

# 다중 참조영상 움직임 추정에 적용을 위한 연속 제거 알고리즘 기반 고속화 알고리즘

김영문<sup>†</sup>, 이재은<sup>\*\*</sup>, 임 찬<sup>\*\*\*</sup>, 강현수<sup>\*\*\*\*</sup>

## 요 약

본 논문에서는 다중 참조영상에 적용을 위한 고속 움직임 추정(motion estimation) 알고리즘을 제안한다. 우선 전역움직임추정방법(full search)의 고속 알고리즘인 연속 제거 알고리즘(successive elimination algorithm)을 다중 참조영상(multi-reference frames)에 적용하였을 때의 결과를 분석하고, 이를 바탕으로 계산량 감축 방안을 제안한다. 제안된 방법은 바로 이전 영상에 대한 움직임 벡터를 연속 제거 알고리즘을 적용하여 추출하고, 그 외의 참조영상에 대해서는 이전 영상에서 얻어진 움직임 벡터를 기반으로 외삽(extrapolation)을 수행함으로써 추정한다. 이 추정된 벡터를 중심으로 탐색범위를 적응적으로 제한함으로써 화질 저하를 최소화 시키면서 계산량을 크게 감축하도록 제안하였다. 실험을 통해 제안된 방법의 성능을 분석하고 검증하였다.

## Fast Algorithm Based on Successive Elimination Algorithm for Multi-Reference Motion Estimation

Young-Moon Kim<sup>†</sup>, Jae-Eun Lee<sup>\*\*</sup>, Chan Lim<sup>\*\*\*</sup>, Hyun-Soo Kang<sup>\*\*\*\*</sup>

## ABSTRACT

This paper presents a new fast motion estimation algorithm for multi-reference frames. We first analyze the experimental results of the successive elimination algorithm, which is a fast version of full search algorithm, being applied to Multi-reference frames. Based on the analysis, a new scheme for alleviating its computational burden is introduced. In the proposed method, the motion vector for the immediately previous reference frame is found by applying the successive elimination algorithm, while the motion vector for other reference frames is estimated by extrapolation of the already obtained motion vector. Adaptively restricting the motion search area to the local area centered on the estimated motion vector, the proposed method provides dramatic computational complexity reduction but slight quality degradation. The proposed method is evaluated by experiments for some image sequences.

**Key words:** Motion Estimation(움직임 추정), Fast Search Algorithm(고속 알고리즘), Successive Elimination Algorithm(연속제거 알고리즘)

\* 교신저자(Corresponding Author) : 강현수, 주소 : 충북 청주시 흥덕구 개신동 12(361-763), 전화 : 043)261-3488, FAX : 043)261-3488, E-mail : hskang@chungbuk.ac.kr  
접수일 : 2004년 10월 7일, 완료일 : 2005년 1월 26일

<sup>†</sup> 정회원, 중앙대학교 첨단영상 대학원 영상공학과 석사과정  
(E-mail : kimoon@wm.cau.ac.kr)

<sup>\*\*</sup> 준회원, 중앙대학교 첨단영상 대학원 영상공학과 석사과정

(E-mail : jlee@wm.cau.ac.kr)

<sup>\*\*\*</sup> 중앙대학교 첨단영상 대학원 영상공학과 석사과정  
(E-mail : mj23cb@wm.cau.ac.kr)

<sup>\*\*\*\*</sup> 충북대학교 전기전자컴퓨터공학부 조교수

\* 이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2004-003-D00372).

### 1. 서 론

비디오 신호에서는 높은 시간적, 공간적 중복성이 존재한다. 이를 제거함으로써 부호화 효율을 높일 수 있다. 움직임 추정(motion estimation: ME)은 비디오 신호의 시간 방향 잔여성분(redundancy)을 제거하는데 효과적이므로 H.261, H.263, H.264, MPEG-1, MPEG-2, MPEG-4 등의 비디오 압축 표준에서 뿐만 아니라 움직임 보상 부호화 기법이 적용된 비디오 부호화기에 널리 채용되고 있다. 그래서 좋은 복원 화질을 유지하면서, 좀 더 빠르고 정확한 움직임 벡터를 찾을 수 있는 방법이 요구된다. 전역탐색법(full search algorithm)은 최적의 움직임 벡터를 찾고 구현이 간단한 반면 많은 계산량이 요구된다. 그래서 계산량을 감축하기 위해, 많은 고속 알고리즘들이 제안되어 왔다. 예를들어, 2-D logarithmic 탐색법, 3단계 탐색법, conjugate direct 탐색법, cross 탐색법, 4단계 탐색법, diamond 탐색법 등이 있다[1-4]. 그러나 이러한 방법들은 화질의 저하가 따른다. 또한, 전역탐색법 자체에 대한 고속화 알고리즘 또한 활발히 연구되었는데, PDE(partial difference elimination algorithm), SEA(successive elimination algorithm), MSEA(multi-level SEA) 등이 대표적이다. PDE는 MPEG의 reference s/w에서 구현되어 있는 방식으로서, SAD(sum of absolute difference)의 계산과정 중 이전의 SAD를 초과하는 경우 SAD의 계산을 더 이상 수행하지 않는 방식이다[5-7]. SEA는 블록의 평균값으로부터 최적 벡터가 될 수 있는지의 여부를 판단함으로써 전체 블록에 대한 SAD의 계산이 불필요한 블록에 대한 계산을 수행하지 않음으로써 계산량을 감축하는 방법이다[8]. MSEA는 SEA의 다계층 접근방식으로 기본적인 아이디어는 동일하나 계산량을 크게 줄이는 반면 상대적으로 많은 메모리를 요구하는 단점이 있다[9,10]. 여기서 언급된 PDE, SEA, MSEA는 모두 전역탐색법과 동일한 성능을 보이면서 계산량을 줄이는 고속 전역탐색법이라고 할 수 있다.

