

이전 프레임의 움직임 정보와 탐색 구간별 예측 후보점을 이용하는 블록 정합

(A Block Matching using the Motion Information of Previous
Frame and the Predictor Candidate Point on each Search Region)

곽 성 근 * 위 영 철 ** 김 하 진 **

(Sung-Keun Kwak) (Youngcheul Wee) (Ha-Jine Kimn)

요 약 동영상의 현재 블록의 움직임 벡터와 이전 블록의 움직임 벡터는 시간적 상관성을 갖고 있다. 본 논문에서는 영상의 시간적인 특성과 움직임 벡터의 가운데 중심 분포 특성을 이용하는 예측 탐색 알고리즘을 제안한다. 제안된 알고리즘은 이전 프레임 블록으로부터 예측된 움직임 벡터와 분할된 탐색 구간에 속하는 후보 벡터 중에서 가장 작은 SAD 값을 갖는 점을 정확한 움직임 벡터를 찾기 위한 초기 탐색점 위치로 결정한다.

실험 결과 제안된 방식은 FS를 제외한 기존의 대표적인 고속 탐색 방식들에 비해 PSNR 값에 있어서 평균적으로 0.19~0.46dB 개선되고 영상에 따라 최고 1.06dB 정도 우수한 결과를 나타내었다.

키워드 : 예측 후보점, 블록 정합 알고리즘

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 prediction search algorithm for block matching using the temporal correlation of the video sequence and the center-biased property of motion vectors.

The proposed algorithm determines the location of a better starting point for the search of an exact motion vector using the point of the smallest SAD(sum of absolute difference) value by the predicted motion vector from the same block of the previous frame and the predictor candidate point on each search region.

Simulation results show that PSNR(Peak-to-Signal Noise Ratio) values are improved up to the 1.06dB as depend on the video sequences and improved about 0.19~0.46dB on an average except the full search(FS) algorithm.

Key words : Predictor Candidate Point, Block Matching Algorithm

1. 서 론

디지털 비디오 신호 전송에서 프레임간의 시간적 중복성을 줄이기 위한 블록 정합 움직임 추정은 알고리즘의 단순성과 고속 처리를 위한 하드웨어적인 접근과 용이성으로 인해 MPEG 계열이나 H.261, H.263 등의 표준안으로 채택되어 많이 사용되고 있다.

일반적으로 움직임 추정 기법은 영상의 각 단위로 블록 또는 화소 단위로 적용되며, 이중 계산 복잡도 및 하

드웨어 구현에 있어서 용이한 블록 단위의 움직임 추정이 널리 사용되고 있다. 블록 단위의 움직임 추정은 동일한 블록 내의 화소들은 동일한 움직임을 갖는 것과 블록들은 수평, 수직으로만 움직인다는 두 가지 전체 조건을 가지므로 영상의 한 프레임을 동일한 크기의 블록들로 나누고 이들의 각 블록들에 대하여 참조 프레임(reference frame)의 탐색 영역 내에서 정합 오차가 가장 작은 블록간의 위치 차이를 구하여 이를 움직임 벡터(motion vector)로 추정한다.

이와 같이 블록 단위로 움직임을 추정하는 대표적인 알고리즘이 FS(Full Search)이다. 이것은 영상을 모양과 크기가 동일한 사각형 블록으로 분할한 후, 블록 단위의 특징을 정의하고 정의된 특징에 의해 구성되는 정합 척도를 일정한 탐색 영역 내의 블록들에 적용하여

* 정 회 원 : 시립인천전문대학 컴퓨터정보과 교수
skkwak@icc.ac.kr

** 송신회원 : 아주대학교 정보및컴퓨터공학부 교수
ycwee@ajou.ac.kr
hjkimn@ajou.ac.kr

논문접수 : 2003년 10월 13일

심사완료 : 2004년 1월 19일

가장 높은 정합 척도를 가지는 블록을 찾는다. 그러나 전역 탐색 블록 정합 알고리즘은 과정이 간단하고 예측 효율과 추정의 정확도를 고려할 때 전체적으로 좋은 특성을 가지며 하드웨어 구현이 용이하고 또한 탐색 영역의 내부 전체를 탐색하면서 가능한 모든 블록들에 대한 정합을 수행하므로 정합 오차가 가장 적은 움직임 벡터를 찾을 수 있지만 많은 계산량이 필요한 단점이 있다.

이러한 전역 탐색법의 단점을 극복하기 위해 속도가 개선된 TSS(Three Step Search)[1], NTSS(New Three Step Search)[3], FSS(Four Step Search)[4], BBGDS(Block-Based Gradient Descent Search)[5], DS(Diamond Search)[6], HS(Hexagon-based Search)[8] 등의 다양한 고속 블록 정합 알고리즘(FBMA: Fast Block Matching Algorithm)이 개발되었다. 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 오차를 검사하는 탐색점들을 의미하며, 이 탐색점들 중에서 최소 정합 오차를 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. 그러나 이러한 고속 블록 정합 알고리즘은 계산량을 줄이기 위해 탐색 영역에 포함되는 특정한 패턴들의 몇몇 점들만 조사하여 움직임 벡터를 찾기 때문에 국부적인 탐색을 하게 되거나, 해당 블록에 대한 움직임 정보를 가지고 있지 않기 때문에 항상 탐색 영역의 원점에서부터 움직임을 추정하는 과정에서 일부 탐색점들을 블록 정합 대상에서 제외시키거나 부정확한 방향으로 움직임을 추정함으로써 복원된 영상의 화질을 저하시키는 단점이 있다.

