

논문 2006-43SP-1-1

다중 참조 영상 움직임 추정을 위한 고속 전역탐색법

(A fast full search algorithm for multiple reference image motion estimation)

강 현 수*, 박 성 모**

(Hyun-Soo Kang and Seong-Mo Park)

요 약

본 논문은 다중 참조영상(multiple reference image)에 적용 가능한 새로운 고속 전역탐색 움직임 추정 기법을 제안한다. 제안된 방법은 기존의 비트율을 고려한 연속제거알고리즘(rate constrained successive elimination algorithm: RSEA)을 다중 참조영상에 확대 적용하는 방법이다. 첫 번째 참조영상에 대한 움직임 추정의 계산량에 비해 그 이후 참조영상에 대한 움직임 추정의 계산량이 적어질 수 있음을 보일 것이다. 계산량 감축을 위해, 본 논문에서 최적 움직임 벡터의 후보 블록의 개수를 감소시킬 수 있는 새로운 조건을 소개한다. 실험 결과를 통해 제안된 방법이 기존의 RSEA와 동일한 움직임 추정오차를 가지면서도 계산량을 감소시킴을 보일 것이다.

Abstract

This paper presents a fast full search algorithm for motion estimation applicable to multiple reference images. The proposed method is an extended version of the rate constrained successive elimination algorithm (RSEA) for multiple reference frame applications. We will show that motion estimation for the reference images temporally preceding the first reference image can be less intensive in computation compared with that for the first reference image. For computational reduction, we will drive a new condition to lead the smaller number of candidate blocks for the best matched block. Simulation results explain that our method reduces computation complexity although it has the same quality as RSEA.

Keywords: Motion estimation, Multiple reference image, Successive elimination algorithm, Image coding

I. 서 론

움직임추정은 비디오 신호의 시간방향 중복성분을 효과적으로 제거함에 따라, 비디오 부호화 시스템에서 널리 채용되고 있는 핵심적인 기술이다. 비록 움직임추정 기술의 연구가 오랫동안 지속되어 왔지만 보다 정확하고 빠른 움직임 벡터의 탐색에 대한 연구는 현재까지도 그 요구가 지속되고 있다. 움직임 추정 알고리즘 중 전역탐색법(full search algorithm)은 화질 측면에서 최적

의 움직임 벡터를 찾아주지만 많은 계산량이 요구된다 는 문제점을 안고 있다. 이러한 계산량 문제를 해결하기 위하여 3단계 탐색, 다이아몬드 탐색, 계층적 탐색 등의 많은 고속 알고리즘이 제안되어 오고 있다^{[1]-[4]}.

한편, 전역탐색법의 움직임 추정오차와 동일한 결과를 주면서 계산량을 감축하는 알고리즘들이 개발되었다. PDE(partial difference elimination algorithm), SEA (successive elimination algorithm), MSEA(multi-level SEA) 등이 이 부류의 알고리즘에 속한다. 그들 중, 본 논문과 밀접하게 관련된 알고리즘은 SEA로서, 우선 탐색점이 최적 움직임 벡터가 될 수 있는지에 대한 검사를 수행하고 이 검사를 통해 최적 움직임 벡터가 될 수 없는 탐색점을 제외시킴으로써 계산량을 감축시키는 알고리즘이다^[5]. MSEA는 SEA의 다단계 접근 방식으로

* 정회원, 충북대학교 전기전자컴퓨터공학부

** 정회원, 한국전자통신연구원 기반기술연구소

※ 이 논문은 2005년도 충북대학교 학술연구지원사업의 연구비지원에 의하여 연구되었음

접수일자: 2005년10월12일 수정완료일: 2005년11월14일

서 저해상도에서 고해상도로 블록의 해상도를 변화시키면서 SEA를 적용한 것으로써, 그 원리는 동일하다^{[7][8]}. 또한 비용함수에 움직임 추정 오차 뿐만 아니라 비트율까지 고려한 SEA 방식(rate-constrained SEA: RSEA)이 [6]에서 제안되었다.

