

# 고속 블록 정합 움직임 추정을 위한 적응적 패턴 탐색 Adaptive Pattern Search for Fast Block-Matching Motion Estimation

곽성근(Sung-Keun Kwak)<sup>1)</sup>

## 요 약

비디오 시퀀스의 현재 블록의 움직임 벡터와 이전 블록의 움직임 벡터는 시간적 상관성을 갖고 있다. 본 논문에서는 현재 프레임 블록의 인접 블록으로부터 예측된 움직임 정보를 구하여, 이를 탐색 원점으로 하여 수정된 다이아몬드 지역 레이더 패턴으로 블록 정합을 수행하는 블록 정합 움직임 추정 방식을 제안한다. 실험 결과 제안된 방식은 전역탐색을 제외한 기존의 방식들에 비해 PSNR 값에 있어서 평균적으로 0.02~0.37[dB] 개선되고 움직임 벡터 예측의 속도에 있어서 약 14~24% 이상의 높은 성능 향상을 보였다.

## Abstract

There is the temporal correlation of the video sequence between the motion vector of current block and the motion vector of previous block. In this paper, we propose the improved diamond search pattern using an motion vector prediction candidate search point by the predicted motion information from the same block of the previous frame.

Simulation results show that PSNR(Peak-to-Signal Noise Ratio) values are improves as high as 14~24% in terms of average number of search point per motion vector estimation and improved about 0.02~0.37dB on an average except the full search(FS) algorithm.

논문접수 : 2004. 11. 10.

심사완료 : 2004. 12. 5.

1) 정회원 : 시립인천전문대학 컴퓨터정보과

### 1. 서 론

현재 많은 비디오 코딩에서는 데이터의 흐름의 규칙성, 계산의 복잡도, 하드웨어의 구현을 고려하여 블록 정합 알고리즘(BMA: Block-Matching Algorithm)을 많이 사용하고 있다. 가장 간단한 블록 정합 알고리즘으로는 전역 탐색(FS: Full Search) 기법이 있는데 이 기법은 움직임 추정을 할 때 탐색 범위 내의 가능한 모든 블록을 조사하여 움직임 벡터를 찾는다. FS는 탐색 범위 내에서 가장 적합한 움직임 벡터(최적의 성능)를 구할 수 있지만 계산량이 많으므로 실시간 비디오 코딩 응용 분야 및 소프트웨어 구현에 많은 어려움을 가지고 있다.

이러한 FS의 단점을 극복하기 위해 속도가 개선된 TSS (Three Step Search)[1], NTSS(New Three Step Search)[2], DS(Diamond Search)[3], HEXBS(Hexagon-based Search)[4] 등의 다양한 고속 블록 정합 알고리즘(FBMA: Fast Block Matching Algorithm)이 개발 되었다. 이들 속도 개선 알고리즘은 주로 탐색 영역 내에서 탐색할 위치의 포인트 개수를 감소시켜 계산량의 감소를 유도하는 탐색 패턴을 사용한다. 탐색 패턴이란 블록 정합을 위해 각 탐색 단계에서 정합 기준값을 검사하는 탐색점들을 의미하며, 이 탐색점들 중에서 최소 정합 오차(BDM: Block Distortion Measure)를 가지는 위치를 중심으로 다음 단계의 움직임 탐색이 수행된다. 따라서 고속 블록 정합 움직임 탐색 방법에서 사용되는 탐색 패턴은 그 모양과 크기에 따라 탐색의 속도와 성능을 좌우하는 중요한 요소가 될 수 있다.

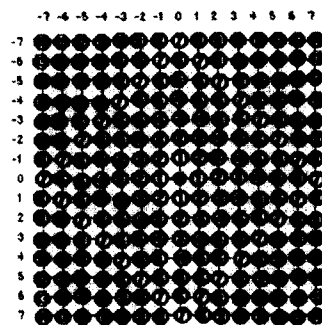
최근 많이 사용되는 탐색 패턴의 형태는 다이아몬드 패턴과 지역 탐색 패턴이 있다. 다이아몬드 탐색은 매 단계마다 적은 수의 탐색점을 사용하려는 반면에 지역 탐색은 다이아몬드 탐색점보다 많은 탐색점을 할당하여 정밀도를 높인다. 이러한 두 방법은 정밀도와 계산량의 관점에서 각각 상보점(trade-off)을 가진다. 즉,

화질을 유지하면서 속도를 크게 개선하기 힘들다.