이러한 기존의 고속 블록 정합 알고리즘의 단점을 개선하기 위해 이전 프레임의 같은 위치의 블록 또는 인접한 이전 블록들의 움직임 벡터들의 정보를 이용하는 예측 탐색 알고리즘(PSA: Prediction Search Algorithm)이 제안되었다[7,9,10]. 이 방식은 이전 블록들의 움직임 정보를 이용함으로써 보다 정확한 움직임 벡터를 찾으려는 것이다. 그러나 이 방식은 참조한 블록의 움직임 벡터들의 상관성이 떨어질 경우 압축 성능이 현저히 떨어지는 단점이 있다.

일반적으로 움직임이 있는 동영상의 경우 짧은 시간에 영상의 움직임이 있더라도 그 시간에는 많은 움직임이 있을 수 없으므로 인해서 움직임 벡터는 중심 지향적인 분포 특성을 갖게 된다. 또한 현재 매크로 블록의 움직임 벡터는 이전 프레임의 같은 위치에 있는 블록들과 상관관계가 높으며, 비슷한 운동을 한다.

따라서 본 논문에서는 이전 프레임 블록의 움직임 정보를 이용하여 현재 영상의 움직임 후보 벡터를 구한다. 이때 영상의 국부적인 변화에 따른 움직임을 효과적으로 반영하기 위해 예측된 움직임 벡터가 속하는 탐색 구간에 따라 중심 지향적인 분포 특성에 알맞은 후보 벡터를 또 하나 설정한다. 그리고 두 개의 후보 벡터에 대한 정합 오차(BDM: Block Distortion Measure)를 비교하여 더 작은 값을 갖는 후보 벡터로부터 탐색을 수행하여 움직임 벡터의 상관성을 높임으로써 효과적으로 탐색점 수를 줄이고, 또한 기존의 탐색 패턴을 이용한 알고리즘의 국부적 최소(local minimum)로 인한 시각적인 화질 저하의 문제점을 개선하는 탐색 기법을 제안한다.

본 논문의 구성은 다음과 같다. 제2장에서 기존의 예측 탐색 알고리즘에 대해 설명하고, 제3장에서는 제안한 알고리즘을 위한 예측된 움직임 벡터의 설정 방법과 구현에 대한 방법을 논하며, 제4장에서는 실험 결과에 대하여 기존 방법과 같은 기준을 통해서 비교 분석하고, 제5장에서는 결론 및 향후 연구 과제를 제시한다.

2. 기존의 예측 탐색 알고리즘

최근에는 기존의 고속 탐색 알고리즘의 단점을 개선하기 위해 이전에 탐색된 움직임 벡터들의 정보를 이용하는 예측 탐색 알고리즘(PSA: prediction search algorithm)이 제안되었다[9,10]. 이러한 예측 알고리즘에서 예측된 움직임 벡터는 일반적으로 현재 블록과 같은 위치에 있는 이전 시점의 블록이나 현재 프레임에서 추출된 주변 블록들의 움직임 벡터를 이용하여 정의된다. 그리고 이렇게 정의된 움직임 벡터를 결정하여 블록 정합을 위한 탐색 패턴으로 TSS 또는 DS 방법을 적절히 적용하고 있다.

2.1 시간 상관적인 예측 방법

이전 프레임과의 시간적 상관관계를 적용하여 예측하는 움직임 벡터인 TPMV(Temporal Predicted Motion Vector)는 영상의 움직임은 급격히 변화하기보다는 완만하게 변화한다는 연속성 제약 사항(motion continuity constraint)[2]에 근거하여 이전 프레임의 같은 위치에 있는 블록의 움직임 벡터를 참조한다. 즉, 탐색 영역에서 탐색 원점의 위치를 참조 프레임의 예측된 움직임 벡터(predicted motion vector)만큼 이동시켜 새로운 탐색 원점을 설정한 후 블록 정합을 수행한다.

$$MV_T = MV_{t-1} \quad (1)$$

그리고 Tourapis는 연속된 프레임간 움직임의 변화가 서로 상관성이 있다는데 착안하여 그림 1과 같이 이전 두 개의 프레임의 같은 위치에 있는 블록의 움직임 벡

터의 변화를 참조하여 움직임 벡터 MV_T 를 식 (2)와 같이 예측 적용하였다[9]. 이전 프레임 블록의 움직임 벡터의 이동 거리를 고려한 이 방식은 연속된 프레임간 움직임 벡터의 변화가 커질 경우에 MV_T 가 탐색창의 경계 범위를 벗어날 수 있으므로 이러한 문제를 전처리할 필요가 있다.

$$MV_T = MV_{t-1} + (MV_{t-1} - MV_{t-2}) \quad (2)$$

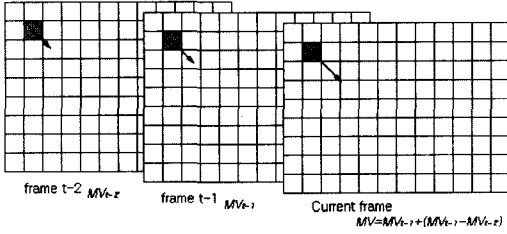


그림 1 참조하는 이전 프레임의 같은 위치의 매크로 블록

2.2 공간 상관적인 예측 방법

현재 프레임의 블록들간의 공간적인 상관관계를 반영하여 예측하는 움직임 벡터인 SPMV(Spatial Predicted Motion Vector)는 공간적으로 인접한 매크로 블록들은 비슷한 속도로 거의 같은 방향으로 움직인다는 점을 고려하여 그림 2와 같이 수평, 수직 그리고 대각 방향에 있는 매크로 블록 MB_0, MB_1, MB_2 을 참조한다[7]. 이와 같은 NNS(Nearest Neighbour Search)의 현재 블록의 예측 움직임 벡터는 식 (3)과 같이 이들 인접 매크로 블록 각각의 움직임 벡터 MV_0, MV_1, MV_2 들의 중간 값으로 한다.