본 논문에서는 H.264^[9]에서 채용하고 있는 다중 참조 영상에 적용을 위한 움직임 추정 알고리즘으로 사용할 수 있는 새로운 SEA를 제안하고자 한다. 제안된 방법은 현재 영상의 바로 이전 참조영상(편의상 첫 번째 참조영상이라고 부름)에 적용하기 위한 것이 아니라 첫 번째 참조영상을 제외한 나머지 참조영상(두 번째, 세 번째 등의 참조영상)에 적용하기 위한 방법이다. 첫 번째 참조영상에서의 움직임 추정 과정에서 얻어진 데이터를 이용하여 다른 참조영상에서의 움직임 추정시 활용함으로써 움직임 추정의 계산량을 감축할 수 있음을 이론적으로 보이고 실험을 통해 검증한다.

II. 배경

제안된 방법을 설명하기 위하여, 전역탐색법의 고속 알고리즘인 SEA를 우선 소개한다. 편의를 위해, 블록크기를 $N \times N$, 탐색창(search window)의 크기를 $(2M+1) \times (2M+1)$, 그리고 시간 t 에서의 프레임(frame)의 위치 (p, q) 에서의 화소값을 $f(p, q, t)$ 라고 하자. 이때, 참고문헌^[5]에서 기술된 바와 같이, 탐색 계산량을 감축시키기 위한 SEA의 검사조건은 아래 수식과 같다.

$$||\mathbf{M}| - |\mathbf{R}|| \leq SAD(m, n) \quad (1)$$

여기서 \mathbf{R} 과 \mathbf{M} 은 현재 블록과 참조블록에 해당하는 N^2 차원의 열벡터로서, 각 열의 성분은 $r_{iN+j} = f(p+i, q+j, t)$, $0 \leq i, j \leq N$ 와 $m_{iN+j} = f(p+i-x, q+j-y, t-1)$, $0 \leq i, j \leq N$ and $-M < x, y < M$ 이다. 그리고, $|\cdot|$ 는 벡터의 sum norm이다. 즉, $|\mathbf{R}|$ 과 $|\mathbf{M}|$ 은 구체적으로 다음과 같이 표현될 수 있다.

$$|\mathbf{R}| = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |f(p+i, q+j, t)| \quad (2)$$

$$|\mathbf{M}| = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |f(p+i-x, q+j-y, t-1)| \quad (3)$$

그리고, 식(1)에 나타난 SAD는 다음과 같이 표기할 수 있다.

$$SAD(m, n) = |\mathbf{R} - \mathbf{O}_{t-1}| \quad (4)$$

여기서 \mathbf{O}_{t-1} 은 움직임 벡터 (m, n) 이 가리키는 블록에 해당되는 N^2 차원의 열벡터로서 각 성분은 $o_{iN+j} = f(p+i-m, q+j-n, t-1)$ 이다.

우선, 움직임 벡터 (m, n) 를 현재까지의 탐색점들 중 최소 SAD ($SAD(m, n)$)를 가진다고 가정하였을 때, 현재 탐색점 (x, y) 가 더 작은 SAD를 가진다면 식(1)을 만족한다. 만약 식(1)을 만족하게 되면 (m, n) 는 (x, y) 로 갱신되며 다음 탐색점들은 (x, y) 에서의 SAD와 비교하는 과정을 반복하게 된다. SEA는 최적 움직임 벡터가 되기 위한 필요조건 식(1)을 이용하여 식(1)을 만족하지 않는 탐색점들에 대한 SAD 연산을 수행하지 않음으로써 계산량 감축을 실현하게 된다. 실험에 의하면 식(1)을 만족하지 못하여 연산이 생략되는 경우가 평균적으로 50%~70% 정도이다.