따라서 본 논문에서는 최적의 탐색점을 찾기 위해 요구되는 계산량을 효과적으로 줄이기 위해 수정된 다이아몬드 지역 탐색 알고리즘을 제안한다. 제안한 알고리즘은 현재 매크로 블록의 움직임 벡터는 공간적으로 인접한 매크로 블록들이 비슷한 속도로 거의 같은 방향으로 움직인다는 점을 고려하여 예측 움직임 벡터로 탐색 원점을 이동시켜 정밀도를 높이고, 함축성 있게 계산량을 줄이기 위해 움직임 방향으로만 탐색점을 추가하여 초기 종료를 하는 수정된 다이아몬드 지역 패턴을 이용하여 탐색하게 된다.

### 2. 제안된 알고리즘

제안된 알고리즘은 기존의 ADZS(Advanced Diamond Zonal Search)[5]가 [그림 1]과 같이 탐색 영역의 중심으로부터 다이아몬드 지역 패턴을 사용하는 것과는 달리, 2개의 공간적으로 인접한 블록을 참조한 움직임 벡터 예측 후보를 중심으로 수정된 다이아몬드 지역 패턴으로 탐색을 수행한다.

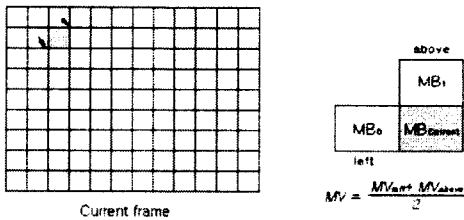


[그림 1] 다이아몬드 지역 탐색법

먼저, 제안된 알고리즘의 현재 프레임 블록의 움직임 벡터를 예측하기 위해 카메라의 움직임이 거의 대부분 수평, 수직 방향임을 고려

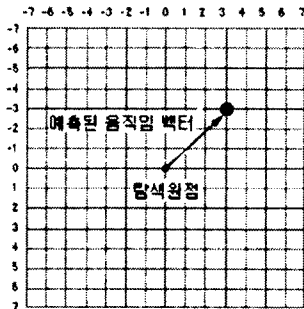
하여 [그림 2]와 같이 현재 프레임 블록의 수평(left), 수직(above)에 있는 매크로 블록을 참조한다. 이때 현재 블록의 예측 움직임 벡터는 식 (1)과 같이 이들 인접 매크로 블록 각각의 움직임 벡터의 평균값으로 한다.

$$MV_p = \frac{MV_{above} + MV_{left}}{2} \quad (1)$$

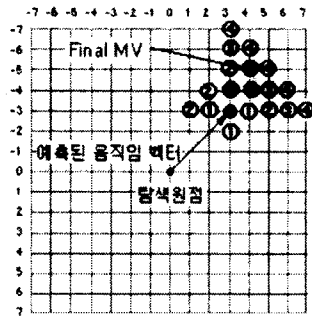


[그림 2] 참조하는 인접 매크로 블록

그리고, 식 (1)에서 구한 예측된 움직임 벡터로 [그림 3]의 (a)와 같이 탐색 원점을 이동시켜 이 점을 중심점으로 [그림 4]의 수정된 다이아몬드 탐색 패턴을 적용한다. 패턴 적용 단계를 라고 한다면, [그림 1]의 ADZS는 매 탐색 단계마다 검사되는 탐색 후보점은 개로 할당된다. 그러나 제안된 알고리즘에서는 탐색할 위치의 포인터 수를 줄여 계산량의 감소를 유도하기 위해 [그림 3]의 (b)와 같이 현재 최소 정합 오차 점의 방향으로만 패턴을 적용하여 탐색을 수행한다.