$$MV_S = median(MV_0, MV_1, MV_2) \quad (3)$$

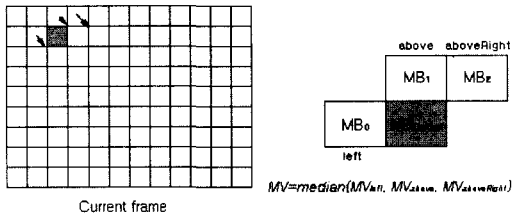


그림 2 참조하는 인접 매크로 블록

그림 2에서 현재 블록의 인접 매크로 블록(adjacent macroblock)인 MB_0, MB_1, MB_2 에서 예측 움직임 벡터 MV_p 를 결정할 수 있다. 인접 블록 MB_i (MB_i for $i \in \{1, 2, 3\}$)의 움직임 벡터를 $V_i = [\Delta c_i, \Delta r_i]$ 라고 할 때, 현재 매크로 블록 MB_c 의 예측 움직임 벡터는 $MV_p = [\Delta c_p, \Delta r_p]$ 가 된다. 예를 들어, 블록 MB_1 와

MB_2 의 예측 벡터인 Δc_1 과 Δc_2 값이 Δc_p 와 유사하다고 가정한다면 식 (4)와 같다.

$$\Delta c_p = \frac{(\Delta c_0 + \Delta c_1)}{2} \quad (4)$$

그리고 Δr_p 도 식 (4)와 같은 방법으로 구한다. 따라서 현재 블록의 좌표가 (m, n) 이면 탐색의 초기 중앙점의 좌표는 $(m + \Delta c_p, n + \Delta r_p)$ 이다.

이러한 예측 탐색 기법은 시공간적인 상관관계를 적용하여 속도뿐만 아니라 PSNR 및 시각적인 측면에서 좋은 결과를 보인다고 알려져 있다[10]. 그러나 예측된 움직임 벡터를 탐색 원점으로 이용하여 빠르게 탐색을 수행함으로써 속도면에서는 좋은 결과를 보일 수 있으나, 움직임 벡터의 상관성이 떨어질 때는 잘못된 움직임을 추정하여 영상의 화질이 크게 저하될 수 있다.

3. 제안 탐색 알고리즘

여기서는 기존의 예측 탐색 알고리즘이 예측된 움직임 벡터로 탐색 원점을 이동시켜 탐색을 수행함으로써 블록 내에 공존하는 최소 정합 오차를 가지는 탐색점을 제외시켜 잘못된 추정을 하는 문제점을 개선하기 위해 이전 움직임 정보에 의해 예측된 움직임 벡터와 분할된 탐색 구간에 속하는 후보 벡터 중에서 가장 작은 정합 오차를 갖는 점을 탐색 원점으로 하여 탐색을 수행하는 알고리즘을 제안한다.

이때 블록 정합 오차를 결정하는 평가 함수로 SAD (Sum of Absolute Difference)을 사용한다.

$$SAD(i, j) = \sum_{k=1}^N \sum_{l=1}^N |I_c(k, l) - I_{t-1}(k+i, l+j)| \quad (5)$$

식 (5)에서 N은 영상의 가로와 세로의 각각의 크기이며, $I_c(k, l)$ 은 원영상의 화면을 나타내고, $I_{t-1}(k+i, l+j)$ 은 움직임 추정 화면을 나타낸다.

3.1 움직임 벡터 예측 방법

본 논문에서 제안하는 알고리즘에서는 2개의 움직임 벡터 예측 후보(motion vector predictor candidate)를 구하여 이들의 SAD를 비교하여 가장 작은 값을 갖는 점으로 탐색 원점을 이동시켜 블록 정합을 수행한다.

첫 번째 움직임 벡터 예측 후보(first predictor)는 이전 프레임의 같은 위치에 있는 블록의 움직임 벡터를 참조하여 t-1시점에서 추출한 움직임 벡터를 이용한다. 즉, 그림 3과 같이 이전 프레임(t-1)의 같은 위치에 있는 블록 MB_{past} 와 좌, 우, 상, 하의 인접 매크로 블록 MB_0, MB_1, MB_2, MB_3 들을 포함한 5개의 움직임 벡터의 평균으로 구한다.

$$MV_p = \frac{MV_{past} + MV_{left} + MV_{right} + MV_{up} + MV_{down}}{5} \quad (6)$$

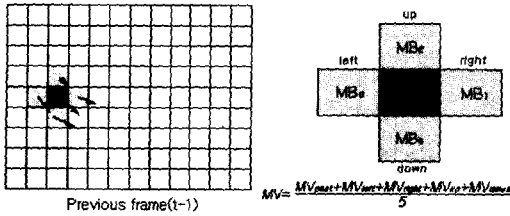


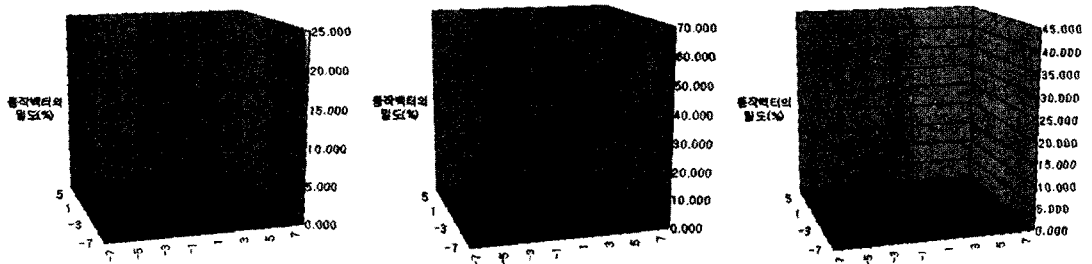
그림 3 이전 프레임의 인접 매크로 블록

두 번째 움직임 벡터 예측 후보(second predictor)는 첫 번째 움직임 벡터 예측 후보의 잘못된 예측으로 블록 정합 대상에서 최소 정합 오차를 갖는 탐색점이 제외될 수 있는 단점을 보완하기 위해 탐색창의 중심점인 (0,0)에 가까운 위치에 설정한다. 후보 벡터의 설정을 위해 그림 5(a)와 같이 움직임을 추정할 수 있는 I(horizontal), II(vertical), III과 IV(diagonal)의 4가지 가능한 초기 방향을 적용하여 그림 5의 (b)와 같이 탐색창을 9개의 탐색 구간으로 구분한다. 각 구간의 후보 벡터의 위치는 식 (6)으로 생성된 예측된 움직임 벡터가 속하는 영역으로 한다. 이때 구간별 후보점들의 위치는 해당 구간내의 중심점으로 설정할 수 있으나, 이전 움직임 벡터의 거리 오차와 움직임 벡터가 탐색창의 중심점

