

다중 후보영역 탐색기반 고속 블록정합 움직임 추정

조 영 창(曹永昌)*, 윤 정 오*(尹正五), 이 태 홍(李泰鴻)**

*경운대학교 컴퓨터전자정보공학부, **영남대학교 전기전자공학부

전화 : (0546) 479-1313 / 팩스 : (0546) 479-1069

Multiple Candidate Region Search Based Fast Block Matching Motion Estimation

Young Chang Cho*, Jeong Oh Yoon*, Tae Hong Lee**

*School of Engineering Kyungwoon University,

**School of Electrical and Electronic Engineering Yeungnam University

E-mail : yccho@kyungwoon.ac.kr, taehong@ynucc.yeungnam.ac.kr

Abstract

In this paper we propose the multiple local search method(MLSM) based on the motion information of the neighbor blocks. In the proposed method motions are estimated from the multiple searches of many candidate local search regions. To reduce the additional search points we avoid to search the same candidate regions previously visited using the distance from the initial search point to the recently found vector points. In the simulation the proposed method shows more excellent results than that of other gradient based method especially in the search of motion boundary.

계산량의 감소를 가져오지만, 평균 절대 오차(Mean Absolute Error : MAE) 공간에서 오차가 전역적 최소해(global minimum)에 근접할수록 단조 감소한다는 가정을 바탕으로 하고 있어서 탐색영역 내에 여러 최소점이 있는 경우[1] 국소 최소점에 고립될 가능성이 크다. 그러므로 전역적 최소해를 찾기 위해서는 탐색영역 내의 여러 국소 영역에 대해 다중적으로 탐색할 필요가 있다. 본 논문에서는 이러한 문제점을 개선하기 위하여 시공간적으로 인접한 블록의 움직임 정보에 기초한 다중 국소 탐색법(multiple search method : MLSM)을 제안한다. 제안한 방법에서 움직임 추정은 후보영역별로 이루어지며, 각 후보영역의 탐색은 이웃 블록의 움직임을 바탕으로 결정된다. 또한, 부가적인 탐색점 수를 줄이기 위해 탐색 중심점과 최소해와의 거리로부터 각 후보영역을 표시하고 같은 후보영역에 대한 중복탐색을 배제하였다.

I. 서론

블록정합법은 알고리즘의 단순성과 하드웨어적인 구현의 편리로 인해 MPEG 계열이나 H.261, H.263 등의 표준안 등에서 채택되어 많이 사용되고 있다. 대표적 방법으로 전역 탐색법은 탐색영역 내에서 최적의 움직임 벡터를 찾아내지만 탐색영역 전체를 찾아야 하는 많은 계산량으로 인해 다른 고속 알고리즘들이 제안되었다. 그러나, 3단계 탐색법(3SS)[2], 2D-LOG 탐색법[3], 4단계 탐색법(4SS)[4], 블록기반 경사 하강탐색(BBGDS)[5] 등은 비교적 정확한 움직임을 추정하면서

II. 다중 탐색 알고리즘

2.1 다중 해공간

블록 정합법은 이전 프레임의 탐색영역 내에서 현재 블록과 가장 유사한 블록의 위치를 움직임 벡터로 결정한다. 이 때, 평균 절대 오차는 유사도 평가를 위해 많이 사용된다. $f_t(m, n)$ 와 $f_{t-1}(m, n)$ 를 각각 t 와 $t-1$ 번째 프레임에서 크기가 $K \times K$, 좌·상단 좌표가 (m, n) 인 블록이라 하고, 수직 및 수평방향의 움직임 벡터를

각각 u, v 라 할 때, 후보벡터 (u, v) 에 대한 평균 절대 오차는 다음 식과 같이 정의된다.

$$MAE_{m,n}(u, v) = \frac{1}{K^2} \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} |f_t(m+i, n+j) - f_{t-1}(m+i+u, n+j+v)| \quad (1)$$

(그림 1)은 탐색공간내의 평균 절대 오차분포를 나타낸 것으로 여러 개의 국소 최소점들이 존재한다. 따라서, 평균 절대 오차값이 작아지는 방향으로 해를 찾아가는 고속 알고리즘들은 (그림 1)과 같은 오차분포에 있어서 지역적 최소해를 고립되기 쉽다.

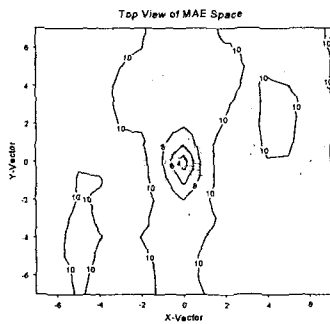


그림 1. MAE의 크기에 따른 윤곽선도
Fig. 1 Contour for the MAE distribution

본 논문에서는 여러 국소 해공간에 대해 다중적으로 해를 찾음으로써 여러 국소 해공간이 존재함으로써 야기되는 문제점을 개선한다. 제안한 방법에서는 전체 탐색영역에서 후보영역들을 표시하고 각 후보영역에 대해 최소해를 찾는다.