이해를 돕기 위해, 지금까지 움직임 벡터의 부호를 위한 비트량의 고려없이 움직임 추정 오차만을 고려한 SEA에 대해 설명하였지만, SEA 수행시 비트량에 대한 비용을 포함시킬 수 있다. 참고문헌 [6]에서 라그랑지안 계수 (lagrangian multiplier)를 도입한 비용함수(cost function)을 도입함으로써 비트량까지 고려한 알고리즘으로 확장하였다.

$$J(x, y, \lambda) = SAD(x, y) + \lambda r(x, y) \quad (5)$$

여기서는 λ 는 라그랑지안 계수, (x, y) 와 $r(x, y)$ 는 움직임 벡터와 그 움직임 벡터를 부호화하는데 소요되는 비트량이다. 비용함수를 SAD대신 J 를 사용함으로써 검사조건 식(1)은 다음과 같이 일반화될 수 있다.

$$||\mathbf{M}| - |\mathbf{R}|| + \lambda r(p, q) \leq SAD(m, n) + \lambda r(m, n) \quad (6)$$

III. 제안된 방법

1. 알고리즘 기술

본 절에서는 RSEA^[6]의 유도과정과 유사한 과정으로 최적 벡터가 만족해야 하는 새로운 검사조건을 유도하고 이를 이용한 제안된 알고리즘을 소개한다. H.264와 같이 다중 참조영상을 사용하는 비디오 시스템에서 움

직임 벡터 추정은 참조영상별로 최적 움직임 벡터를 추정하고 그 중 가장 최적의 움직임 벡터로 결정한다. 그리고 최적 벡터와 참조된 영상의 인덱스를 부호화하게 된다. 그러나 이와 같이 단순 확장형태의 방법보다 효과적인 방법이 존재할 수 있고 필요하다. 그래서 본 논문에서는 단순 확장에 의한 방법보다 계산량 측면에서 효율적인 다중 참조 영상을 위한 RSEA를 제안한다.

설명을 간편히 하기 위해서, 두장의 참조영상에 대한 움직임 벡터 추정을 중심으로 설명한다. 두 번째 참조영상, 즉 시간적으로 $t-2$ 에 위치한 프레임(두 번째 참조영상)의 (x, y) 에 위치한 블록을 열벡터 \mathbf{P} 로 나타내자. 여기서 \mathbf{P} 의 각 성분은 $p_{iN+j} = f(p+i-x, q+j-y, t-2)$, $0 \leq i, j < N$ 으로 나타난다.

제안된 알고리즘을 설명하기 전에, 제안된 방법의 우수성을 입증하기 위하여 참조 알고리즘으로서 단순 확장에 의한 방법(단순확장법)에 대해 먼저 소개한다. 단순확장법은 아래와 같은 순서에 의해 수행된다고 하자.

- (1) 첫 번째 참조영상($t-1$ 프레임)에 대해 RSEA를 적용하고 최적 움직임 벡터 (m^*, n^*) 를 얻는다.
- (2) 같은 방식으로 두 번째 참조영상($t-2$ 프레임)에 대해 RSEA를 적용한다. 다만, 첫 번째 참조영상에 적용한 RSEA와의 차이점은 두 번째 참조영상에 대한 탐색과정에서 초기의 최소 비용을 첫 번째 참조영상의 움직임 탐색과정에서 얻은 최적 벡터 (m^*, n^*) 를 이용하여 얻는다는 점이다. 즉, 초기 최소 비용을 $J(m^*, n^*, \lambda)$ 로 두는 것이다. 이 초기 비용값은 식(6)의 우변의 $(x, y) = (m^*, n^*)$ 인 경우에 해당된다. 여기서 사용된 라그랑지안 계수는 참고문헌^[6]에서 주어진 값을 사용한다.

Minkowski의 부등식 $(\|\mathbf{A}\| - \|\mathbf{B}\| \leq \|\mathbf{A} - \mathbf{B}\|)$ 으로부터 다음과 같은 관계식을 얻을 수 있다.

$$\| \|\mathbf{P} - \mathbf{M}\| - \|\mathbf{M} - \mathbf{R}\| \| \leq \|\mathbf{P} - \mathbf{R}\| \quad (7)$$