을 중심으로 반경 2화소 이내에 분포할 확률이 약 56.72%~98.70%[6]이라는 중심 지향적인 분포 특성(center-biased property)을 고려하여 탐색 영역의 중심점 (0,0)에서 예측된 움직임 벡터가 속한 구간 L로 향하는 점들 중에서 L₀내의 끝점으로 함으로써 움직임 벡터를 찾는 확률을 높인다.

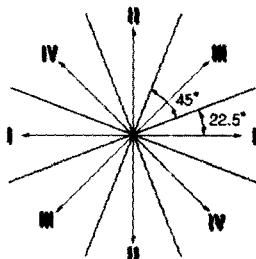
그림 4는 FS 알고리즘을 이용한 실험 영상의 움직임 벡터 분포의 예를 보인 것으로 얼굴을 흔들며 크게 웃으면서 말을 하는 "Foreman"과 어머니와 딸이 이야기를 나누는 "Mother & Daughter"는 그림 4의 (a), (b)와 같이 (0,0) 좌표를 중심으로 수직과 수평으로 중심 부근에 움직임 벡터가 집중되어 고르게 분포되어 있으며, 빠르게 수평으로 이동하는 보트를 카메라가 따라가고 있는 "Coastguard"는 그림 4의 (c)와 같이 전체적인 움직임이 수평 방향으로 발생하고 있으나 대부분 움직임 벡터가 탐색 영역의 (0,0) 좌표를 중심으로 집중되어 있음을 볼 수 있다.

그림 5의 (b)는 예측된 움직임 벡터가 속하는 각 구간별 후보점 위치를 나타낸다. 이때 분할된 탐색 구간의 크기를 s라 할 때, 중심점에서 각 탐색 구간과의 거리는 $d = \int (s/2)$ 로 이 경우 $s=5, d=2$ 이므로 탐색창의 가운

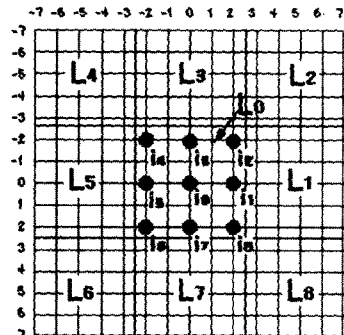


(a) Foreman (b) Mother & Daughter (c) Flower Garden

그림 4 FS 알고리즘을 이용한 실험 영상의 움직임 벡터 분포



(a) 탐색 영역내의 가능한 초기 방향



(b) 각 구간과 구간별 후보점

그림 5 구간별 후보점의 설정을 위한 탐색창의 구간 분할

데 구간 L_0 의 5×5 에서 2화소 간격의 정사각형 형태로 분포하는 9개의 점을 탐색 구간 각각의 후보 벡터로 설정한다. 즉, 블록 정합을 위한 탐색 영역의 크기를 $(2w+1) \times (2w+1)$ 로 설정하고, 이전 프레임의 블록의 움직임 벡터가 MV_{p1} 이고 이 움직임 벡터가 속하는 탐색 구간이 L_i 라 하면 구간별 탐색 후보점은 탐색 구간 L_i 의 대표점인 I_i 가 된다. 이때 각 구간의 후보점 위치를 (i, j) 라 할 때, 움직임 벡터를 추정하기 위한 후보 벡터의 위치는 식 (7)과 같다.

$$MV_{p2} \leftarrow I_i(i, j) \leftarrow L_i(MV_{p1}), \quad (7)$$

$$-w \leq i \leq w, \quad -w \leq j \leq w$$

그리고 생성된 2개의 후보 벡터의 SAD를 비교하여 가장 작은 값을 가지는 벡터를 최종적으로 탐색 원점으로 결정한다.

$$MV_p = \min SAD(MV_{p1}, MV_{p2}) \quad (8)$$

3.2 제안된 탐색 알고리즘에 의한 움직임 추정

이동된 탐색 원점에서 탐색 영역의 가장자리로 점진적인 탐색을 해 나갈 때 탐색 영역의 중심에 있는 일부 탐색점들을 블록정합 대상에서 제외되어 화질의 저하를 초래하는 탐색 후보점 사이의 간격이 있는 패턴을 사용하지 않고, 그림 7과 같이 탐색 후보점간의 간격이 없는 3×3 화소의 정사각형 블록 패턴으로 탐색을 수행하여 모든 탐색점들이 블록 정합의 대상이 되도록 함으로써 움직임 추정의 성능을 높인다.