본 논문에서는 SEA를 이용하여 다중 참조 영상에 적용(SEA-SM)하고, 탐색범위별로 그 성능을 분석한다. 또한 거리에 따른 탐색범위별로 성능을 분석하여, 그 거리에 따라 탐색영역의 범위를 다르게 함으로써 계산량을 감축하도록 하였다.

구성은 다음과 같다. 2장에서는 전역 탐색법(FS), 연속제거 알고리즘(SEA), 다중 참조 연속제거 알고리즘(SEA-SM)과 같은 기존 알고리즘을 간략하게 기술하였고, 3장에서는 계산량을 줄인 다중 참조 연속제거 알고리즘을 제안하고 실험을 통해 제안된 방법의 성능을 평가하였다. 마지막으로, 4장에서는 제안된 방법을 요약 기술하고 결론을 맺는다.

### 2. 기존 알고리즘

(1) 전역 탐색 알고리즘(full search algorithm)

움직임 벡터를 찾는 가장 간단한 방법은 탐색 범위 내의 모든 위치에 대하여 SAD를 구하고, 그 SAD 값이 최소가 되는 위치를 찾는 것이다. 이와 같은 방법을 전역탐색법이라고 한다. 전역탐색법은 최적의 움직임 벡터를 찾을 수 있지만, 많은 계산량이 요구된다는 단점을 가지고 있다. 탐색 영역 내에 모든 탐색점에 대해 다음 식을 적용함으로써 움직임 벡터를 찾게 된다.

$$SAD_{N \times N} = \sum_{i=1}^N \sum_{j=1}^N |original - previous|$$

(2) 연속 제거 알고리즘(successive elimination algorithm)

SEA는 최적의 움직임 벡터라면 기본적으로 만족해야 하는 부등식을 정의하고, 즉, 최적 움직임 벡터이기 위한 필요조건을 정의하고, 그 부등식을 만족하지 않는 탐색점들을 제외시킴으로써 계산량을 줄이는 방법이다. 수학적인 부등식은 다음과 같이 얻을 수 있다.

$$\begin{aligned} \sum_{i=1}^N \sum_{j=1}^N |f(i,j,t)| - \sum_{i=1}^N \sum_{j=1}^N |f(i-x,j-y,t-1)| \\ \leq \sum_{i=1}^N \sum_{j=1}^N |f(i,j,t) - f(i-x,j-y,t-1)| \end{aligned} \tag{1}$$

$$\begin{aligned} \sum_{i=1}^N \sum_{j=1}^N |f(i-x,j-y,t-1)| - \sum_{\substack{i=1 \\ i \neq 1, N}}^N \sum_{\substack{j=1 \\ j \neq 1, N}}^N |f(i,j,t)| \\ \leq \sum_{i=1}^N \sum_{j=1}^N |f(i,j,t) - f(i-x,j-y,t-1)| \end{aligned} \tag{2}$$

여기서  $x, y$ 는 움직임 벡터의  $x$ 방향 성분과  $y$ 방향 성분을 나타내며,  $x, y$ 의 범위는  $-M \leq x, y \leq M$ 이다. 식(1)에서 표현을 단순화하기 위하여, 첫 번째 항을 R, 두 번째 항은 M으로 표기하고, 우변은 움직임 벡터( $x, y$ )에 대한 SAD 값을  $SAD(x, y)$ 로 표기한다.

이와 같이 표기를 변경하면 식(1)과 (2)는 다음과 같이 표현 할 수 있다.

$$R - M(x, y) \leq SAD(x, y) \quad (3)$$

$$M(x, y) - R \leq SAD(x, y) \quad (4)$$

현재까지 고려한 탐색점들 중 최소의 오차를 가지는 탐색점, 즉 움직임 벡터를  $(m, n)$ 이라고 하고 이의 오차  $SAD(m, n)$ 를 알고 있다고 가정하자. 여기서  $SAD(m, n)$ 보다  $SAD(x, y)$ 가 더 작은 값을 가질 때, 더 나은 움직임 벡터를 얻을 수 있게 된다.

$$SAD(x, y) \leq SAD(m, n) \quad (5)$$

여기서, 식(5)를 (3)과 (4)에 대응시키면 다음과 같다.

$$R - M(x, y) \leq SAD(m, n) \quad (6)$$

$$M(x, y) - R \leq SAD(m, n) \quad (7)$$

위 수식은 다음과 같이 정리 할 수 있다.

$$R - SAD(m, n) \leq M(x, y) \leq R + SAD(m, n) \quad (8)$$

식(8)은 SEA에서 사용되는 중요한 결과이다. 이 수식은  $(x, y)$ 가  $R - SAD(m, n)$ 와  $R + SAD(m, n)$ 의 사이에 존재 할 때만이 최적의 움직임 벡터가 될 수 있음을 의미한다. 따라서, 이 범위를 벗어나는 탐색 블록은 움직임 추정에서 제외시킬 수 있다.

