

## 고속 블록정합 움직임 추정을 위한 최적의 탐색 패턴

임 동근(林東根), 호 요 성(扈堯盛)

광주과학기술원 정보통신공학과

전화 : (062) 970-2247, 팩스 : (062) 970-2204

E-mail : dklim@kjist.ac.kr, hoyo@kjist.ac.kr

### Optimal Search Patterns for Fast Block Matching Motion Estimation

Dong-Keun Lim and Yo-Sung Ho

Kwangju Institute of Science and Technology

#### Abstract

Motion estimation plays an important role for video coding. In this paper, we derive optimal search patterns for fast block matching motion estimation. By analyzing the block matching algorithm as a function of block shape and size, we can find an optimal search pattern for initial motion estimation. The proposed idea, which has been verified experimentally by computer simulations, can provide an analytical basis for the current MPEG-2 proposals. In order to choose a more compact search pattern for BMA, we exploit the statistical relationship between the motion and the frame difference of each block.

#### I. 서론

최근 무선 통신이나 인터넷을 이용한 실시간 멀티미디어 서비스에 대한 관심이 증가하고 있다. 전송 대역폭이 제한된 채널에서 멀티미디어 데이터를 효과적으로 전송하기 위해 여러 가지 압축 알고리즘이 사용되고 있다. 특히, 영상 신호에는 시간적으로 연속하는 화면 사이에 중복되는 정보가 많으므로, 이러한 중복 정보를 움직임 추정 기법을 사용하여 효과적으로 제거할 수 있다.

영상 신호에서 물체의 움직임을 추정하기 위해 가장 많이 사용되는 블록정합 알고리즘에서는 각 화면을 사각형 블록으로 나누고, 정합 기준(matching criteria)에 따라 이전 화면의 탐색 영역에서 현재 화면의 각 블록과 가장 비슷한 블록을 찾아낸다. 선택된 블록은 현재 블록

을 예측하는데 사용하며, 두 블록 사이의 상대적인 위치 차이는 현재 블록의 움직임 벡터로 표현된다.

전역탐색 방법은 오차 함수값을 최소화하는 움직임 벡터를 구하기 위해 가능한 모든 경우를 탐색하기 때문에 일반적으로 좋은 성능을 나타낸다. 그러나, 탐색 영역 전체에서 적합한 움직임 벡터를 찾기 때문에 많은 계산이 필요하다. 이러한 계산량을 줄이기 위해 여러 종류의 고속 탐색 방법들이 제안되었으나, 대부분의 고속 탐색 방법들은 경험예측적인(heuristic) 접근 방법을 사용하여 계산량을 줄이기 때문에 복원된 영상의 화질을 다소 저하시킨다[1-4].

본 논문에서는 움직임 벡터를 찾는데 요구되는 계산량을 줄이도록 블록정합 알고리즘을 위한 최적의 탐색 패턴을 수식적으로 유도하였고, 실제적인 움직임 정보를 살펴봄으로써 제안한 방법이 옳다는 것을 입증하였다. 제안한 방법은 사각형 블록뿐만 아니라, MPEG-4 표준에서와 같이 임의의 형태를 가진 블록을 사용하는 블록정합 방법에서도 최적의 탐색 패턴을 예측할 수 있다. 또한, 이를 이용하여 효과적인 움직임 추정을 할 수 있다는 것을 모의실험을 통해 보여준다. 그 결과로서 시간적으로 인접한 두 화면 사이에서 물체 움직임의 통계적인 특성을 고려하여 새로운 블록정합 움직임 추정 방법을 제안하였다. 각 블록의 움직임 정도와 예측 오류 사이의 관계를 이용하여 해당 블록에 적절한 탐색 패턴을 가변적으로 설정하였다[5]. 탐색 패턴을 적응적으로 변화시켜 다른 고속 블록정합 알고리즘보다 계산량은 감소하였고, 움직임 예측 성능은 향상되었다.

## II. 최적의 탐색 패턴

블록정합 방법을 수학적으로 분석하기 위해, 그림 1과 같은 문제를 생각해 보자. 이것은 블록 정합을 수행할 때 어떤 형태의 탐색 패턴을 취하는 것이 계산량과 최적도(optimality)의 관점에서 유리한 지를 예측하기 위한 것이다.