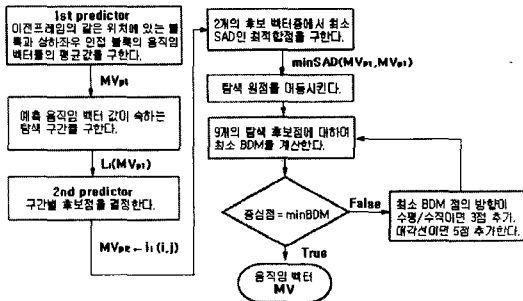


그림 6 제안한 움직임 벡터 추정 방식의 흐름도

그림 6은 제안한 알고리즘의 순서를 간단히 나타낸 흐름도로 제안한 알고리즘에서는 이전 참조 프레임에서 구한 움직임 예측 벡터 MV_{p1} 와 이 벡터가 속하는 탐색 구간 L_i 의 후보점 I_i 의 벡터 MV_{p2} 의 SAD를 비교하여 가장 작은 값을 가지는 점을 구한 후, 이 점을 중심으로 추가된 8개를 포함한 9개의 탐색 후보점에 대하여 다음과 같은 알고리즘을 수행하여 움직임 벡터를 추정한다.

1단계: 현재 블록과 같은 위치에 있는 이전 시점의

프레임 블록과 주변 매크로 블록들의 평균으로 예측된 움직임 벡터(1st predictor)를 구한다.

2단계: 예측 움직임 벡터가 속하는 탐색 구간을 구한 후, 해당 후보점(2nd predictor)을 결정한다.

3단계: 움직임 벡터 예측점과 구간의 후보점 중에서 최소 SAD인 최적합점을 구한다.

4단계: 최적합점으로 탐색 원점을 이동하여 이동된 탐색 원점을 중심으로 3×3 화소를 가지는 정사각형 블록 탐색 패턴 내에 존재하는 9개의 탐색점에 대하여 최소 정합 오차를 계산한다. 이때 중심점이 최소 정합 오차면 5단계로 가고, 아니면 최소 정합 오차를 가지는 점을 중심점으로 하여 5단계로 간다.

5단계: 이전 단계에서 발견된 최소 정합 오차를 가진 점을 정사각형 블록의 중심점으로 지정하여 새로 계산된 최소 정합 오차가 중심점에 위치하면 6단계로 가고, 아니면 5단계를 반복한다.

a) 만약 이전의 탐색 과정에서 최소 정합 오차를 갖는 점이 수평 또는 수직 방향이라면 새로운 3개의 탐색 점을 추가하여 최소 정합 오차를 갖는 점을 중심점으로 하여 움직임 벡터를 추정하게 한다.

b) 만약 이전의 탐색 과정에서 최소 정합 오차를 갖는 점이 대각 방향이라면 새로운 5개의 탐색점을 추가하여 최소 정합 오차를 갖는 점을 중심점으로 하여 움직임 벡터를 추정하게 한다.

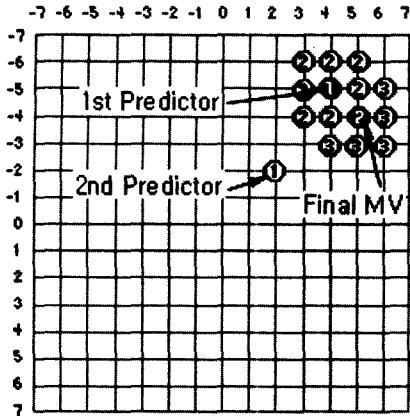
단, 탐색 영역($w=±7$)을 벗어나는 모든 점들은 무시한다. 매 단계마다 최소 정합 오차는 재정의되고, 최소 정합 오차를 갖는 점이 이전 탐색 단계에서 최소 정합 오차 점과 같다면 6단계로 이동하고 그렇지 않으면 5단계를 반복한다.

6단계: 탐색을 중단하고 이전 단계에서 구한 최소 정합 오차를 가진 점을 최종적으로 움직임 벡터로 결정한다.

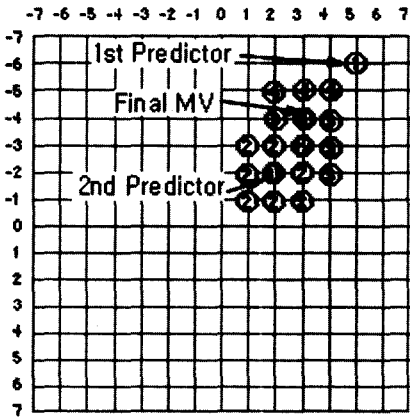
그림 7의 (a)와 (b)는 제안 알고리즘의 탐색 수행의 예로서 2개의 후보 벡터중에서 최소 SAD인 최적합점으로 탐색 원점을 이동하여 블록 정합을 수행하는 과정을 각각 보인 것이다. 제안된 방식은 탐색 영역이 벗어나는 모든 점을 무시하고 중간에 탐색을 멈추는 방법을 제시하였는데 중간에 멈출 수 있는 조건은 현재 탐색 단계에서 최소 정합 오차를 갖는 점의 위치가 이전 탐색 단계의 최소 정합 오차를 갖는 점의 위치와 같다면 중간에 탐색을 멈추고 최종적인 움직임 벡터로 결정한다.

4. 실험 결과

제안된 기법의 성능을 평가하기 위하여 6개의 실험 영상 News, Miss America, Foreman, Stefan, Flower Garden, Mother and Daughter에 대해 각각 80프레임씩을 대상으로 실험하였고, 비교 탐색 기법으로는 FS,



(a) 첫 번째 후보가 최소 SAD일 때



(b) 두 번째 후보가 최소 SAD일 때

그림 7 제안 알고리즘의 탐색 수행의 예

DS, HS, CDS, BBGDS와 예측 탐색 알고리즘인 TPA (Temporal Prediction search Algorithm), SPA (Spatial Prediction search Algorithm), 그리고 제안한 탐색 기법을 사용하였다. 예측 탐색 알고리즘인 TPA