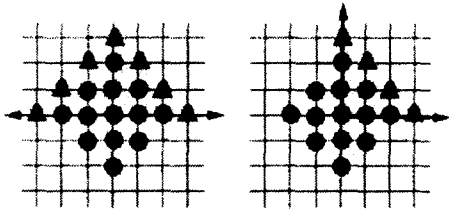
(a) 탐색 원점의 이동



(b) 탐색 경로의 예

[그림 3] 제안한 움직임 벡터 추정 과정

탐색 과정은 [그림 3]의 (b)의 예와 같이에서 지역 ①의 4개 탐색 후보점을 추가하여 최소 정합 오차를 계산하여 지역 ①의 최소 정합 오차가 탐색 원점 ●의 정합 오차보다 작다면 단계로 간다.에서 지역 ②의 탐색 후보점들을 추가하여 구한 최소 정합 오차가 지역 ①보다 작으면에서 지역 ③의 최소 정합 오차를 계산한다.에서 지역 ④의 최소 정합 오차가 ③의 최소 정합 오차보다 크면 탐색을 종료하고 ③의 최소 정합 오차를 움직임 벡터 최종해로 한다. 즉, 단계의 최소 정합 오차가 단계에서 검사된 탐색 후보점의 최소 정합 오차보다 작으면 탐색을 종료하고, 최소 정합 오차 점을 움직임 벡터로 한다. 만약 단계의 검사된 점들 중에 최소 정합 오차가 있다면, 최소 정합 오차 점이 있는 방향으로 단계를 위한 탐색 후보점을 추가한다. 매 단계마다 탐색 후보점들의 추가는 단계에서 계산된 최소 정합 오차가 수평/수직 방향에 위치할 경우는 [그림 4]의 (a)와 같이 180도 범위 내에 추가하며, 대각선 방향에 위치할 경우는 [그림 4]의 (b)와 같이 90도 범위 내에 추가한다.



(a) 수직/수평에 위치 (b) 대각 방향에 위치  
 [그림 4] 제안된 알고리즘의 탐색 후보점 추가

제안된 알고리즘을 요약하면 다음과 같다.

**1단계:** 현재 프레임 블록의 인접 블록들의 움직임 벡터 정보를 이용하여 예측된 움직임 벡터 값을 구하여 이 점으로 탐색 원점을 이동시킨다.

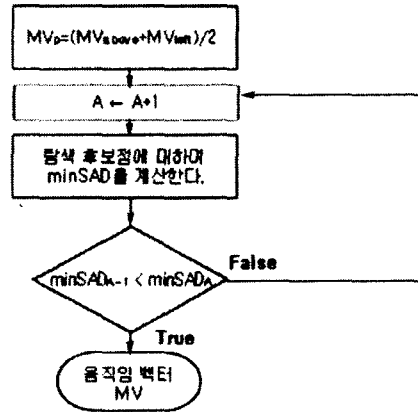
**2단계:** 이동된 탐색 원점을 중심으로 수평/수직 방향으로 1화소 간격에 위치하는 탐색 후보점에 대하여 최소 정합 오차를 계산한다. 중심점이 최소 정합 오차면 4단계로 가고, 아니면 3단계로 간다.

**3단계:** 패턴 적용 단계를 1 증가시키고 이전 단계의 최소 정합 오차 점이 있는 방향으로 탐색 후보점을 추가한다. 이 단계에서 추가된 점보다 이전 단계의 점이 최소 정합 오차면 4단계로 가고, 그렇지 않으면 3단계를 반복한다. 단, 탐색 영역( $w=±7$ )을 벗어나는 모든 점들은 무시한다.

**4단계:** 이전 단계에서 구한 최소 정합 오차 점이 움직임 벡터의 최종 해가 된다.

따라서 제안된 기법은 예측된 움직임 벡터를 초기점으로 하여 매 탐색 단계마다 지역 전체를 할당하지 않고 해당 방향의 탐색 후보점만 탐색함으로써, ADZS보다 탐색 대상 영역이 줄어들게 되어 움직임 추정에 대한 계산량이 상대적으로 많이 줄어들게 된다. 그리고 초기점을 중심으로 상대적으로 축소된 탐색 영역 내에서 미소 변위 추정을 행하게 되므로 ADZS

보다 효율적으로 움직임을 추정할 수 있으며, FS와 ADZS와 유사한 화질을 갖는다는 장점이 있다. [그림 5]는 제안한 알고리즘의 순서를 간단히 나타낸 흐름도이다.



