

탐색 범위를 적용한 비교 루틴 고속 블록 움직임 추정방법 알고리즘

임유찬, 민경욱, 정정화
 한양대학교 전자공학과
 전화 : 02-2290-0558 / 핸드폰 : 016-439-8954

Comparison Fast-Block Matching Motion Estimation Algorithm for Adaptive Search Range

Yu-Chan Lim, Kyeong Yuk Min, Jong Wha Chong
 Dept. of Electronic Engineering, Hanyang University
 E-mail : eeyclim@hanmail.net

Abstract

This paper presents a fast block-matching algorithm to improve the conventional Three-Step Search (TSS) based method. The proposed Comparison Fast Block Matching Algorithm (CFBMA) begins with D_{AB} for adaptive search range to choose searching method, and searches a part of search window that has high possibility of motion vector like other partial search algorithms. The CFBMA also considers the opposite direction to reduce local minimum, which is ignored in almost conventional based partial search algorithms. CFBMA uses the summation half-stop technique to reduce the computational load. Experimental results show that the proposed algorithm achieves the high computational complexity compression effect and very close or better image quality compared with TSS, SES, NTSS based partial search algorithms.

I. 서론

블록 정합 알고리즘은 움직임 추정의 한 방법으로 동영상에서의 연속된 프레임 사이의 유사성을 고려하여 이전 프레임(previous frame)과 현재프레임(current

frame)사이의 움직임 벡터를 산출하여, 움직임 보상에 이 벡터를 이용하여 이미지를 재구성한다. 움직임 벡터는 식(1)과 같이 SAE(Sum Absolute Error)값을 이용하여 최소오차를 갖는 블록의 위치로 정해지게 된다[4].

$$SAE = \sum_{m=x}^{x+N-1} \sum_{n=y}^{y+N-1} |I_k(m, n) - I_{k-1}(m+dx, n+dy)|$$

$$- w \leq dx, dy \leq w \text{ 식(1)}$$

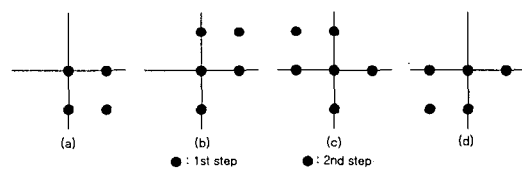


그림 3. SES 알고리즘의 비교점에 대한 추정 영역 (a) 제1사분면 (b) 제2사분면 (c) 제3사분면 (d) 제4사분면

$I_k(m, n)$, $I_k(m+dx, n+dy)$ 는 현재 프레임과 이전 프레임 블록내의 $N \times N$ pixels 후보블록에 상당하는 (m, n) , $(m+dx, n+dy)$ 이며, w 는 해당 탐색 영역의 크기를 나타낸다. 전역탐색 기법(Full Search Block Matching Algorithm ; FSBMA)은 탐색영역내

모든 점을 탐색하여 블록정합을 행하는 방식이며 추정
의 정확도를 고려하였을 때 매우 좋은 화질의 이미지
를 재구성할 수 있다. 그러나 이 알고리즘은 특성상
매우 많은 연산량을 필요로 하며 HDTV나 고화질을
요구하는 TV에서는 탐색영역이 커지므로 많은 연산량
을 요구하게 된다[6],[8]. 그러므로 연산량을 줄이는 동
시에 좀 더 간단하고 효율적인 알고리즘이 제안되었다
[1]-[8]. 그 중에서 TSS(Three Step Search)는 이전단
계에서 찾은 최소추정오차가 점진적으로 단조 감소하
다는 가정하에 영역을 탐색하는 방식이다[2]. SES
(Simple and Efficient Search) 탐색방법은 그림1에서
와 같이 4방향 중 최소 SAE를 찾는 방향으로 탐색을
한다. 이 알고리즘은 정합오차가 탐색영역안에서 움직
임 벡터가 확률이 높은 곳으로 접근할수록 SAE가 단
조 감소한다는 특성을 고려하여 정합시 비교과정을 통
해 탐색방향을 결정한 후 움직임을 추정하는 방식이다
[1],[7]. 본 논문에서 제안한 CFBMA (Comparison Fast
Block Matching Algorithm)은 적응 탐색 범위를 적용하
여 최소 SAE를 찾아 움직임을 예측하는 동시에 누산
중단 기술을 채택하여 연산량을 줄여 화질은 기존의 제
안된 여러 알고리즘과 동일하거나 보다 좋은 특성을 유
지하였다. 또한 국소 최소점(near-local optimum)의 영
향을 줄이기 위해 탐색 반대 방향에서도 정합을 실시하
여 움직임이 크고, 작은 영상에 대해서도 움직임을 추정
할 수 있다.