및 SPA는 정확한 비교를 위해 제안한 기법과 같은 3×3 정사각형 블록 탐색 패턴을 적용하여 실험하였다. 그리고 움직임 추정에 사용된 매크로 블록의 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 (9)와 같은 평균 제곱 오차(MSE: Mean Squared Error), 식 (10)의 평균 절대값 오차(MAD: Mean Absolute Difference)와 정합 오차 측정 함수로는 식 (5)의 절대값 오차의 합(SAD: Sum of Absolute Difference)을 이용하였다. 또한 제안하는 기법의 성능 향상을 측정하기 위해 블록 당 탐색점의 개수를 기존 방법들과 비교하였다.

$$MSE(i, j) = \left(\frac{1}{N^2}\right) \sum_{k=1}^N \sum_{l=1}^N [I_t(k, l) - I_{t-1}(k+i, l+j)]^2 \quad (9)$$

$$MAD(i, j) = \left(\frac{1}{N^2}\right) \sum_{k=1}^N \sum_{l=1}^N |I_t(k, l) - I_{t-1}(k+i, l+j)| \quad (10)$$

그리고 화질의 평가를 위한 PSNR은 식 (11)과 같다.

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (11)$$

실험 영상에 대한 실험 결과는 표 1과 표 2, 그림 8, 그림 9에 각각 나타내었다.

표 1은 기존의 탐색 기법과 제안한 알고리즘의 각 블록에 대한 MAD와 평균 PSNR 값을 비교한 결과이며, 표 2는 각 블록에 대한 평균 탐색회수와 FS의 탐색 회수를 1로 기준하여 각 기법의 상대적인 속도 향상에 대한 비교 결과를 나타낸다. 표 1과 표 2의 CIF(Common Intermediate Format)는 352×288, SIF(Source Input Format)는 352×240, 그리고 QCIF(Quarter-CIF)는 176×144로 프레임의 크기를 나타낸다.

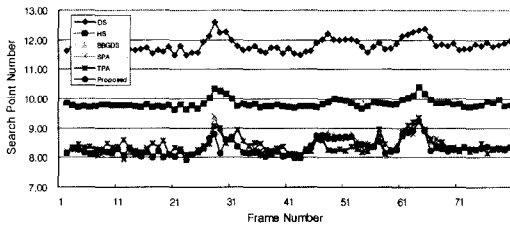
이러한 결과를 본다면 제안한 방식이 FS에 비해 평균 탐색점 수가 약 95% 정도 감소하였고, 탐색 속도면에서는 약 18~23배 정도의 성능 향상을 나타내었으며, 기존

표 1 각 실험 영상에 대한 성능 비교 평가 함수의 결과 비교 값

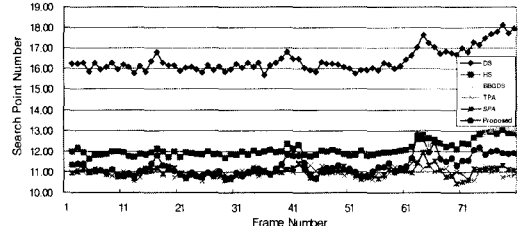
Sequence	CIF								SIF		QCIF		Average	
	News		MissAmerica		Foreman		Stefan		FlowerGarden	Mother&Daughter				
Method	MAD	PSNR	MAD	PSNR	MAD	PSNR	MAD	PSNR	MAD	PSNR	MAD	PSNR	MAD	PSNR
FS	1.042	38.225	1.930	39.095	2.817	33.456	6.715	25.693	8.520	23.994	1.168	41.160	3.699	33.603
DS	1.045	38.156	2.018	38.762	2.904	33.175	7.717	24.655	8.768	23.806	1.178	41.072	3.938	33.271
HS	1.062	38.000	2.115	38.264	3.196	32.923	7.929	24.501	9.509	23.320	1.188	40.972	4.167	32.997
CDS	1.046	38.129	1.960	39.006	2.960	32.933	8.076	24.409	8.822	23.729	1.181	41.041	4.008	33.208
BBGDS	1.045	38.147	1.959	39.025	2.850	33.282	8.202	24.193	8.776	23.741	1.177	41.079	4.002	33.245
TPA	1.102	37.449	1.981	38.945	2.892	33.075	7.464	24.526	9.063	23.399	1.179	41.056	3.947	33.075
SPA	1.046	38.146	1.961	39.022	2.857	33.057	7.629	24.732	8.801	23.505	1.189	40.586	3.914	33.175
Proposed	1.044	38.158	1.959	39.027	2.834	33.351	7.030	25.256	8.651	23.867	1.175	41.095	3.782	33.459

표 2 각 실험 영상에 대한 제안된 기법과 기존 기법의 탐색 속도 측면

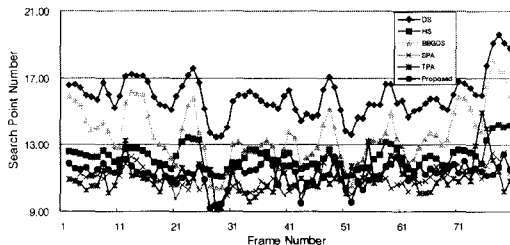
Sequence	CIF						SIF		QCIF		Average			
	News		MissAmerica		Foreman		Stefan		FlowerGarden		Mother&Daughter			
Method	탐색수	SpUp	탐색수	SpUp	탐색수	SpUp	탐색수	SpUp	탐색수	SpUp	탐색수	SpUp		
FS	204.28	1.000	204.28	1.000	204.28	1.000	204.28	1.000	202.05	1.000	184.56	1.000	200.62	1.000
DS	12.52	16.316	16.40	12.453	15.90	12.847	16.77	12.180	16.81	12.018	11.81	15.628	12.04	16.663
HS	10.51	19.446	12.09	16.897	12.31	16.598	12.98	15.737	13.15	15.365	9.83	18.767	11.81	16.987
CDS	9.06	22.546	11.53	17.723	13.45	15.186	14.44	14.143	14.96	13.504	8.70	21.213	12.02	16.691
BBGDS	8.84	23.102	11.39	17.942	13.67	14.944	14.52	14.073	14.12	14.307	8.36	22.072	11.82	16.973
TPA	8.87	23.023	10.97	18.618	11.09	18.417	9.93	20.575	9.43	21.432	8.42	21.931	9.79	20.666
SPA	8.85	23.075	10.98	18.613	10.93	18.682	10.29	19.850	10.28	19.655	8.36	22.067	9.95	20.163
Proposed	8.82	23.158	11.03	18.523	11.14	18.342	10.46	19.538	10.01	20.194	8.33	22.168	9.97	20.321