[그림 5] 제안한 움직임 벡터 추정 방식의 흐름도

### 3. 실험 결과

제안된 기법의 성능을 평가하기 위하여 CIF(Common Intermediate Format: 352×288 pixels) 형식의 6개 영상에 대해 각각 80프레임씩을 대상으로 실험하였고, 비교 탐색 기법으로는 FS, TSS[1], DS[3]와 예측 탐색 알고리즘인 NNS[6], ADZS[5], 그리고 제안한 탐색 기법을 사용하였다. 그리고 움직임 추정에 사용된 매크로블록의 크기는 16×16 화소며, 탐색 영역의 변위는 ±7을 적용하여 Pentium IV 1.6GHz와 256MB 메모리가 장착된 컴퓨터상에서 실험을 수행하였다.

블록 정합의 정도를 평가하기 위해 대표적인 정합 기준인 평가 함수(cost function)로 영상 화질의 품질을 평가하기 위한 식 (2)의 평균 제곱 오차(MSE: Mean Squared Error)와 정합 오차 측정 함수로는 식 (3)의 절대 값 오차의 합(SAD: Sum of Absolute Difference)을 이용

하였다.

$$MSE(i, j) = \left(\frac{1}{N^2}\right) \sum_{k=1}^N \sum_{l=1}^N [I_r(k, l) - I_{r-1}(k+i, l+j)]^2 \quad (2)$$

$$SAD(i, j) = \sum_{k=1}^N \sum_{l=1}^N |I_r(k, l) - I_{r-1}(k+i, l+j)| \quad (3)$$

여기서 N은 영상의 가로와 세로의 각각의 크기이며,  $I_r$ 은 원영상의 화면을 나타내고,  $I_{r-1}$ 은 움직임 추정 화면을 나타내며, 이들 정합 기준들은 최소 값을 가지는 위치를 움직임 벡터로 결정한다. 그리고 화질의 평가를 위한 PSNR은 식 (4)와 같다.

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) \quad (4)$$

실험 영상에 대한 실험 결과는 <표 1>에 나타내었다. <표 1>은 각각의 실험 영상별로 기존의 탐색 기법과 제안하는 탐색 기법에 대한 평균 PSNR[dB]과 FS의 탐색회수를 1로 기준하여 각 기법의 상대적인 속도 향상에 대한 비교 결과를 나타낸다.

<표 1> 각 실험 영상에 대한 성능 비교 평가 함수의 결과 비교

Sequenc e	Motion Estimation Algorithm						
	FS	TSS	DS	NNS	ADZS	제안	
News	PSNR	38.23	38.10	38.15	38.11	38.20	38.21
	Speed Up	1	9	16	39	31	36
Miss America	PSNR	39.10	38.66	38.76	38.77	38.99	39.03
	Speed Up	1	8	12	28	23	27
Foreman	PSNR	33.46	32.79	33.17	32.67	33.28	33.31
	Speed Up	1	9	13	26	19	24
Akiyo	PSNR	42.78	42.65	42.76	42.72	42.78	42.78
	Speed Up	1	9	12	41	32	40
Coastguard	PSNR	29.64	29.32	29.41	29.40	29.51	29.53
	Speed Up	1	8	12	26	19	25
Stefan	PSNR	25.69	25.28	24.66	24.15	25.47	25.50
	Speed Up	1	8	12	24	18	21
평 균	PSNR	34.82	34.47	34.49	34.30	34.71	34.73
	Speed Up	1	8.5	12.8	30.7	23.7	28.8