이 알고리즘을 H.264에서 사용되는 다중참조영상에 적용하는 방법을 생각해 볼 수 있다. 예를 들어, 이전영상( $t-1$ )과 이전 영상의 이전영상 ( $t-2$ )에서 움직임 벡터를 찾는 경우, 이 알고리즘을 순차적으로 적용할 수 있다. 이처럼 이전영상( $t-1$ )과 이전 이전영상( $t-2$ )에 순차적으로 적용하는 방법( $t-1$ ) 영상의 탐색영역과 ( $t-2$ ) 영상의 탐색영역이 같고, 두 장의 영상 모두를 탐색하는 방법을 SEA-SM (Straightforward Method)이라고 부르자. SEA-SM은 참조영상의 개수에 거의 정비례하여 계산량이 증가한다. 그러나, 이전 참조 영상에서 찾은 움직임 벡터의 정보를 이용한다면, 좀 더 효과적으로 그 외의 참조 영상에 대한 움직임 벡터를 찾을 수 있을 것이다. 본 논문은 이점에 착안하여 고속 알고리즘을 제안한다.

### 3. 제안된 알고리즘

본 장에서는 SEA를 다중참조 영상에 적용 했을 때, 계산량을 감축시킬 수 있는 방법을 제안한다. 다중참조알고리즘은 이전영상( $t-1$ )만이 아닌 둘 이상의 영상에서 움직임 벡터를 찾을 수 있지만,  $t-3$  이상의 영상을 이용하게 된다면 계산량에 비해 화질의 상승이 크지 않아 본 논문에서는 영상의 범위를  $t-2$  영상까지로 제한하였다.

최적의 움직임 벡터를 검출하는 가장 일반적인 방법은 바로 이전영상( $t-1$ )에 대해서 SAD를 구하여, 최소의 SAD 값을 가진 블록을 움직임 벡터로 취하는 것이다. 여기서 탐색 영상의 영역을  $t-2$ 까지 확장하게 된다면, 보다 최적화된 움직임 벡터를 찾을 수 있지만 계산량은 그만큼 증가한다. 제안 방법은  $t-1$  영상과 함께  $t-2$ 영상도 탐색 영역에 포함시켰을 때, 계산량을 감축시키는 방법이다. 제안 방법은  $t-2$  영상에서 최적의 움직임 벡터가 나올 수 있는 범위에 대한 분석에 따라 적응적으로 탐색 범위를 제한하여 계산량을 줄인다.

#### 3.1 탐색범위에 따른 추정 오차

본 소절에서는  $t-2$  영상의 탐색범위를 제한하여 탐색범위별 오차를 분석하고자한다. 그림 1은  $t-2$  영상의 탐색 중심점을 구하여, 그 중심점으로부터 나선형으로 탐색하는 방법이다. 이 방법은  $t-1$  영상에서 SEA 그대로 적용하여 최적의 움직임 벡터를 구하고,  $t-2$  영상에서는  $t-1$  영상에서 구한 움직임 벡터를 이용하여 중심점을 구하는 방법이다. 강체 (rigid body) 병진운동의 가정하에,  $t-2$  영상에서의 움직임벡터는  $t-1$  영상에서의 움직임 벡터를 2배로

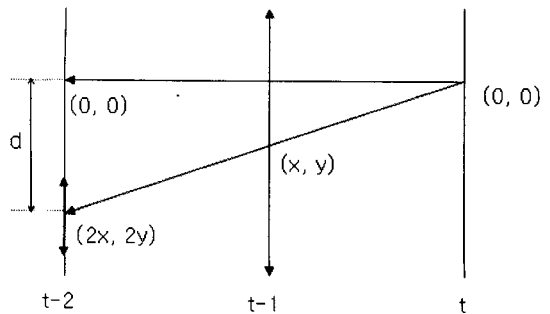


그림 1. 다중 참조를 이용한 움직임 추정

하여 정하였다. 이렇게 정해진 움직임 벡터를 중심으로 하여 탐색범위에 따른 PSNR을 구하였다. 그림 2, 3, 4는 SEA-SM과 탐색범위를 제한하여 실험한 결과이다. 여기서 탐색범위가 증가할 수록 PSNR은 SEA-SM으로 수렴하는 것을 볼 수 있다. 특히 mobile의 경우 탐색범위  $\pm 1$ 에서 SEA-SM과 PSNR이 같아지는 것을 볼 수 있다. 이러한 실험결과로 볼 때,  $t-2$  영상에 대한 탐색범위를 제한하여도 우수한 화질을 유지할 수 있음을 알 수 있다. 또한, 일반적으로  $t-1$  영상에서 최적의 움직임 벡터가 나올 가능성이 많으므로  $t-2$  영상에 대한 움직임 추정에 같은 계산량을 부여하는 것은 바람직하지 못하다.