여기서 $\|\mathbf{P} - \mathbf{R}\|$ 은 현재 프레임의 블록과 두 번째 참조영상의 (x, y) 에 위치한 블록과의 SAD에 해당된다. RSEA에서의 유도과정과 유사하게, 두 번째 참조영상에 대한 움직임 벡터 (m, n) 에 대한 초기 최소 비용을 $J_2(m, n, \lambda)$ 라고 하자. 이 때 $J_2(m, n, \lambda)$ 는 첫 번째 참조영상의 최적 벡터에 대한 비용을 고려하여 최소 비용이 되도록 다음과 같이 정의한다.

$$J_2(m, n, \lambda) = \min\{ \|\mathbf{R} - \mathbf{Q}_{t-2}\| - \lambda r(m, n, t-2), J(m^*, n^*, \lambda) \} \quad (8)$$

현재 위치 (x, y) 가 보다 나은 정합 후보(matching candidate)가 되기 위해서는 현재 위치의 비용이 이전 최소 비용에 비해 작아야 한다. 즉, 아래 조건을 만족하여야 한다.

$$\|\mathbf{P} - \mathbf{R}\| + \lambda r(x, y, t-2) \leq J_2(m, n, \lambda) \quad (9)$$

결국, 식(7)과 식(9)를 이용하면 다음과 같은 조건을 얻게 된다.

$$\| \|\mathbf{P} - \mathbf{M}\| - \|\mathbf{M} - \mathbf{R}\| \| + \lambda r(x, y, t-2) \leq J_2(m, n, \lambda) \quad (10)$$

식(10)은 제안된 방법의 핵심 아이디어이다. 식(10)에서 $\|\mathbf{M} - \mathbf{R}\|$ 은 현재 블록의 열벡터 \mathbf{R} 과 첫 번째 참조영상의 참조블록의 열벡터 \mathbf{M} 사이의 sum norm (블록 간 SAD)에 해당되고, $\|\mathbf{P} - \mathbf{M}\|$ 은 $(p+i-x, q+j-y)$ 의 동일 위치에 위치한 첫 번째 참조영상의 블록 열벡터 \mathbf{M} 와 두 번째 참조영상의 블록 \mathbf{P} 사이의 SAD에 해당된다.

식(10)과 더불어 기존의 RSEA 방식의 검사조건을 두 번째 참조영상에 대한 움직임 벡터 추정시 그대로 적용할 수 있다. 즉, 아래 수식과 같다.

$$\|\mathbf{P} - \mathbf{R}\| + \lambda r(x, y, t-2) \leq J_2(m, n, \lambda) \quad (11)$$

따라서, 두 번째 참조영상에 대한 움직임 벡터 추정에 있어서 기존의 방법의 한 가지 검사조건(식(11))과 달리 두 가지 검사조건(식(10)과 식(11))이 존재하게 된다.

앞에서 기술한 바와 같이 두 가지 검사조건에 대한 검사를 수행한 후 두 가지 조건을 모두 만족하는 경우에 대해서만 움직임 벡터 추정을 수행함으로써 계산량을 감축할 수 있다. 제안된 방법은 첫 번째 참조영상에 대한 움직임 추정은 기존의 RSEA를 그대로 따르고, 두 번째 참조영상에 대해 적용된다. 첫 번째 참조영상에 대한 검사조건 식(6)이 만족된 경우, 해당 벡터는 최적 벡터 후보가 되므로 $\|\mathbf{M} - \mathbf{R}\|$ 가 계산되었다. 따라서 식(10)의 $\|\mathbf{P} - \mathbf{M}\|$ 가 간단한 계산에 의해 얻어진다면 두 번째 검사조건 식(10)을 검사하는데 계산량 측면에서 충분히 장점이 있다. $\|\mathbf{P} - \mathbf{M}\|$ 를 효과적으로 얻는 방법은 다음 절에서 다루어질 수 있다. 현재로서는