2.2 후보영역에 대한 초기 탐색점의 결정

일반적으로 블록은 이웃블록들과 비슷한 움직임을 가지며, 여러 움직임이 공존하는 움직임 경계영역에서도 고립된 경우를 제외하고는 적어도 하나 이상의 이웃블록과 비슷한 움직임을 갖는다[6]. (m, n) 번째 블록의 움직임 벡터를 $d_{(m,n)}=[u, v]$ 라 하고, (x, y) 번째 이웃블록의 움직임 벡터를 $d_{(x,y)}=[u', v']$ 라 할 때, 블록간의 유사도 분포는 다음 식과 같이 정의된다.

$$S_{(m,n)}(x, y) = \frac{D_{(m,n)}^{\max} - D(d_{(m,n)}, d_{(x,y)})}{D_{(m,n)}^{\max}} \quad (2)$$

식(2)에서 $D(\cdot)$ 는 두 움직임 벡터간의 거리를 나타내며, 다음과 같이 주어진다.

$$D(d_{(m,n)}, d_{(x,y)}) = [(u-u')^2 + (v-v')^2]^{1/2} \quad (3)$$

그리고, 식(2) 우변의 첫째 항은 다음 식과 같다.

$$D_{(m,n)}^{\max} = \max_D [D(d_{(m,n)}, d_{(x,y)}), \forall x, y] \quad (4)$$

(그림 2)는 'Garden'영상의 첫 프레임에 대한 움직임 벡터 유사도 분포를 나타낸 그림이다.

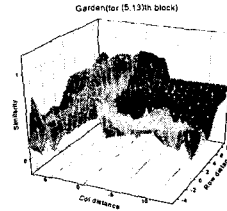


그림 2. 이웃블록 간의 움직임 벡터 유사도 분포
Fig. 2 Similarity of motion vectors between the neighbor blocks

그림에서와 같이 블록의 움직임은 이웃하는 블록의 움직임과 매우 높은 상관성을 갖는다. 본 논문에서는 후보영역별 탐색을 위해 이웃블록들의 움직임 벡터를 사용한다. 각 후보영역은 (그림 3)의 순서대로 탐색된다.

t 번째 프레임			t-1 번째 프레임		
3	4	5	7	8	9
2			6	1	10
			13	12	11

그림 3. 이웃블록의 움직임 벡터 참조 순서
Fig. 3 Block order for the reference of motion vectors (■ : current block)

2.3 탐색점 맵(Search Point Map:SPM)의 결정

탐색점 맵은 후보영역을 표시하며, 각 후보영역은 탐색 시작점과 최종 탐색점으로부터 정의된다. (그림 4)는 두 개의 탐색 시작점(■ 표시)과 추정된 최종 벡터를 이용하여 표시된 두 후보영역을 보여준다. 이웃하는 블록의 움직임 벡터를 시작점으로 얻은 최종 움직임 벡터는 해당 후보영역 내의 국소 최소점이다. 탐색 후, 후보영역은 국소 최소점을 기준으로 탐색 시작점과 국소 최소점까지의 거리를 반지름으로 하는 원으로 표시되고 후보영역 내의 후보벡터들은 '1'로 표시된다. 이로써 탐색영역 내의 움직임 추정이 후보영역 단위로 행해지게 되어 화소방문횟수를 증가시키지 않으면서 추정벡터의 정확성을 증대시킬 수 있다. 탐색영역내의 임의 후보벡터 $d=[u, v]$ 에 대한 탐색점 s_d 은 다음 식에 의해 후보영역으로 표시된다.

$$s_d = \begin{cases} 1, & \text{if } \|d - d_c\| \leq \|d_i - d_c\| \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

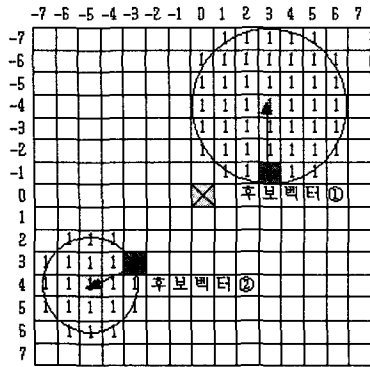


그림 4. 탐색점 맵
Fig. 4 Search map (x:(0,0), 1:visited sites)

식(5)에서 $d_c=[u_c, v_c]$ 는 후보영역에서 얻은 최종 움직임 벡터이며, $d_i=[u_i, v_i]$ 는 이웃블록의 움직임 벡터로써 각 후보영역의 탐색 시작점에 해당된다. 또한, $\| \cdot \|$ 는 2차원 벡터간의 거리를 나타낸다.