II. 본론

2.1 제안알고리즘의 구조

제안된 움직임 추정 알고리즘은 탐색영역을 적응적
으로 설정하고, 최소 SAE를 찾아 비교하여, 움직임 벡
터가 있을 확률이 큰 영역을 탐색한다. 탐색영역 설정
은 움직임 정도를 나타내는 식(1)에 의해 $D_{AB} = 0$ 일
경우와 $D_{AB} \neq 0$ 일 경우로 나누어서 결정하였다. 기존
의 DSWA (Dynamic Search-Window Algorithm)[3]에
서는 SAE가 작은 순서대로 2개의 탐색점을 적응적인
탐색영역에 사용하였으나 제안방법은 그림 1에서와 같
이 탐색 MB(Macro Block)의 중앙에 위치한 첫 번째
점(A)과 주변 최소 SAE를 갖는 두 번째 점(B)을 사용
하여 식(2)과 같이 적응적인 탐색영역을 결정한다. 이는
움직임 벡터가 MB의 중앙에 위치할 확률이 큰 정적인
영역에서 중앙에 있는 점(A)을 중심으로 비교하여 탐색
함으로써 더욱 정확한 움직임 벡터를 찾기 위함이다. 표
1의 실험결과에서 움직임이 적은 영상에서 FSBM과

NTSS(New Three Step Search)를 제외한 다른 알고
리즘에 비해 좋은 결과를 얻을 수 있었다.

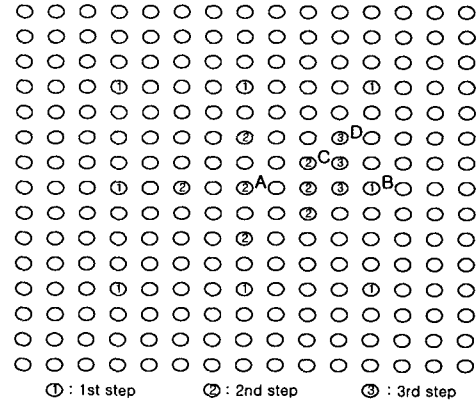


그림 4. 제안 알고리즘 예측방법($D_{AB} \neq 0$)

(SAE(A)가 SAE(B)보다 작거나
같을 경우일 때 : 점(C)을 결정
→SAE(A)가 SAE(C)보다 클 경
우 : 점(D)을 움직임 벡터로 결정,
 $W=7, MV: A \rightarrow B \rightarrow C \rightarrow D = [3, -2]$)

$$D_{AB} = \left| 1 - \frac{B(\text{Min}_{SAE})}{A(\text{SAE}_{center})} \right| \quad \text{식(2)}$$

위의 식(2)에서 $A(\text{SAE}_{center})$ 는 그림 2에서와 같이
가운데 점(A)에서의 SAE값을 말하며, $B(\text{Min}_{SAE})$ 는
 $A(\text{SAE}_{center})$ 를 제외한 주변 8개의 점 중 최소 SAE
값을 말한다. 그러므로 D_{AB} 는 적응 탐색영역을 설정
하기 위한 기준값으로 쓰이며, 이를 이용한 구현 예를
그림 2와 그림 3에서 표현하였다.

2.2 CFBMA(Case_1 : $D_{AB} \neq 0$)

1단계 : 그림2에서와 같이 점(A)을 제외한 8개의 점
에 대한 정합을 실시하여 최소 SAE값을 갖는 점(B)을
구한다.

2단계 : 점(A)의 정합을 실시한 후, 점(A)과 점(B)의
SAE값에 대해서 크고 작음을 비교하여 그림 2의 2nd
step 같이 탐색 방향을 결정한다. 결정 후 점(A) 또는
점(B)주위의 6개의 탐색점에 대한 정합이 이루어진다.
그림2는 점(A)이 점(B)보다 SAE값이 같거나 작을 경

위의 예를 보여주며 점(A)을 중심으로 6개에 대한 점 탐색 정합 후 최소 SAE값을 갖는 점(C)이 결정된다.

3단계 : 그림 2에서와 같이 점(A)이 점(C)보다 클 경우, 점(C) 주위 3개에 대한 점을 정합 후 최소 SAE값을 갖는 점(D)을 움직임 벡터로 지정하게 된다.

