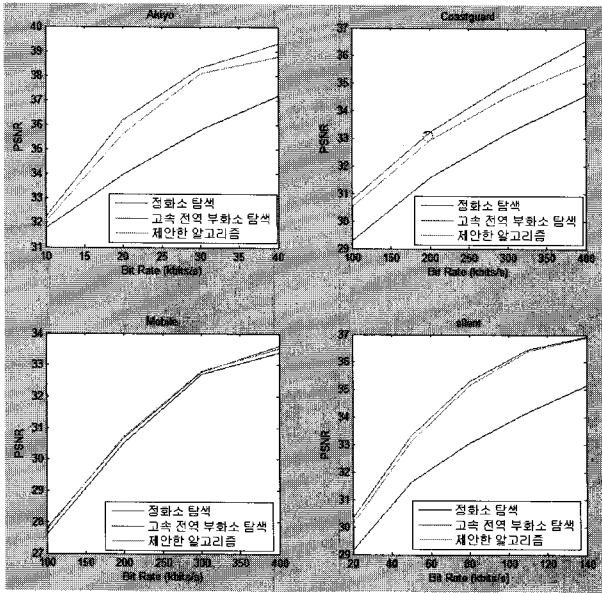


그림 5. 영상의 RD-Curve 곡선  
Fig. 5. RD-curve comparisons.

평균 18.1% 감소하였다.

표 5는 다른 부화소 탐색 알고리즘들과 제안한 화소 재구성 부화소 움직임 추정 알고리즘을 비교한 표이다. 제안한 알고리즘이 다른 알고리즘에 비해 PSNR 감소는 미비하면서 효과적으로 연산량을 감소시켰음을 알 수 있다. 각각의 알고리즘과 비교하여 보았을 때 평균 1dB 이상 성능이 좋음을 알 수 있다.

그림 5는 각 영상 RD\_curve 곡선을 나타낸 그림이다. 그림에서 보여지는 것처럼 제안한 알고리즘은 정확소 움직임 추정에 비해서는 좋은 성능을 보이고, 고속 전역 부화소 탐색에 비하여서는 약간의 성능 열화가 있다. 그러나 제안한 알고리즘은 부화소 움직임 추정의 연산량을 18% 이상 감소시키는 것에 비하여 성능의 열화는 미비하다는 것을 알 수 있다.

### V. 결 론

본 논문에서는 부화소 움직임 추정의 특성을 이용한 화소 재구성 부화소 움직임 추정 알고리즘을 제안하였다. 모든 블록에 대해서 부화소 움직임 추정 연산을 수행하여도 기존의 부화소 움직임 추정기보다 연산량을 감소시키는 알고리즘을 제안하였다. 화소 보간 시 중복 보간을 제거하여 연산량 감소 및 메모리 접근을 감소시켰다. 다른 부화소 추정기의 경우는 복잡한 연산을 사용하여 부화소 움직임 추정기의 성능을 높이는 반면, 제안한 알고리즘은 보간에 필요한 화소를 초기에 읽어

와 중복 보간 요소를 제거하고 덧셈 트리를 이용하여 41개 블록을 생성 하고 비교함으로써 보다 효과 적으로 부화소 움직임 추정을 할 수 있다. 시뮬레이션 결과 제안한 알고리즘은 다른 알고리즘에 비하여 PSNR의 최소 1db 이상 높음을 알 수 있다. 또한 고속 전역 부화소 탐색 알고리즘과 비교하여 보간 시 연산량이 18.1% 감소하였으며, PSNR 감소는 평균 0.067dB로 화질의 열화는 매우 미비했다. 그러므로 제안한 알고리즘은 저전력의 고효율을 요구하는 모바일 응용에 적합하다.

### 참 고 문 헌

- [1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC), Mar. 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, pp.3974-3977, 2006.
- [3] 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, pp. 917-920, 2005.
- [4] 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.
- [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] Nam Thang Ta, Jim Rim Choi, Jae Hoon Kim, Seon Cheol Hwang, Shi Hye Kim, "Fully parallel fractional motion estimation for H.264 AVC encoder" Intelligent Computing and Intelligent Systems, ICIS 2009. IEEE International Conference on, 2009.
- [7] Aminlou. A. , Alvandi. P and Hashemi. M.R. "Low complexity hardware implementation of reciprocal fractional motion estimation for H.264/AVC in mobile applications," Picture Coding Symposium, PCS 2009, 2009.
- [8] Yiqing Huang, Qin Liu and Ikenaga, T "Highly Parallel Fractional Motion Estimation Engine for Super Hi-Vision 4k×4k@60fps," Multimedia Signal Processing, MMSP '09. IEEE International Workshop on, 2009.
- [9] Lin, Y.; Wang, Y.C, "Improved parabolic

prediction-based fractional search for H.264/AVC video coding” in Proc. PCS 2009, iet-ipr.2008.01.92, Page(s): 261 - 271

- [10] Hyosun Yoon, Hyesuk Kim, Miyoung Kim, Lai Nga; Gueesang Lee, “Hierarchical Integer Pixel and Adaptive Fractional Pixel Motion Estimation” in Proc. CIT Workshops 2008, Page(s): 391 - 395

---

저 자 소 개

---



신 왕 호(학생회원)

2008년 호서대학교 New IT 공과  
대학 전자공학 학사 졸업  
2010년 아주대학교 정보통신대학  
전자공학 석사 졸업  
2010년 현재 삼성전자(주) 연구원

<주관심분야 : 멀티미디어 코덱, 멀티미디어 신호  
처리, 멀티미디어용 SOC설계, 저전력 반도체 설  
계>



선우명훈(정회원)-교신저자

1980년 서강대학교 전자공학  
학사 졸업  
1982년 한국과학기술원 전자공학  
석사 졸업  
1982년~1985년 한국전자통신  
연구소(ETRI) 연구원

1985년~1990년 Univ. of Texas at Austion  
전자공학 박사

1990년~1992년 Motorola, DSP Chip Divison  
(미국)

1992년~1996년 아주대학교 전기전자공학부  
조교수

1996년~2001년 아주대학교 전자공학부 부교수

2001년~현재 아주대학교 전자공학부 교수

2010년~현재 대한전자공학회 반도체 소사이어터  
수석 부회장

2009년~현재 IEEE CASS Distinguished  
Lecturer

2011년 IEEE CASS Board of Governor

<주관심분야 : SoC 설계, VLSI Architecture, 통  
신 및 멀티미디어 ASIP 설계, 저전력 설계>