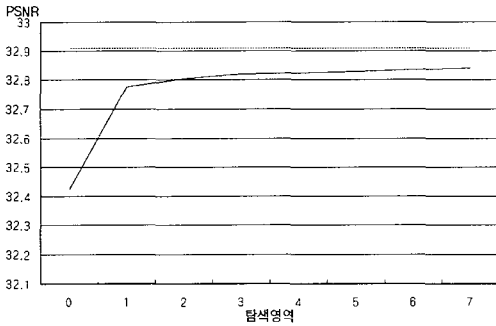


그림 2. 탐색범위에 따른 추정 오차 (foreman)

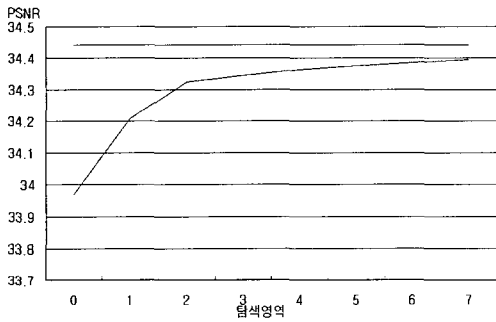


그림 3. 탐색범위에 따른 추정 오차 (carphone)

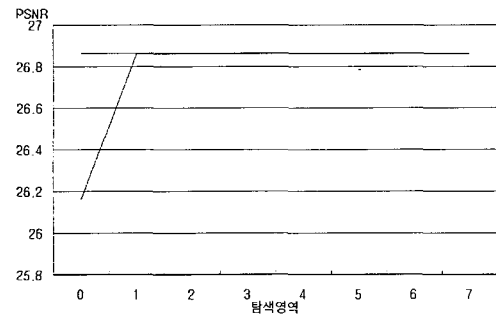


그림 4. 탐색범위에 따른 추정 오차 (mobile)

### 3.2 거리와 탐색범위에 따른 추정 오차

본 소절에서는  $t-2$  영상의 거리에 따른 탐색범위별 오차를 보일 것이다.  $t-2$  영상의 중심점( $2x, 2y$ )과 원점( $0,0$ )과의 거리( $d$ )에 따라 탐색범위별로 SAD 값을 조사하였다. 이것은 거리( $d$ )에 따라 탐색범위를 결정하는 중요한 잣대가 된다. (그림 5, 6, 7)에서,  $d=0$  일 때, 탐색범위  $\pm 1$ 까지는 SAD 값이 급격히 하향곡선을 그리는 것을 볼 수 있고, 이후 탐색범위에서는 SAD 값의 변동이 거의 없는 것을 볼 수 있다.  $0 < d \leq 2$  일 때, 탐색범위  $\pm 2$ 까지 급격한 하향곡선을 그리고 이후는 변동이 거의 없다.  $2 < d \leq 3$  일 때,  $\pm 3$ 까지 하향곡선을 그리고 이후에는 변동이 거의 없는 것을 알 수 있다.  $d \geq 4$  일 때, 지속적인 하향곡선을 그리고 있다. 이러한 분석으로 보아 각각의 거리( $d$ )에 따라 탐색범위( $\pm 1, \pm 2, \pm 3, \dots$ )으로 제한을 한다하더라도 최적의 움직임 벡터를 찾으면서 화질에는 큰 변화가 없음을 알 수 있다. 또한 표 1을 보면 그 거리가 가까운 쪽에 많이 분포되어 있다. 이것은 탐색범위를 제한하여 탐색하는 비율이 높다는 것을 의미하여 많은 계산량을 줄일 수 있음을 의미한다.

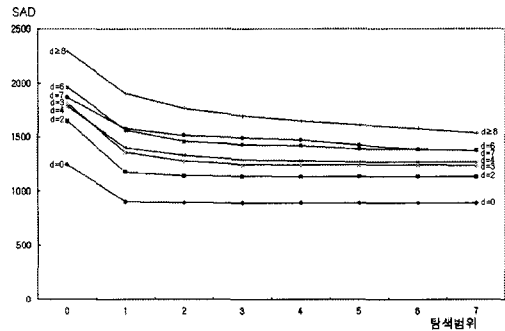


그림 5. 거리에 따른 탐색범위별 추정 오차 (Foreman)