그림2에서 제안알고리즘은 탐색시 결정된 방향뿐 아니라 반대방향도 탐색정합에 이용된다는 것을 보여주는데 점(A)을 중심으로 2 pixel 떨어진 위, 아래, 왼쪽 부분에 해당되며, 이는 국소 최소점에 고립될 확률을 줄여주며 특히 다양한 움직임을 갖는 영상에서 움직임 보상을 할 때 재구성한 영상의 화질을 향상시킬 수 있다. 또한 그림 2는 식(2)에 의해 비교적 큰 움직임을 포함한 전반적인 움직임에 대하여($D_{AB} \neq 0$), 탐색영역에 차등을 두어 정합함으로써 움직임이 크거나 작거나 관계없이 모든 움직임에 대해서 추정할 수 있다. 또한 누산 중단 기술을 채택하여 연산량을 줄이면서 화질은 기존의 제안된 여러 알고리즘과 동일하거나 보다 좋은 특성을 유지하였다. 누산중단기술은 다음과 같다 [7]. 움직임 벡터는 최소 오차를 갖는 블록에서 산출되므로 이전에 계산된 최소의 오차합 보다 큰 오차 합이 발생한 블록에서는 움직임 벡터가 발생할 수 없다. 그러므로 한 블록의 정합 중 이전의 오차합이 크게 되는 순간 누산을 중단하고 다음 블록에 대한 정합을 실시한다. 이는 전체 정합 연산량을 줄일 수 있게 해준다.

$$Total_{error} = Error_{sum} - X_{shs}$$

$$X_{shs} = Error_{BLK} - Error_{ACC}$$

- $Error_{sum}$: 각 단계별 Error의 대한 합
- $Error_{BLK}$: 한 MB범위 내에서의 정합 Error값
- $Error_{ACC}$: 현재의 구하는 SAE가 지금까지 구한 최소 SAE값보다 커질 때의 Error값

2.3 CFBMA(Case_2 : $D_{AB} = 0$)

그림 3은 식(2)에서 $D_{AB} = 0$ 인 경우 SAE_{center} 값과 주변 8개의 점 중 최소 SAE값을 갖는 Min_{SAE} 값이 거의 차이가 없다는 사실(움직임이 작다.)을 고려하여 그림3과 같이 탐색영역을 작게 설정하여 탐색함으로써 더욱 정확한 움직임 벡터를 찾을 수 있다.

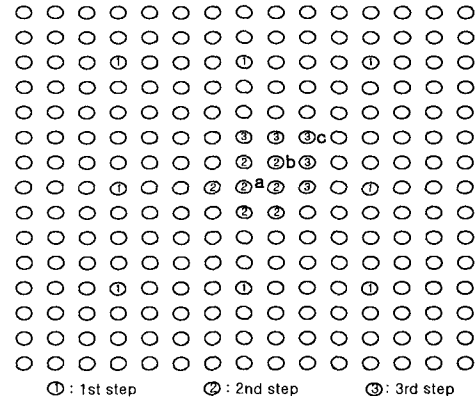


그림 5. 제안 알고리즘 예측방법($D_{AB} = 0$)

(점(a)을 중심으로 주변 6개의 점에서 최소 SAE(b)를 찾는다.→SAE (a)가 SAE(b)보다 클 경우→5개의 점 중 최소 SAE값을 갖는 점(c)을 움직임 벡터로 결정, $W=7, MV:$
 $a \rightarrow b \rightarrow c = [2, -2]$)

III. 실험결과 및 검증

본 논문은 CIF(30frames/s with 352×240pixels/frames)과 CCIR(30frames/s with 352×288pixels/frames)을 사용하였고, 16×16의 MB와 ±7 pixels의 탐색영역을 사용하였다. 사용된 이미지는 정적이고 동적인 이미지와 불규칙적인 이미지를 선택하였다. 검증항목은 화질을 나타내는 PSNR과 한 Frame내에서의 전체 정합수를 나타내는 연산량을 선택하였다. Table1과 그림4에서 보는 것과 같이 missa 이미지에서는 FSBM과 NTSS를 제외하고, 제안알고리즘은 TSS와 SES보다 높은 PSNR값을 가지고 있으며, football 이미지와 Table Tennis 이미지에서는 FSBM를 제외하고, TSS와 유사하거나 좋은 특성을 가지고 있다. 위의 결과에서 평균적으로는 제안된 CFBMA는 탐색 범위를 적용하여 동적인 움직임과 불규칙적인 움직임 영상뿐만 아니라 작은 영상의 이미지에서도 기존의 다른 알고리즘과 유사하거나 보다 좋은 성능을 보임을 알 수 있다. Table2와 그림5는 평균 연산량에 대한 비교 결과를 보여준다. 평균 연산량은 FSBM에 비해서 TSS 89.56%, SES 93.30%, NTSS 92.89% 제안알고리즘은 94.33%의 적은 연산량을 나타낸다. 이는 제안 알고리즘이 다른 알고리즘에 비해 적은 연산량으로 고속 움직임 추정이 가능함을 보여준다.