$|\mathbf{P}-\mathbf{M}|$ 가 이미 가지고 있는 데이터라고 가정하자. 두 번째 참조영상에 대한 움직임 벡터 탐색은 식(10) 뿐만 아니라 기존의 검사조건인 식(11)을 적용되며 이 두 조건을 모두 만족할 때, 실제 움직임 벡터 추정을 수행하며 그렇지 않을 경우에는 움직임 벡터 추정을 생략함으로써 계산량을 대폭 감소시킬 수 있다. 한편, 첫 번째 참조영상에 대한 검사조건 식(6)이 만족되지 않은 경우, 제안된 방법의 검사조건 식(10)이 적용되지 않는다. 이것은 적용할 수 없기 때문이 아니라 추가적인 계산을 필요로 하기 때문에 계산량 측면에서 장점이 없기 때문이다. 즉, 식(6)이 만족되지 않은 경우, 첫 번째 참조영상의 움직임 추정 과정에서 $|\mathbf{M}-\mathbf{R}|$ 를 계산하지 않았기 때문에 식(10)을 검사하기 위해서는 $|\mathbf{M}-\mathbf{R}|$ 를 계산하여야 하는 문제점을 발생시키기 때문이다.

그림1은 제안된 알고리즘의 의사코드(pseudo-code)이다. 의사코드의 13번째 줄의 검사조건이 새로 제안된 부분이며 이 조건을 만족하지 않는 경우가 많으면 많을수록 제안된 방법은 계산량에 있어서 장점을 가지게 된다. 첫 번째 참조영상에 대한 탐색과정(2번째 줄~10번째 줄)은 기존의 RSEA와 동일하지만, 두 번째 참조영상에 대한 탐색과정(11번째 줄~21번째 줄)은 추가적인

검사조건을 추가함으로써 검사조건을 보다 까다롭게 하여 조건을 만족하는 비율을 크게 낮추었다. 따라서, 제안된 방법은 두 번째 참조영상에 대한 탐색 계산량을 효과적으로 감축할 수 있다.

2. $|\mathbf{P}-\mathbf{M}|$ 의 연산

본 절에서는 제안된 검사조건에서 필요로 하는 $|\mathbf{P}-\mathbf{M}|$ 의 계산 방법 및 계산량에 대해 소개하고자 한다.

\mathbf{P} 와 \mathbf{M} 은 서로 다른 참조 영상의 블록 열벡터이지만 공간적 위치가 동일한 특징을 가진다. 그래서, $|\mathbf{P}-\mathbf{M}|$ 는 프레임차(frame difference)의 절대값의 블록 단위 합이다. $|\mathbf{P}-\mathbf{M}|$ 의 고속 연산을 살펴보기 위해, 우선 전체 영상의 크기를 $W \times H$ 이라고 하고, 블록 열벡터의 위치정보를 포함하여 나타내기 위해, (p, q) 에 위치한 \mathbf{P} 와 \mathbf{M} 를 $\mathbf{P}_{p,q}$ 와 $\mathbf{M}_{p,q}$ 라고 표기하자. 즉, $\mathbf{P}_{p,q}$ 와 $\mathbf{M}_{p,q}$ 는 블록 $\{f(p+i, q+j, t-1, 0 \leq i, j < N)\}$ 와 블록 $\{f(p+i, q+j, t-2, 0 \leq i, j < N)\}$ 에 해당한다. 이 때, $|\mathbf{P}-\mathbf{M}|$ 의 연산은 다음 두 단계 과정을 통해 계

```

1  M_R_available = 0; // flag for check if  $|\mathbf{M}-\mathbf{R}|$  is available
2  if( $|\mathbf{M}| - |\mathbf{R}| + \lambda r(x, y, t-1) > J(m, n, \lambda)$ ) continue;
3  else{
4       $J_{t-1} = |\mathbf{M}-\mathbf{R}| + \lambda r(x, y, t-1)$ ;
5      M_R_available = 1;
6      if( $J_{t-1} < J(m, n, \lambda)$ ){ // in case that  $(x, y)$  is better vector than  $(m, n)$ 
7           $J(m, n, \lambda) = J_{t-1}$ ; // update the minimal cost
8           $(m, n) = (x, y)$ ; // update the candidate vector
9          Ref_index = 0; // update the reference frame index
10 }
11 if( $|\mathbf{P}| - |\mathbf{R}| + \lambda r(x, y, t-2) > J_2(m, n, \lambda)$ ) continue;
12 else{
13     if(M_R_available &&  $|\mathbf{P}-\mathbf{M}| - |\mathbf{M}-\mathbf{R}| + \lambda r(x, y, t-2) > J_2(m, n, \lambda)$ )
14         continue;
15     else{
16          $J_{t-2} = |\mathbf{P}-\mathbf{R}| + \lambda r(x, y, t-2)$ ;
17         if( $J_{t-2} < J_2(m, n, \lambda)$ ){
18              $J_2(m, n, \lambda) = J_{t-2}$ ; // update the minimal cost
19              $(m, n) = (x, y)$ ; // update the candidate vector
20             Ref_index = 1; // update the reference frame index
21         }