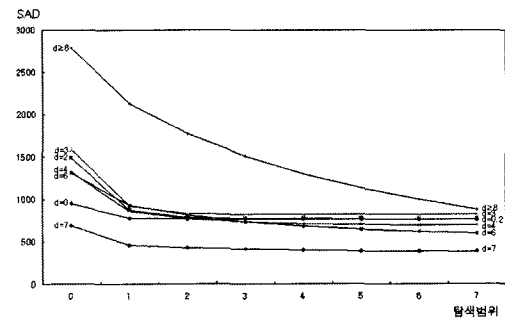


그림 6. 거리에 따른 탐색범위별 추정 오차 (Carphone)

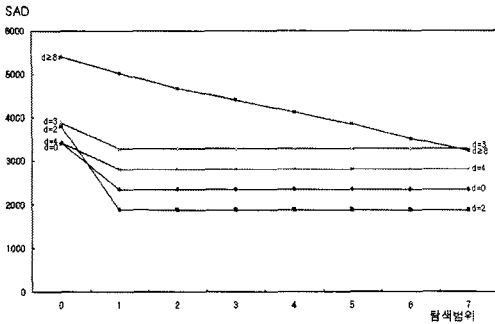


그림 7. 거리에 따른 탐색범위별 추정 오차 (Mobile)

표 1. 거리(d)의 비율

거리(d)	foreman	carphone	mobile
d=0	48.8%	52.6%	91.7%
0<d≤2	28.8%	24.4%	7.3%
2<d≤3	7.7%	7.8%	0.2%
3<d≤4	5.6%	6.8%	0.4%
4<d≤6	2.9%	2.6%	0%
6<d≤7	0.9%	0.2%	0%
d>=8	6.3%	5.5%	0.4%

### 3.3 다중 참조 연속 제거 알고리즘

3.1과 3.2에서 살펴본 탐색범위에 따른 추정 오차와 거리와 탐색범위에 따른 추정 오차의 결과로부터, 다음과 같은 알고리즘을 제안한다.

①  $t-1$  영상에 대해서는 SEA를 그대로 적용한다. 여기서  $t-1$  영상의 움직임 벡터  $(x, y)_{t-1}$ 와  $SAD(x, y)_{t-1}$ 를 구할 수 있다.

②  $t-2$  영상에 대해서는 움직임 벡터  $(0, 0)_{t-2}$ 의  $SAD(0, 0)_{t-2}$ 를 우선적으로 구한다.

③  $t-2$  영상에서 탐색 범위의 중심은  $t$  영상의 원점(0,0)과  $t-1$  영상의 움직임 벡터  $(x, y)_{t-1}$ 를 직선으로 연결하여  $t-2$  영상과 교차되는 지점으로 설정한다.

④  $t-2$  영상에서 탐색 범위는 중심점으로부터 최적의 움직임 벡터가 나올 확률이 높은 영역으로 그 범위를 제한한다. 이 영역의 범위는 그림 5, 6, 7과 같은 분석 결과에 따라 적용적으로 정해질 수 있다. 여기서 분석 결과는  $d=0$  일 경우, 탐색범위를  $\pm 1$ ,  $0 < d \leq 2$  일때  $\pm 2$ ,  $2 < d \leq 4$  일때  $\pm 3$ ,  $4 < d$ 일때, 탐색범위 전체를 탐색한다.

⑤ 설정된 범위에 대해  $(x, y)_{t-2}$ 와  $SAD(x, y)_{t-2}$ 를 구한다.

제안된 알고리즘은  $t-2$  영상에서의 움직임 벡터를 구할 때, 과정 ②를 통해 초기에 낮은 SAD 값을 취하여 계산량을 감축하도록 하였으며, 과정 ③, ④, ⑤를 통하여 탐색범위에 제한을 두어 계산량을 감축하도록 하였다. 여기서 탐색범위는 3.2에서 보인 거리와 탐색범위에 따른 추정 오차에 거리(d)에 따라 탐색범위를 제한하여 최적의 움직임 벡터가 나올 확률이 높은 범위로 한정되며, 다음 장의 실험 결과로 확인할 수 있다.

실험을 통하여  $t-2$  영상에서 최적 움직임 벡터는 (0,0)과  $t$  영상의 원점과  $t-1$  영상의 움직임 벡터를 연결하는 직선을 중심으로 하는 좁은 영역 내에 있을 확률이 높다는 것을 확인하였다. 이는 계산량을 현격히 감축시키면서, 비슷한 움직임 오차를 얻을 수 있다는 것을 의미한다. 또한  $t-2$ 에 의해서 움직임 벡터를 찾을 때 필요한 sum norm은 이미  $t-1$ 에서 움직임 벡터를 찾을 때 계산하였으므로 추가적인 계산은 요구되지 않는 장점 또한 있다.

### 4. 실험 결과

탐색범위는  $\pm 16$ 으로 하였으며, 제안된 방식을 사용하여  $t-2$  영상에서 탐색범위를 제한하고 움직임 벡터를 구하였다. 그리고 FS, SEA, SEA-SM과 제안된 방식의 결과를 비교분석하였다.