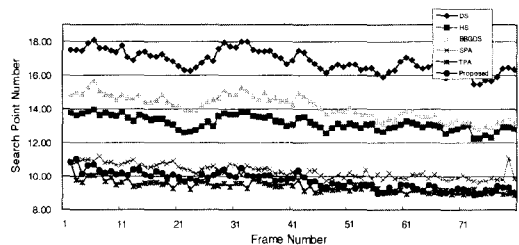
(a) "Mother and Daughter" 실험 영상



(b) "Miss America" 실험 영상



(c) "Foreman" 실험 영상



(d) "Flower Garden" 실험 영상

그림 8 각 실험 영상의 프레임별 블록당 평균 탐색점 수

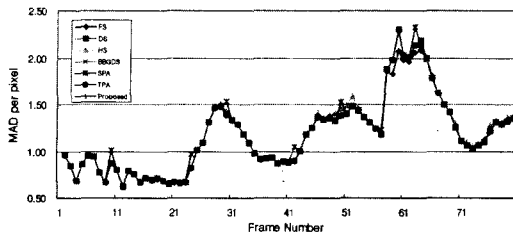
의 탐색 기법에 비해 탐색점 수가 감소하여 탐색 속도가 향상되었다. 또한 화질면에서 FS를 제외한 기존의 방식보다 우수함을 볼 수 있다. 즉, 평균 PSNR 값의 6개의 실험 영상에 대한 평균이 제안한 방식이 33.459[dB]로 나타나 33.603[dB]인 FS 다음으로 우수하였다.

그리고 실험에 사용한 영상별로 볼 때, 움직임이 아주 작은 News와 Mother and Daughter 영상은 FS와 비교하였을 때 탐색점 수를 95% 감소시키면서도 PSNR 값이 0.07[dB] 차이로 거의 화질의 저하가 없었으며, FS를 제외한 다른 기법에 비해 PSNR 값이 가장 우수하였다. 그보다 움직임이 조금 있는 Miss America 영상의 PSNR 값에서는 제안된 기법이 FS보다 0.07[dB] 정도 떨어지나 탐색점 수에서 약 18배의 많은 탐색점 수를 사용하여 움직임을 추정한 것에 비해 거의 근접한 값을 얻을 수 있었다. 그보다 움직임이 있는 Foreman 영상에서는 FS가 제안된 기법보다 0.105[dB] 정도 우수

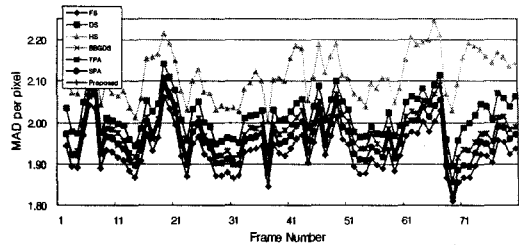
하나 탐색점 수를 18배 이상 사용되었고, 움직임이 큰 Stefan과 Flower Garden 영상에서는 FS가 각각 0.437[dB], 0.127[dB]정도가 좋으나 약 20배 정도의 많은 탐색점 수를 사용하였다.

이상의 실험 결과에 의하면, PSNR 측면에서 FS를 제외한 모든 기법에 비해 제안된 방식이 가장 우수한 것으로 나타났다. 속도면에서는 움직임이 적은 영상에서는 제안된 방식이 가장 우수하였으며, 움직임이 큰 경우는 SPA와 TPA 기법이 평균 탐색점 수에 있어서 각각 0.2%, 1.8% 감소로 거의 차이가 없음에도 불구하고 PSNR 값이 제안된 기법보다 평균 0.28~0.38[dB] 정도 나쁘게 나타난 것은 예측된 움직임 벡터로부터 탐색을 수행함으로써 블록 내에 공존하는 최소 정합 오차를 가지는 탐색점을 제외하기 때문이다.

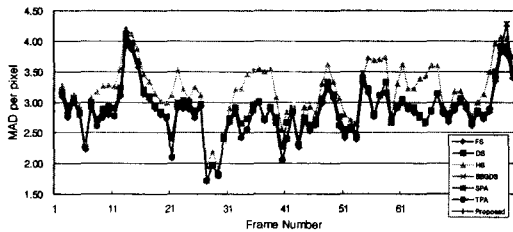
그림 8의 프레임별 블록당 평균 탐색수에 있어서 움직임이 적은 경우는 제안한 기법이 우수하게 나타났으



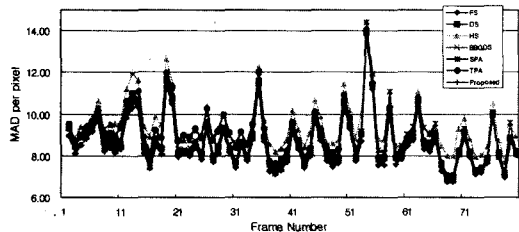
(a) "Mother & Daughter" 실험 영상



(b) "Miss America" 실험 영상



(c) "Foreman" 실험 영상



(d) "Flower Garden" 실험 영상

그림 9 각 실험 영상에 대한 프레임별 화소당 절대값 오차

며, 움직임이 큰 영상에서는 제안된 기법보다 예측 탐색 알고리즘들이 속도면에서는 우수하게 나타났으나, 그림 9의 프레임별 MAD 값에서는 제안된 기법보다 떨어졌으로써 화질면에서는 좋지 않게 나타났다. MAD 값에 있어서 제안된 기법이 기존의 예측 탐색 알고리즘인 TPA 와 SPA에 비해 이전 프레임의 변화 정도에 따라 MAD 값의 차이가 적음을 볼 수 있었으며, FS 다음으로 가장 우수하게 나타났다.