```

그림 1. 제안된 방법의 의사 코드

Fig. 1. Pseudo-code of the proposed method

산량을 최소화할 수 있다.

- (1) 전체 영상내 전체 화소에 대해 차의 절대값 $d(x, y) = |f(x, y, t-1) - f(x, y, t-2)|$ 을 구한다. 이 연산은 $W \times H$ 의 차 연산과 절대값 연산을 요구하며, N^2 크기의 각 블록에 대해서는 N^2 의 차 연산과 절대값 연산을 요구한다.
- (2) 본 단계에서는 알고리즘의 쉬운 설명 뿐만 아니라 일반적으로 널리 채용되고 있는 무제한 움직임 추정 방법(unrestricted motion estimation method)을 채용한다고 가정한다. 여기서 무제한 움직임 추정 방법은 영상 경계 바깥도 영상 내와 마찬가지로 움직임 벡터 탐색영역에 포함시키는 방법이다. 이 방법의 채용은 영상 경계부분에 대한 움직임 추정 과정에 대한 특별한 처리를 필요로 하지 않으므로 설명이 간단해진다. 이 때, $|\mathbf{P}_{p+1,q} - \mathbf{M}_{p+1,q}|$ 은 바로 이전 위치의 블록간의 SAD ($|\mathbf{P}_{p,q} - \mathbf{M}_{p,q}|$)를 이용함으로써 약간의 연산만으로 얻을 수 있다. 즉, $|\mathbf{P}_{p+1,q} - \mathbf{M}_{p+1,q}|$ 는 $|\mathbf{P}_{p,q} - \mathbf{M}_{p,q}|$ 의 연산에서 공통으로 사용되는 화소를 제외한 $\mathbf{P}_{p+1,q}$ 와 $\mathbf{M}_{p+1,q}$ 에 새로 추가된 N 개의 화소에 대한 덧셈 연산과 제외된 N 개의 화소에 대한 뺄셈 연산을 수행함으로써 얻을 수 있다. 즉, \mathbf{P} 와 \mathbf{M} 에 대해 각각 N 번의 덧셈과 뺄셈 연산을 필요로 하므로 총 $2N$ 번의 덧셈과 $2N$ 번의 뺄셈을 필요로 한다.

절대값 연산은 덧셈 연산에 비해 계산량이 적으므로 이를 무시하면, 각 블록에서 필요로 하는 계산량 오버헤드(overhead) OH 는 다음과 같다.

$$OH = N^2 + 4N \quad (12)$$

여기서 N^2 과 $4N$ 은 첫 번째 단계와 두 번째 단계에서 각각 필요로 하는 덧셈 연산량이다. 움직임 추정시 각 탐색점에서의 요구 계산량이 N^2 번의 덧셈 연산임을 감안한다면, $|\mathbf{P} - \mathbf{M}|$ 에 필요한 연산은 2개의 탐색점에 대한 연산보다 작으므로 제안된 방법이 요구하는 계산량 오버헤드는 무시할 수 있을 정도이다.