2.4 제안한 다중 탐색법(Proposed MLSM)

제안한 다중 탐색법은 이웃블록의 움직임 벡터를 이용하여 여러 탐색 후보영역에 걸쳐 최소해를 찾는다. 또한, 각 후보영역은 탐색점 맵에 표시되어 동일 후보영역을 중복해서 탐색하지 않음으로써 다중 탐색에 의한 계산량을 줄인다. 제안한 방법에 의한 움직임 추정절차는 다음과 같다.

- (i) 탐색점 맵의 전체 탐색점에 대해, $s_d=0$, $d=[u, v]$ ($-7 \leq u, v \leq 7$)로 설정하고, d_g 와 MAE_{min} 을 할당한다.
- (ii) 초기 움직임 벡터 d_i 에 대해, $d_i=d_k$ 로 설정한다. 여기서 d_k 는 k번째 이웃 블록의 움직임 벡터이다.
- (iii) $s_{d_k} = 0$ 이면, 단계(iv)로, $s_{d_k} = 1$ 이면, 단계(vi)로 이동한다.
- (iv) d_i 를 중심으로 블록기반 경사 하강 탐색을 시행한다. 탐색결과, 국소 최소해와 평균 절대 오차를 d_c 와 MAE_k 에 할당하고, $MAE_{min} > MAE_k$ 이면, $MAE_{min} = MAE_k$, $d_g=d_c$ 로 설정한다.
- (v) 단계(iii)에서의 초기 탐색점 d_i 와 단계(iv)에서 찾은 최소해 d_k 를 식(5)에 적용하여 현재 후보영역을 탐색점 맵에 표시한다.
- (vi) 이웃블록이 마지막이 아니면($k \neq 13$), $k=k+1$ 로 설정한 후, 단계(iii)으로 이동하여 다음 후보영역을 위해 (iii)~(v)단계를 반복하고, 마지막 이웃블록이면($k=13$), 단계(vii)로 이동한다.
- (vii) 다음 탐색블록에 대한 움직임 추정을 위해 단계(i)에서 단계(vi)까지 반복한다.

III. 실험 결과

제안한 움직임 추정법을 검증하기 위해 Garden(352×240 : 100 프레임), Claire(352×240 : 100 프레임), Salesman(352×240 : 100 프레임) 그리고 Tabel Tennis(360×240 : 100 프레임)에 대해 모의실험을 실시하였다. 프레임별로 움직임 보상영상에 대한 PSNR과 방문 화소수(Number of Search Point: NSP)에 대해서 제안한 방법과 기존의 방법을 비교하여 평가하였으며, 각 방법에 대한 움직임 벡터의 니들그램을 통해 전역 탐색법에 의한 결과와 비교하였다. 식(6)과 식(7)은 PSNR 계산을 위해 사용하였다.

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (6)$$

$$MSE = \frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [f_t(i, j) - \hat{f}_t(i, j)]^2 \quad (7)$$

식(7)은 t번째 프레임 $f_t(i, j)$ 과 움직임 보상된 t번째 프레임 $\hat{f}_t(i, j)$ 영상간의 보상오차를 나타낸다. (그림 5)는 'Garden' 시퀀스에 대해 각 방법에 의한 PSNR을 프레임에 따라 나타낸 것으로 제안한 방법은 전역 탐색법과 비슷한 결과를 나타냈으며, 3SS, 4SS, BBGDS에 의한 결과보다 우수한 것으로 확인되었다. (그림 6)은 평균 방문화소수를 비교한 그림이다. 제안한 방법은 10번째 프레임 근처에서 다소 많은 화소를 방문하지만, 전반적으로 BBGDS나 4SS보다 방문화소가 작게 나타났다. 이는 이웃블록에 의한 탐색과 후보영역에 대한 중복탐색을 배제함으로써 얻어지는 결과이다. 이밖의 'Table tennis', 'Claire' 및 'Salesman' 시퀀스에 대한 실험결과를 PSNR과 NSP에 대해 표 1과 표 2에 표기하였다. (그림 7)은 각 방법에 의한 움직임 추정결과를 니들그램으로 나타낸 것이다. 특히, 움직임 경계에서 제안한 방법에 의한 추정결과가 더 정확한 것을 관찰할 수 있다.

VI. 결론

본 논문은 블록정합을 기반으로 탐색영역 내의 여러 후보영역들에 대해 다중적으로 움직임을 추정하여 탐색영역 내의 전역적 최소해를 찾는 방법을 제안하였으며, 부가적인 계산량을 줄이기 위해 각 후보영역을 탐색점 맵에 표시하여 동일한 후보영역에 대한 중복탐색을 배제하였다. 제안한 방법에 대한 검증결과 특히, 움직임 경계영역에서 정확한 움직임을 추정할 수 있음을 확인할 수 있었으며, 계산량의 큰 증가없이 움직임 추정의 정확성을 개선시킬 수 있었다.