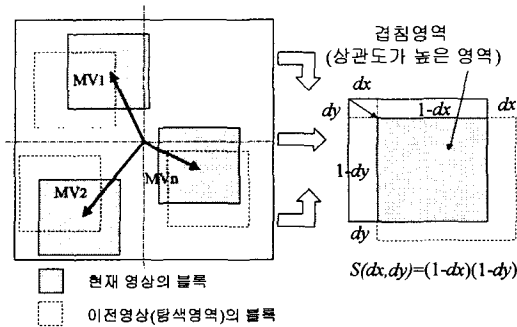


그림 1. 탐색 패턴에 대한 문제 설정  
Fig. 1. Problem Setting for Search Pattern

일반적으로 블록 정합의 평가함수(similarity measure)로서 상관도(correlation)를 사용하는데, 그림 1에서 겹쳐진 부분이 상관도의 정도를 나타낸다. 여기서 고려하는 사항은 움직임 방향을 탐색하는 것이 아니라, 움직임 탐색 단계에서 최소 개수의 탐색 후보를 효과적으로 결정하는 데 있다.

그림 1에서 고려되고 있는 블록은 모든 방향으로 움직일 수 있다. 블록의 위치를 중심으로 놓여 있는 이전 화면의 탐색 영역에서 블록의 겹침 영역은 상관도를 나타내며, 그 면적은 다음과 같이 계산된다.

$$S(dx, dy) = (1-dx)(1-dy), \quad 0 \leq dx \leq 1, \quad 0 \leq dy \leq 1 \quad (1)$$

그림 1로부터  $S(dx, dy)$ 는 다음과 같은 대칭성(symmetry)을 가진다.

$$S(dx, dy) = S(-dx, dy) = S(dx, -dy) = S(-dx, -dy) \quad (2)$$

식 (1)로부터  $(dx, dy)$ 를 구할 수 있는데, 이것은 동일한 상관도를 가진 등고선을 의미한다.

$$dx \cdot dy - dx - dy + 1 - S(dx, dy) = 0 \quad (3)$$

식 (3)을 풀기 위해  $dx$ 와  $dy$ 의 가능한 관계를 고려할 수 있으며, 다음과 같은 선형 관계를 사용할 수 있다.

$$dy = f \cdot dx, \quad f \in R \quad (4)$$

이것은 블록 내에 있는 모든 위치를 쉽게 표현할 수 있기 때문에 적절한 선택이다.

식 (4)를 식 (3)에 대입하고 정리한 후, 식 (1)과 식

(2)의 조건을 고려하면 다음과 같은  $(dx, dy)$ 의 관계식을 구할 수 있다.

$$dx = \frac{(f+1) - \sqrt{(f+1)^2 - 4f \cdot S(dx, dy)}}{2f} \quad (5)$$

$$dy = \frac{(f+1) - \sqrt{(f+1)^2 - 4f \cdot S(dx, dy)}}{2} \quad (6)$$

이와 같이 이론적으로 구한 식의 신뢰도를 검사하기 위해 실제 영상의 움직임 패턴과 비교하였다. 블록 정합을 위해  $16 \times 16$  크기의 블록을 사용하고, 탐색 영역의 크기는  $\pm 7$ 로 잡았다. 즉,  $0 < |dx|, |dy| < 7/16 \approx 1/2$ 로 설정하였다. 식 (5)와 식 (6)에서  $S(dx, dy)$ 가  $1/2$ 보다 작은 경우에 구한 탐색 패턴을 그림 2(a)에 보였다.