실험 조건은 다음과 같다. 영상은 크기는  $176 \times 144$ 의 "foreman", "carphone", "mobile", "table"과  $352 \times 288$ 인 "mother\_daughter"의 영상을 사용하였고, 탐색범위는  $\pm 16$ pixel(M=16)이며 정수 단위 움직임 벡터만을 사용하였다. 그리고 총 100 프레임에 대해 실험을 수행하였다. 움직임 벡터를 구하기 위한 평가함수는 SAD를 사용하였으며, 결과 비교를 위한 결과 영상의 평가는 PSNR(Peak Signal-to-Noise Ratio)을 사용하였다.

그림 9, 11, 13, 15는 SEA와 SEA-SM 그리고 제안된 알고리즘의 PSNR을 프레임별로 비교하여 그래프로 나타낸 것이다. 여기서 제안된 방식과 SEA-SM의 PSNR이 거의 동일한 곡선이 이어지는 것을 확인할 수 있다. 그림 10, 12, 14, 16은 SEA, SEA-SM, 제안된 방식의 계산량(블록당 SAD 계산 횟수)을 프레임

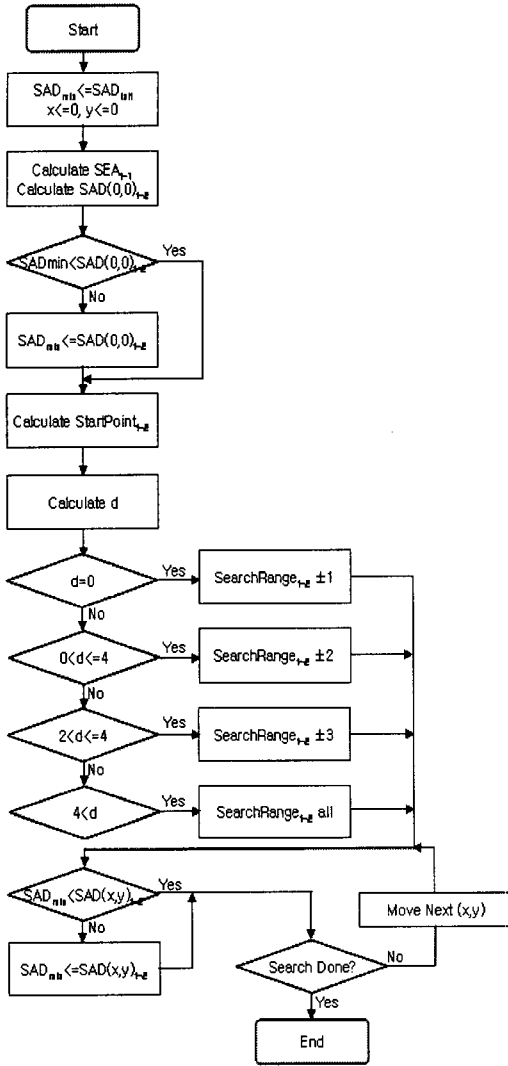


그림 8. 제안 알고리즘의 흐름도

별로 비교하여 그래프로 나타낸 것이다. 이 그래프에서 볼때, 기존방식(SEA-SM)보다 제안된 방식의 계산량이 많이 감축된 것을 확인 할 수 있다. 표 2는 FS, SEA, SEA-SM, 제안된방법의 PSNR과 계산량을 수치로 나타내고 있다. 비록 두장의 참조영상에 대해 제안된방법을 적용하였지만, 한 장의 참조영상에 적용한 SEA와 거의 비슷한 수준의 계산량을 유지하면서도, PSNR은 SEA-SM과 거의 비슷한 수준을 보이고 있음을 확인 할 수 있다.

(1) Foreman QCIF 30Hz

Foreman 영상의 경우, 제안된 방법의 PSNR이

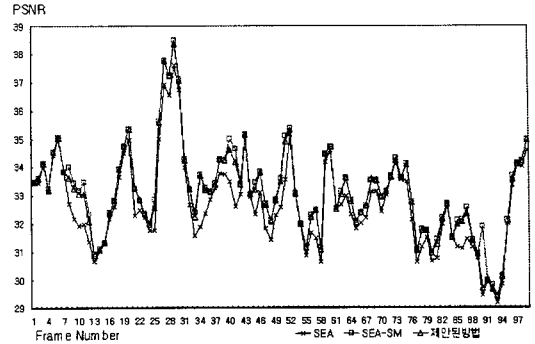


그림 9. 기존방식과 제안방식의 프레임별 PSNR 비교 (Foreman)

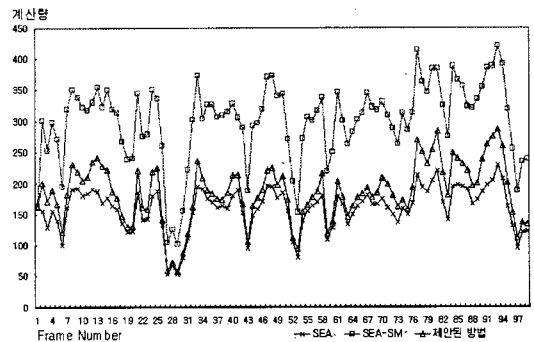


그림 10. 기존방식과 제안방식의 프레임별 계산량 비교 (Foreman)

SEA-SM과 거의 동일하고, 계산량은 SEA-SM 보다 약 39% 감축되었으며 SEA와 거의 비슷한 것을 볼 수 있다. 여기서 SEA는 한 장의 참조영상을 사용한 경우를 말한다.