IV. 결론

본 논문은 적응탐색영역을 크게 $D_{AB} \neq 0$, $D_{AB} = 0$ 나누어 크고, 작은 움직임뿐만 아니라 불규칙한 움직임을 갖는 다양한 영상에 대해서도 좋은 특성을 나타내었다. 움직임 추정과정 중 국소 최솟점(near-local minimum)의 영향을 고려한 정합을 수행하였으며, 또한 비교 루틴을 적용하여 최소 SAE값이 확률이 높은 곳을 탐색함과 동시에 누산 중단 기술을 적용하여 연산량을 크게 줄일 수 있었다.

참고문헌

[1] Jianhua Lu and Ming L. Liou, "A sample and efficient search algorithm for block-matching motion estimation," IEEE Trans. Circuits Syst., video Technol., vol. 7, no 2, April 1997.

[2] Renxiang Li, Bing Zeng, and Ming L. Liou, "A new three-step search algorithm for block motion estimation" IEEE Trans. Circuits Syst., video Technol., vol. 4, no 4, August 1994.

[3] Liang-Wei Lee, Jhing-Fa Wang, Jau-Yien Lee, and Jung-Dar, "Dynamic search-window adjustment and interlaced search for block-matching algorithm" IEEE Trans. Circuits Syst., video Technol., vol. 3, no 1, February 1993.

[4] M. Ghanbari, "The cross-search algorithm for motion estimation" IEEE Trans. Commun., vol. 38, no. 7, July 1990.

[5] Lai-Man Po and Wing-Chung Ma, "A novel four-step search algorithm for fast block motion estimation" IEEE Trans. Circuit Syst., video Technol., vol. 6, no 6, June 1996.

[6] K. M. Yang, M. T. Sun, and L. Wu, "A family of VLSI designs for the motion compensation block-matching algorithm," IEEE Trans. Circuit Syst., vol. 36, no. 10, Oct. 1989.

[7] Humaira Nisar and Tae-Sun Choi, "An advanced center biased three step search algorithm for motion estimation" IEEE Trans., Vol. 1, pp. 95-98, 2000.

[8] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," IEEE Trans. Circuit Syst. Video

표 1. AVERAGE PSNR OF THE FIRST 90 FRAMES

Algorithm	PSNR(dB)			
	tennis	football	missa	average
FSBM	28.383	22.080	37.672	29.378
TSS	27.457	21.482	35.799	28.246
SES	26.834	20.965	35.273	27.690
NTSS	26.850	20.941	36.712	28.167
CFBMA	27.203	21.375	36.102	28.225

표 2. AVERAGE COMPUTABILITY OF THE FIRST 90 FRAMES

Algorithm	Computability(*1000)				
	tennis	football	missa	average	%diff
FSBM	19960	19960	24144	21354	0%
TSS	2089	2089	2508	2228	-89.56%
SES	1340	1347	1602	1429	-93.30%
NTSS	1422	1423	1708	1517	-92.89%
CFBMA	1139	1089	1401	1209	-94.33%

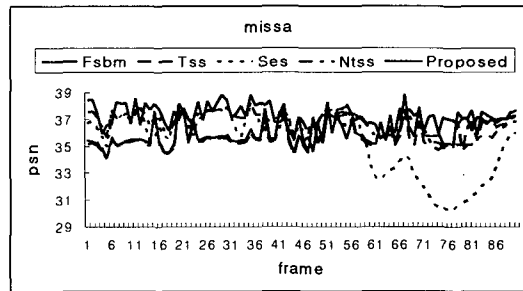


그림 6. Missa 영상에 대한 PSNR 그래프

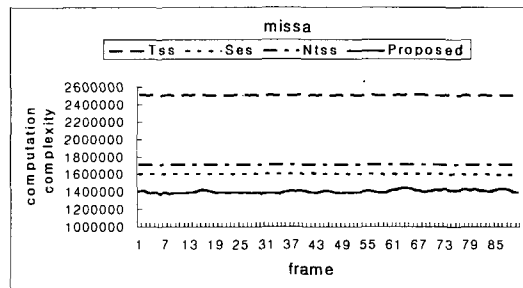


그림 7. Missa 영상에 대한 연산량 그래프