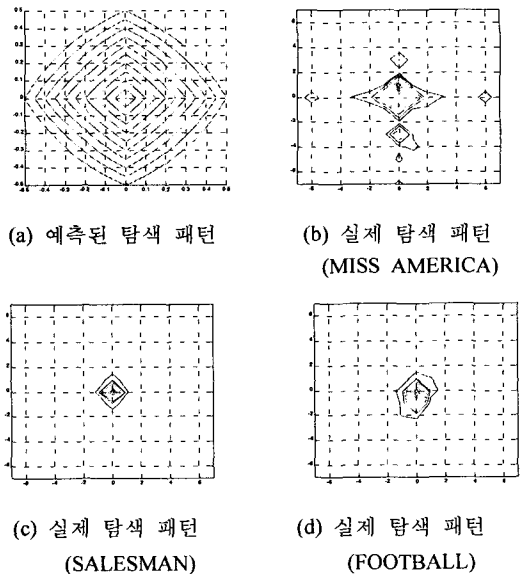


그림 2. 탐색 패턴의 비교

Fig. 2. Comparison of Search Patterns

실제 동영상에서 움직임 벡터의 분포를 살펴보면, 그림 2에 보인 것과 같이, 대부분 중심 부근에 집중되어 있다. 특히, 수평과 수직선상으로 많은 움직임이 있음을 알 수 있다. 실제 탐색 패턴과 이론적인 탐색 패턴은 다이하몬드와 비슷한 형태를 띠는 등고선임을 알 수 있다.

## III. 최적 탐색 패턴을 사용한 움직임 추정

고속 블록정합 방법에서 고려할 중요 사항들은 다음과 같다[5].

- 지역적 최소값에 빠지지 말아야 한다.
- 정확한 움직임 방향으로 탐색을 진행하기 위해 초기 탐색점이 충분히 많이 배치되어야 한다.

c) 다음 단계에서 추가되는 탐색점의 개수는 감소해야 하며, 이전 단계에서 검사된 탐색점은 제외되어야 한다.

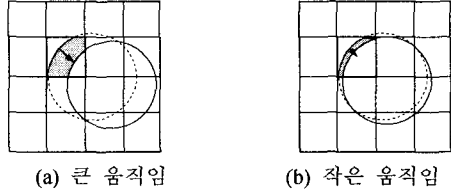


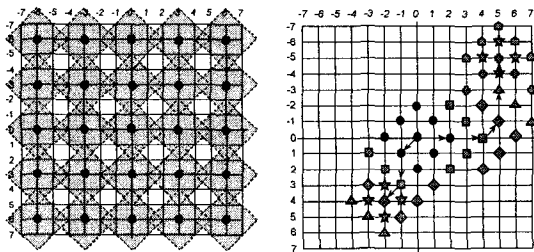
그림 3. 블록의 움직임과 BFD의 상호관계

Fig. 3. Block Motion and BFD

그림 3에 보인 것과 같이, 블록의 큰 움직임은 탐색 영역 안에서 일반적으로 큰 블록차이 (Block Frame Difference, BFD)를 가진다. 그렇지만 다음과 같은 몇 가지 예외 상황이 발생할 수 있다.

경우 1: BFD 값은 임계값보다 작지만, 움직임이 큰 경우로 판정될 수 있다. 이 경우는 정적 영역에 잡음과 같은 것이 존재하므로, BFD를 관측하여 이 블록들은 작은 움직임을 가진다고 판단한다.

경우 2: BFD 값은 임계값보다 크지만, 움직임이 작은 경우로 판정될 수 있다. 이런 경우는 배경과 물체 사이의 화소값의 차이가 큰 경우에 발생할 수 있다. 만약 블록을 특징짓는 화소의 개수가 적으면 이 경우가 발생했다고 예상할 수 있고, 해당 블록은 작은 움직임을 가진다고 판단한다.



(a) 초기 탐색 방법 (b) 기본적인 탐색 방법

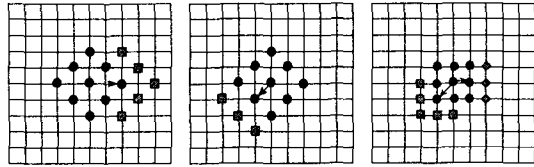
그림 4. 제안된 탐색 방법

Fig. 4. Proposed Search Method

BFD 값이 임계값보다 크면 탐색을 수행하기 전에 그림 4(a)와 같은 초기 탐색 패턴을 사용한다. 이 그림에서 각 탐색 패턴은 초기 탐색에서 빠진 위치를 다음 단계에서 점검한다. 만약 다음 단계에서 초기 탐색에서 찾은 최소값보다 더 작은 값을 갖는 위치를 찾게 되면, 그 위치가 그림 4(b)에 있는 탐색 중심의 새로운 후보가 된다.