(2) Carphone QCIF 30Hz

Carphone 영상의 실험결과는 기존방식과 제안된 방식의 PSNR이 거의 동일하고, 계산량은 약 40% 감축된 것을 확인 할 수 있다.

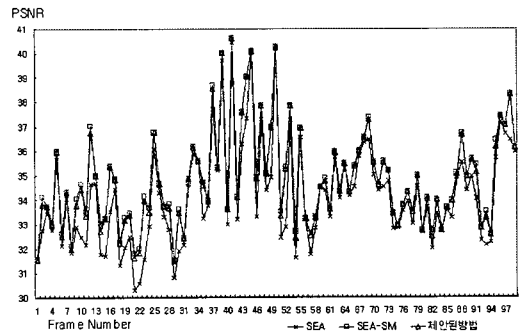


그림 11. 기존방식과 제안방식의 프레임별 PSNR 비교 (Carphone)

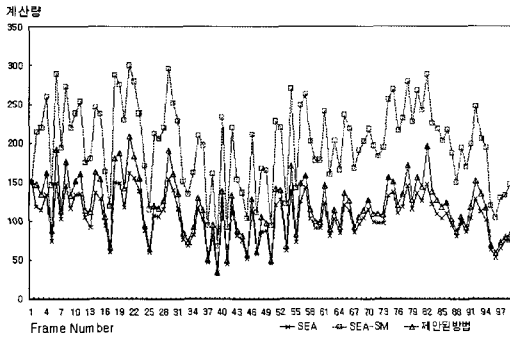


그림 12. 기존방식과 제안방식의 프레임별 계산량 비교 (Carphone)

(3) Mobile QCIF 30Hz

Mobile 영상의 실험결과로 보아 PSNR은 몇몇 프레임 제외하고 나머지가 SEA-SM과 완전히 동일하고, 계산량을 볼때도 SEA-SM 보다 약 47%가 감축된 것을 볼 수 있다.

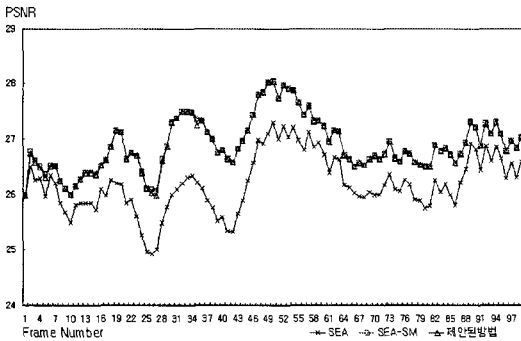


그림 13. 기존방식과 제안방식의 프레임별 PSNR 비교 (Mobile QCIF)

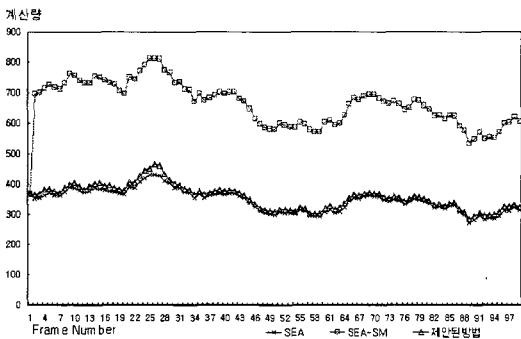


그림 14. 기존방식과 제안방식의 프레임별 계산량 비교 (Mobile QCIF)

(4) Table QCIF 30Hz

Table 영상의 실험결과이다. 이 영상 또한 SEA-SM과 제안된 방식의 PSNR이 거의 동일한 반면 계

산량은 감축되면서 계산량이 SEA에 접근하는 것을 볼 수 있다. 제안된 방식이 SEA-SM보다 약 41% 감축된 것을 확인 할 수 있다.

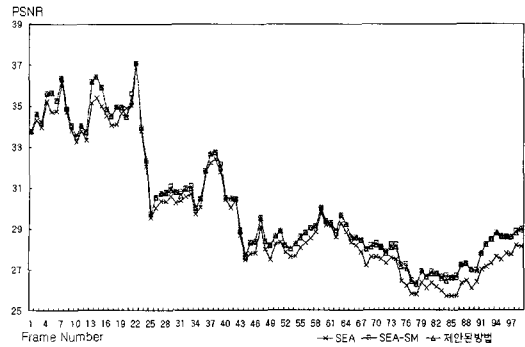


그림 15. 기존방식과 제안방식의 프레임별 PSNR 비교 (Table)