전체적으로 제안된 기법이 영상의 화질을 평가하기 위한 비교 평가 함수로 사용된 MAD에서는 0.13~0.39, 평균 PSNR 값에서는 0.19~0.46[dB] 개선되어 화질면에서도 FS에 근접하는 안정된 예측 정확도를 얻을 수 있었다. 그리고 같은 패턴을 사용한 BBGDS에 비해 제안된 방식이 PSNR 값에서는 0.214[dB], 속도면에서는 1.19배 향상되었다.

5. 결론

본 논문에서는 기존의 예측 탐색 알고리즘이 예측된 움직임 벡터로 탐색 원점을 이동시켜 탐색을 수행함으로써 블록 내에 공존하는 최소 정합 오차를 가지는 탐색점이 제외되어 화질 저하를 초래하는 문제점을 개선하기 위해 이전 움직임 정보에 의해 예측된 움직임 벡터와 탐색 영역의 중심에 가깝게 정의한 탐색 구간별 후보 벡터 중에서 가장 작은 SAD 값을 갖는 점을 탐색 원점으로 하여 탐색을 수행하는 알고리즘을 제안하였다. 실험의 결과를 보면 제안된 기법이 화질면에서는 전역 탐색에 근접한 우수한 성능을 보였으며, 기존의 예측 탐

색 기법과 거의 같은 탐색점 수를 사용하면서도 화질면에서는 나은 결과를 보였다. 이 경우 FS를 제외한 다른 기법과 비교하였을 때 움직임 예측면에서는 평균적으로 0.19~0.46[dB] 정도의 성능 향상을 보였다.

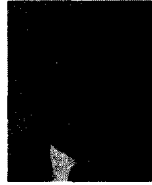
움직임 추정시 본 논문에서 제안한 예측 탐색 기법을 사용하여 움직임 추정을 한다면 보다 빠르게 움직임 벡터를 찾을 수 있을 것이며, 고속 블록 정합 방법에 적합한 탐색 패턴을 도출하고, 탐색 패턴을 적응적으로 변화시키면 다른 고속 블록 정합 방법들보다 탐색점 수를 감소시키고, 우수한 보상 결과를 얻을 수 있을 것으로 기대된다. 또한 일정한 시간 간격을 두고 연속적으로 추출되는 움직임 벡터의 크기를 일정하게 조절할 수 있는 시간적인 동기화에 대한 연구도 보장될 필요가 있다.

참고 문헌

- [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] Vincent S., S. Hwang, "Tracking Feature Points in Time-Varying Image using an Opportunistic Selection Approach," Pattern Recognition, Vol. 22, pp.247-256, 1989.
- [3] 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.
- [4] L. M. Po, W.C. Ma, "A Novel Four-Step Search

Algorithm for Fast Block-Matching Motion Estimation," IEEE Transactions on Circuits & System for Video Tech., Vol. 6, No. 3, pp.313-317, June 1996.

- [5] L. K. Kuo, E. Feig, "A Block-Based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding," IEEE Transactions on Circuits & System for Video Tech., Vol. 6, No. 4, pp.419-422, Aug. 1996.
- [6] 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.
- [7] Oscar T.-C. Chen, "Motion Estimation Using a One-Dimensional Gradient Descent Search," IEEE Transactions on Circuits & System for Video Tech., Vol. 10, No. 4, pp.608-616, June 2000.
- [8] Ce Zhu, Xiao Lin, Lap-Pui 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.
- [9] A. M. Tourapis, O. C. Au, M. L. Liou, "Highly efficient predictive zonal algorithms for fast block-matching motion estimation," IEEE Transactions on Circuits & System for Video Tech., Vol. 12, No. 10, pp.934-947, Oct., 2002.
- [10] F. Moschetti, M. Kunt, E. Debes, "A Statistical Block-Matching Motion Estimation," IEEE Transactions on Circuits & System for Video Tech., Vol. 13, No. 4, pp.417-431, Apr., 2003.



김 하 진

학사 서울대학교 문리과대학 수학과(1962). 석사 프랑스 Grenoble 1대학교 대학원 응용수학과(1978). 박사 프랑스 Saint-Etienne 대학교 대학원 응용수학과(1980). 1991년~1992년 한국정보과학회 회장. 1992년~현재 국제정보올림픽아드 추진위원장. 1992년~현재 JTC1/IEC SC24(그래픽스 표준화) 국내위원회 위원장. 1974년~현재 아주대학교 정보통신대학 교수. 관심분야는 컴퓨터그래픽스, 이미지프로세싱 및 응용수학



곽 성 근

학사 건국대학교 공과대학 전자공학과(1973). 석사 연세대학교 대학원 전자공학과(1980). 1995년~현재 아주대학교 대학원 컴퓨터공학과 박사과정. 1980년~현재 시립인천전문대학 컴퓨터정보과 교수

관심분야는 컴퓨터그래픽스, 이미지프로세싱 및 멀티미디어



위 영 철

학사 연세대학교 수학과(1980). 학사 SUNY at Albany Computer Science Department(1982). 석사 SUNY at Albany Computer Science Department(1984). 박사 SUNY at Albany Computer Science Department(1989). 1990

년~1995년 삼성종합기술원 수석연구원. 1995년~1998년 현대전자 부장. 1998년~현재 아주대학교 정보통신대학 조교수. 관심분야는 컴퓨터그래픽스, 이미지프로세싱 및 알고리즘