그림 4(b)는 최대 탐색 영역의 크기가 7일 때, 탐색 방법에 대한 두 가지 예를 보여주고 있다. 그림 4(b)는 탐색이 다음 단계로 진행됨에 따라 3개 혹은 5개의 후보

탐색점이 추가된다. 그림 4(b)의 오른쪽 예에 보인 것처럼, 탐색이 진행되면서 탐색 방향을 급격히 변화할 수도 있다. 즉, 만약 지역 최소값에 빠지지 않으면 탐색 영역 전체에서 최소값을 찾게 된다.



(a) 큰 움직임A (b) 큰 움직임B (c) 작은 움직임

그림 5. 움직임 블록에 대한 탐색 패턴

Fig. 5. Search Patterns for Block Motion

그림 5는 큰 움직임과 작은 움직임에 따라 다른 방법으로 적용되는 탐색 패턴을 보여준다.

제안한 방법은 다음과 같이 정리할 수 있다.

단계 1: 원점(0,0)을 중심으로 9개의 탐색점에 대해 최소값을 찾는다. 탐색 패턴은 수평과 수직 방향으로 2점 간격을 두고 배치되며, 대각선상에 있는 다른 점들은 1점 간격으로 배치된다. 만약 원점에서 최소값이 발견되면 단계 3을 수행한다.

단계 2: 이 단계에서는 이전 단계의 최소값을 가지는 점의 위치에 따라서 두 종류의 탐색 패턴을 가진다. 이 단계는 현재 탐색 패턴의 원점에서 최소값이 될 때까지 반복적으로 수행된다.

- a) 만약 이전 단계에서 최소값을 가지는 점이 이전 탐색 영역의 수평 혹은 수직 방향에 있다면, 그 점을 따라서 5개의 새로운 탐색점이 다음 단계의 후보 탐색점이 된다 [그림 5(a)].
- b) 만약 이전 단계에서 최소값을 가지는 점이 이전 탐색 영역의 대각선 방향에 있다면, 그 점을 따라서 3개의 새로운 탐색점이 다음 단계의 후보 탐색점이 된다 [그림 5(b)].

단계 3: 이전 단계에서 찾은 탐색점을 중심으로 1점 간격으로 주변 9개의 탐색점 중에서 최소값을 가지는 점의 위치를 찾는다.

#### IV. 실험 결과

제안된 방법을 검증하기 위해 ITU-T 시험 영상인 MISS AMERICA, SALESMAN, CLAIRE, FOOTBALL 영상을 각각 88 프레임씩 사용하여 컴퓨터 모의시험을 수행하였다. 각 영상은 CIF 형식 (352×288 화소)의 휘도 성분을 가진다. 프레임율은 원래는 30 Hz인데, 모의시험에서는 10 Hz로서 1번째, 4번째, ..., 88번째 프레임을 사용하였고, 영상 화질은 식 (7)의 PSNR (peak signal-to-noise ratio)을 사용하여 비교하였다.

$$MSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (I_i(m,n) - \tilde{I}_i(m,n))^2 \quad (7)$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad [dB] \quad (8)$$

여기서  $I_i$  는 원래 현재 화면을 나타내고,  $\tilde{I}_i$  는 움직임 보상 예측된 현재 화면을 나타낸다. CIF 형식에 대해서  $M = 352$ ,  $N = 288$ , Block Size = 16이다.

그림 6은 MISS AMERICA 영상에 대해 수행된 결과를 나타내고 있는데, 제한된 방법이 FS (full search), TDL (two-dimensional logarithmic search), TSS (three-step search), 4SS (four-step search), 그리고 DSWA (dynamic search window adjust and interlaced search) 방법들[1-3]과 비교되어 있다.