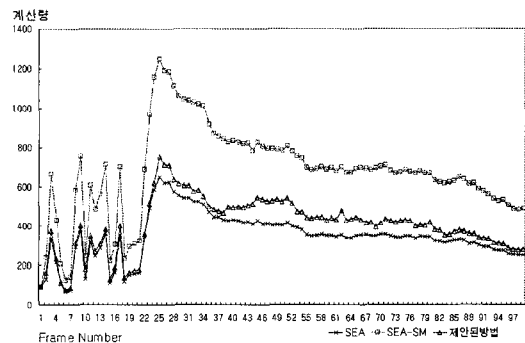


그림 16. 기존방식과 제안방식의 프레임별 계산량 비교 (Table)

표 2. 기존방식과 제안된 방식의 계산량과 PSNR 비교

Image	Algorithms	계산량	PSNR(dB)
foreman (QCIF)	FS	1089	32.409
	SEA	159	32.409
	SEA-SM	301	32.909
	제안된방법	185	32.893
carphone (QCIF)	FS	1089	33.705
	SEA	105	33.705
	SEA-SM	199	34.442
	제안된방법	120	34.431
mobile (QCIF)	FS	1089	26.154
	SEA	346	26.154
	SEA-SM	665	26.865
	제안된방법	357	26.865
table (QCIF)	FS	1089	28.855
	SEA	349	28.855
	SEA-SM	681	29.447
	제안된방법	407	29.406

\*계산량은 블록당 SAD 계산 횟수.

### 5. 결론 및 추후 연구

본 논문에서는 다중 참조영상에 적용 할 수 있는 연속 제거 알고리즘 기반의 움직임 추정 알고리즘을 제안하였다. 제안된 방법은 탐색 범위에 따른 움직임 오차 분석 결과를 기초로 하여 탐색 범위를 제한함으로써 계산량을 크게 감축한다. 실험 결과에 따르면, 제안된 방법이 계산량 감축에 매우 효과적이며, 화질 측면에서 기존의 방식과 거의 동일한 수준의 결과를 얻을 수 있었다. 본 논문에서는 연속 제거 알고리즘 기반의 알고리즘을 제안하였지만 다단계 연속 제거 알고리즘에 대해서는 제안된 방법이 적용될 수 있고 보다 나은 성능 향상이 기대된다.

### 참고 문헌

[1] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proceedings of the IEEE*, Vol. 83, pp. 858-879, 1995.

[2] L.M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. on Circuits Syst. Video Technology*, Vol. 6, pp. 313-317, 1996.

[3] L.K.Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Trans. on Circuits Syst. Video Technology*, Vol. 6, pp. 419-423, 1996.

[4] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block matching motion estimation," in *Proc. Int. Conf. Inform., Comm., Signal Processing*, pp. 292-296, 1997.

[5] S. Eckart and C. Fogg, "ISO/IEC MPEG-2 software video codec," *Proc. SPIE*, Vol. 2419, pp. 100-118, 1995.

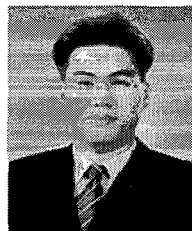
[6] "ITU-T recommendation H.263 software implementation," *Digital Video Coding Group, Telenor R&D*, 1995.

[7] J. N. Kim and T. S. Choi, "A fast full-search motion-estimation algorithm using representative pixels and adaptive matching scan," *IEEE Trans. on Circuits Syst. Video Technology*, Vol. 10, No. 7, pp. 1040-1048, 2000.

[8] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. on Image Processing*, Vol. 4, No. 1, pp. 105-107, 1995.

[9] X. Q. Gao, C. J. Duanmu, and C. R. Zou, "A multilevel successive elimination algorithm for block matching motion estimation," *IEEE Trans. on Image Processing*, Vol. 9, No. 3, pp. 501-504, 2000.

[10] J. Y. Lu, K. S. Wu, and J. C. Lin, "Fast full search in motion estimation by hierarchical use of Minkowski's inequality," *Pattern Recognition*, Vol. 31, No. 7, pp. 945-952, 1998.



김 영 문

2003년 중앙대학교 컴퓨터공학과 졸업(공학사)  
 2003년 현재 중앙대학교 첨단영상 대학원 영상공학과 석사과정

관심분야: 영상처리, 영상통신, 신호처리



이 재 은

2001년 중앙대학교 기계공학부 졸업(공학사)  
 2001년~2003년 LG CNS 연구원  
 2003년 현재 중앙대학교 첨단영상 대학원 영상공학과 석사과정

관심분야: 사운드, 영상처리, 영상통신, 신호처리





임 찬

2000년 아주대학교 전자공학과  
졸업(공학사)  
2000년~2002년 LG전자 연구원  
2003년 현재 중앙대학교 첨단영  
상 대학원 영상공학과 석  
사과정

관심분야 : 영상처리, 영상통신, 신호처리



강 현 수

1999년 한국과학기술원 전기및  
전자공학과 졸업 (공학  
박사)  
1995년 하이닉스반도체(주) 선임  
연구원  
2001년 한국전자통신연구원 선  
임 연구원

2002년 중앙대학교 첨단영상 대학원 영상공학과 조교수  
2005년 충북대학교 전기전자컴퓨터공학부 조교수  
관심분야 : 영상처리, 부호화, 콘텐츠보호기술, 사운드 등