IV. 실험 결과

제안된 방법의 성능 평가를 위해 RSEA를 참조영상

에 대해 순차적으로 적용하는 단순확장법과 비교하였다. 실험은 CIF, 30Hz의 7가지의 영상에 대해 수행되었다. 여기서 각 영상은 100장의 영상으로 구성되어 있다. 움직임 추정에 사용된 블록 크기는 16×16 이고 탐색영역은 ± 15 이며 사용된 참조영상의 개수는 두장이다. 그리고, H.264에서 사용된 R-D 최적화 과정에서 사용된 라그랑지안 계수를 사용되었다.

표1은 실험에 의해 얻어진 결과이다. 여기서 $ANSP_{t-1}$ 와 $ANSP_{t-2}$ 는 각각 프레임 $t-1$ 과 프레임 $t-2$ 의 블록당 SAD 연산이 수행된 평균 탐색점의 개수 (average number of searching points)이다. 그리고 $\lambda = 0$ 인 경우는 움직임 벡터의 부호화 비트량을 고려하지 않고 SAD만으로 움직임 벡터를 찾은 경우에 해당된다. 제안된 방법이 첫 번째 참조영상에 대한 탐색은 기존의 RSEA를 그대로 받아들임으로써 두 가지 방법이 같은 $ANSP_{t-1}$ 를 가지므로 하나의 열로 표현되었다. 반면, 두 번째 참조영상에 대한 계산량 $ANSP_{t-2}$ 은 기존의 단순확장법과 제안된 방법의 두열로 표현되어 있다. 표1은 제안된 방법이 $ANSP_{t-2}$ 에 있어서 단순확장법과 비교하여 39%~91%의 계산량을 감축시켜줌을 보여주고 있다. 제안된 방법은 Container와 Mobile 영상에 대해 가장 효과적인 것으로 나타났다. 예를 들어 $qp = 20$ 이고 Container 영상의 경우, 단순확장법에서는 총 고려해야할 탐색점 961개 중 평균 94.2개의 탐색점에 대해 SAD가 계산되었지만, 제안된 방법에서는 단지 평균 9.1개의 탐색점에 대해 SAD를 계산함으로써 10배 이상의 계산량 감축을 이루었음을 알 수 있다. 제안된 방법은 이미 언급한 바와 같이 $|\mathbf{P} - \mathbf{M}|$ 의 연산을 위한 오버헤드가 존재하지만 2개의 탐색점에 대한 연산보다 적기 때문에 무시할 수 있으므로 표1에서는 이 오버헤드를 포함시키지 않았다. 포함시킨다면 +2하면 된다.

한편, λ 의 증가에 따라 $ANSP$ 가 감소하는 경향이 있음을 관찰할 수 있다. 이것은 양자화 계수 qp 의 증가는 움직임 추정 오차 SAD와 라그랑지안 계수 λ 의 증가를 가져오지만 λ 의 증가가 SAD의 증가량에 비해 상대적으로 커져 탐색점의 원점 근처에서의 비용이 작게 나타나기 때문인 것으로 판단된다. 즉, RSEA는 탐색점의 원점을 중심으로 나선형으로 탐색하므로 원점 근처에서 최소 비용이 나타나게 되면 식(6)의 상한값(upper bound)가 작아서 탐색점들이 좀 더 많이 제외되어 계산량이 줄어드는 것으로 추정된다.

IV. 결 론

본 논문에서는 비트율을 고려한 SEA인 RSEA의 다중 참조영상에의 적용을 위한 효과적인 알고리즘을 제안하였다. 제안된 방법은 기존의 단순확장법과 비교하여 첫 번째 참조영상에 대한 움직임 추정 과정은 동일하지만 두 번째 참조영상부터는 첫 번째 참조영상에 대한 탐색과정에서 얻어진 결과를 이용하여, 새로운 검사조건을 추가하여 한번 더 검사를 수행함으로써 계산량을 크게 감소시키는 방법이다. 또한 제안된 방법은 전역탐색법의 고속 알고리즘이므로 전역탐색법과 동일한 움직임 추정 오차를 가진다는 장점을 가지고 있다. 실험 결과는 제안된 방법이 기존의 단순확장법에 비해 최대 91%까지 계산량을 감축시킴을 보였으며, 이러한 결과는 새로이 추가된 검사조건이 매우 효과적인 조건임을 증

명하고 있다. 본 논문에서는 비록 두장의 참조영상에 대해 적용한 것을 중심으로 설명하였지만, 제안된 방법은 같은 원리에 의해 쉽게 일반화될 수 있다. 그러나, 보다 많은 참조영상에 대한 제안된 방법의 적용은 추가적으로 필요로 하는 계산량 오버헤드를 고려하여 신중하게 확장하여야 할 것으로 보인다.