<표 1>에서와 같이 제안된 탐색 기법은 움직임 추정 속도면이나 움직임 추정 정확도면에서 기존의 기법보다 더 나은 성능을 보였다. 전반적으로 제안된 탐색 기법은 FS에 비해 탐색 속도면에서는 약 25~40배 정도의 성능 향상을 나타내면서도 PSNR 값은 6개의 실험 영상의 평균이 0.15[dB] 차이로 근접한 성능을 나타내었다. 그리고 속도면에서는 NNS 기법이 우수하게 나타났으나 PSNR 값이 제안된 기법보다 평균 0.37[dB] 정도 나쁘게 나타난 것으로 일정한 고속 패턴 탐색으로 국부적 탐색에 의해 조기 수렴하는 최소화 문제에 빠지기 때문으로 예측의 정확성에서 효율적인 방식이 아님을 알 수 있었다.

이상의 실험 결과에 의하면, PSNR 측면에서 FS를 제외한 모든 기법에 비해 제안된 방식이 가장 우수한 것으로 나타났다. 또한 다른 모든 기법에 비해 움직임 추정에 필요한 탐색점 수

가 NNS와 유사하게 나타나 속도면에서 높은 향상을 보였으며, PSNR 값은 평균 0.02~0.37[dB] 개선되어 화질면에서도 안정된 예측 정확도를 얻을 수 있었다. 특히, 제안된 기법과 다이아몬드 지역 패턴을 이용한 ADZS와 비교할 때, PSNR 값은 거의 유사하게 나타난 반면, 속도가 14~24% 향상되었다.

### 5. 결 론

제안한 알고리즘은 현재 프레임 블록의 인접 블록에 의해 강인한 매크로 블록 단위의 움직임 벡터를 생성하였으며, 한 방향으로만 진행하는 수정된 다이아몬드 지역 패턴을 사용함으로써 적은 탐색 수만으로도 화질면에서 FS에 근접한 성능을 나타내었다. 또한 제안된 기법은 FS를 제외한 다른 비교 기법에서도 속도면에서나 화질면에서도 우수한 성능을 보였다. 이 경우 움직임 예측면에서도 평균적으로 0.02~0.37[dB] 정도의 성능 향상을 보였다.

수정된 다이아몬드 지역 패턴을 이용하여 현재 프레임의 매크로 블록 탐색 영역을 재정의하여 탐색 패턴을 적용적으로 변화시키면 다른 고속 블록 정합 방법들보다 탐색점 수를 감소시키고, 우수한 보상 결과를 얻을 수 있을 것으로 기대된다.

### 참 고 문 헌

[1] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated Interframe Coding for Video Conferencing", in Proc. National Telecommunications Conf., New Orleans, LA, pp.G5.3.1-G5.3.5, Nov. 1981.  
 [2] R. Li, Bing Zeng, "A New Three-Step Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Circuits & System for Video Tech., Vol. 4,

No. 4, pp.438-442, Aug., 1994.  
 [3] S. Zhu, K. K. Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Image Processing, Vol. 9, No. 2, pp.287-290, Feb., 2000.  
 [4] C. Zhu, X. Lin, L. P. Chau, "Hexagon-Based Search Pattern for Fast Block Motion Estimation", IEEE Transactions on Circuits & System for Video Tech., Vol. 12, No. 5, pp.349-355, May 2002.  
 [5] A. Tourapis, O. Au, M. Liou, G. Shen, "Optimizing the Mpeg-4 Encoder-Advanced Diamond Zonal Search", in Proc. of 2000 IEEE Inter. Symposium on Circuits and Systems(ISCAS-2000), Geneva, Switzerland, May, 2000  
 [6] M. Gallant, G. Cote, F. Kossentini, "An Efficient Computation-Constrained Block-Based Motion Estimation Algorithm for Low Bit Rate Video Coding", IEEE Transactions on Image Processing, Vol. 8, No. 12, pp.1816-1823, Dec., 1999.

### 곽 성 근

1980.2 연세대학교 대학원 졸업(공학석사)  
 2004.8 아주대학교 대학원 컴퓨터공학과(공학박사)  
 1980~현재 시립인천전문대학 컴퓨터정보과 교수  
 관심분야: 동영상압축, 컴퓨터

그래픽스, 애니메이션