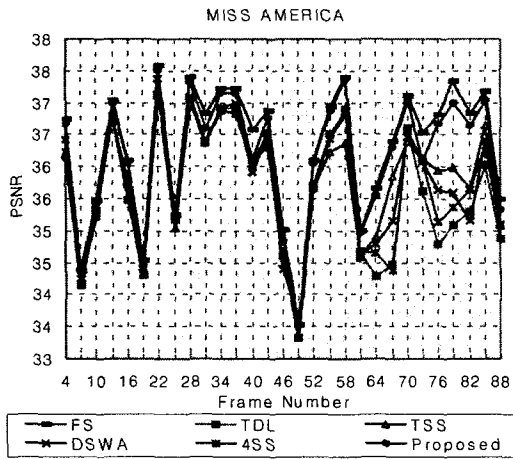


그림 6. 성능 비교

Fig. 6. Performance Comparison

제한된 방법과 전역 탐색 방법의 예측 결과는 70번째의 화면을 복원할 때까지는 거의 동일하지만, 움직임이 많은 70번째에서 80번째까지의 화면을 복원할 때에는 전역 탐색 방법의 예측 결과보다 약간 떨어진다. 이에 비해, 다른 고속 블록 정합 방법들의 성능은 전역 탐색 방법의 예측 결과와 상당히 큰 차이를 가진다. 즉, 제안한 방법이 움직임이 많은 화면들의 예측에서 보다 향상된 성능을 가짐을 알 수 있다.

표 1은 이러한 결과를 정리한 것인데, 제안된 방법이 전역 탐색 방법에 매우 근사한 움직임 예측 성능을 가지면서, 다른 고속 탐색 방법보다 평균 탐색점 수를 많이 줄임을 알 수 있다.

### V. 결론

본 논문에서는 매크로블록 단위의 블록 정합 움직임 예측에서 최적의 탐색 패턴이 다이아몬드 형태임을 이론적으로 유도하고, 실제 움직임 분포와 비교하였다.

표 1. 블록정합 방법들의 성능 비교

Table 1. Performance Comparison

움직임 탐색 방법	PSNR(dB)	탐색점 수
전역탐색(FS)	36.29	225
3단계 탐색(TSS)	35.78	25
4단계 탐색(4SS)	35.81	20.91
2차원 로그 탐색(TDL)	35.62	16.62
동적영역 가변탐색(DSWA)	35.77	19.79
제안한 방법	36.17	14.78

(주의) 블록크기 = 16, 탐색영역 = [-7, +7]

제안한 분석을 이용하면 여러 가지 다른 형태의 블록 구조를 사용할 때에도 효과적인 탐색 패턴을 유도할 수 있다. 본 논문에서는 새로운 정합 방법을 사용한 고속 블록정합 방법을 제안하였으며, 각 블록의 움직임을 관측하여 서로 다른 탐색 패턴을 적용하였다. 결과적으로 새로운 알고리즘은 필요한 계산량을 효과적으로 줄이고, 우수한 예측 성능을 보였다.

### 감사의 글

본 연구는 광주과학기술원(K-JIST) 초고속광네트워크연구센터(UFON)를 통한 한국과학재단 우수연구센터(ERC)와 교육부 두뇌한국21(BK21) 정보기술사업단의 지원에 의한 것입니다.

### 참고문헌

- [1] H.G. Musmann, P. Pirsh and H.J. Grallert, "Advances in Picture Coding," *Proc. IEEE*, vol. 73, no. 4, pp. 523-548, Apr. 1985.
- [2] L.M. Po and W.C. Ma, "A Novel Four-step Search Algorithm for Fast Block Motion Estimation," *IEEE Trans. Circuit and Systems for Video Technology*, vol. 6, no. 3, pp. 313-317, June 1996.
- [3] L.W. Lee, et al., "Dynamic Search-window Adjustment and Interlaced Search for Block Matching Algorithm," *IEEE Trans. Circuit and Systems for Video Technology*, vol. 3, no. 1, Feb. 1993.
- [4] T. Chiang and H. Sun, Report of Ad Hoc Group on Encoder Optimization, *ISO/IEC/JTC1/SC29/WG11 MPEG99/M5528*, Dec. 1999.
- [5] D.K. Lim and Y.S. Ho, "A Fast Block Matching Motion Estimation Algorithm based on Statistical Properties of Object Displacement," *IEEE Region 10 Annual Conference*, pp. 138-141, Dec. 1998.