참 고 문 헌

- [1] F.Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proc. IEEE*, vol. 83, pp. 858-879, June 1995.
- [2] L.M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video*

표 1. 제안된 방법과 기존 방법의 성능 비교

Table 1. Performance comparison of the proposed method and the conventional method.

$\lambda(qp)^*$	Image	전역탐색법의 $ANSP$	$ANSP_{t-1}$	$ANSP_{t-2}$		PSNR (추정오차)
				단순확장법	제안된방법	
$\lambda = 0$	Coastguard	961	429.6	423.0	122.7	30.41
	Container	961	223.1	221.3	31.4	38.35
	Foreman	961	188.8	179.0	98.3	33.95
	Mobile	961	335.1	334.0	46.6	25.95
	Mot.&dau**	961	231.6	229.0	141.4	40.50
	Stefan	961	322.9	311.5	174.6	26.73
	Table tennis	961	469.9	458.5	280.3	30.68
$\lambda(qp = 10)$	Coastguard	961	416.4	408.3	113.1	30.41
	Container	961	190.6	185.1	24.7	38.35
	Foreman	961	173.4	161.5	85.0	33.92
	Mobile	961	325.0	322.6	43.0	25.92
	Mot.&dau.	961	206.0	199.7	123.6	40.47
	Stefan	961	307.7	294.5	160.9	26.73
	Table tennis	961	448.7	434.8	263.1	30.65
$\lambda(qp = 20)$	Coastguard	961	384.8	373.1	95.5	30.40
	Container	961	108.8	94.2	9.1	38.34
	Foreman	961	126.5	111.2	51.7	33.83
	Mobile	961	301.4	296.5	37.2	25.94
	Mot.&dau.	961	114.9	96.8	49.7	40.39
	Stefan	961	271.9	254.7	139.3	26.72
	Table tennis	961	393.8	372.9	220.1	30.61

* $\lambda = \sqrt{0.85 * 2^{qp/3}}$, ** Mot.&dau. = mother and daughter

- Technol.*, vol. 6, pp. 313-317, June 1996.
- [3] L.K.Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 419-423, Aug. 1996.
 - [4] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block matching motion estimation," *Proc. Int. Conf. Inform., Comm., Signal Processing*, Singapore, Sept. 9-12, 1997, pp. 292-296.
 - [5] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. on Image Processing*, vol. 4, no. 1, pp. 105-107, Jan. 1995
 - [6] M. Z. Coban and R. M. Mersereau, "A fast exhaustive search algorithm for rate-constrained motion estimation," *IEEE Trans. on Image Processing*, vol. 7, no. 5, May 1998.
 - [7] 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, March 2000.
 - [8] 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, pp. 945-952, 1998.
 - [9] ITU-T recommendation H.264 | ISO/IEC 14496-10 AVC, "Draft text of final draft standard for advanced video coding," Mar. 2003

저 자 소 개



강 현 수 (정회원)
 1991년 경북대학교 전자공학과
 학사
 1994년 한국과학기술원 전기 및
 전자공학과 석사
 2005년 3월~현재 충북대학교
 전기전자컴퓨터공학부
 조교수

<주관심분야 : 영상처리, 부호화, 콘텐츠보호>



박 성 모 (정회원)
 1987년 경북대학교 전기 및
 전자공학과 석사
 2003년 경북대학교 전기 및
 전자공학과 박사 과정
 1987년 1월~1992년 9월 하이닉스
 반도체(주) 주임연구원
 1992년 10월~현재 한국전자통신연구원
 책임연구원

<주관심분야 : 영상압축, SoC설계, 디지털설계>