

# 이전 프레임의 시공간 모션 정보에 의한 예측 탐색 알고리즘

## A Prediction Search Algorithm by using Temporal and Spatial Motion Information from the Previous Frame

곽성근\*, 위영철\*\*, 김하진\*\*

Sung-Keun Kwak\*, Youngcheul Wee\*\*, Ha-Jine Kimn\*\*

\* 시립 인천전문대학 컴퓨터정보과 교수

\*\* 아주대학교 정보통신대학 컴퓨터공학과 교수

Incheon City College\*, College of Information Technology, AJOU University\*\*

### 요 약

비디오 시퀀스의 현재 블록의 모션 벡터와 이전 블록의 모션 벡터는 시간적 상관성을 갖고 있다. 따라서 이전 프레임 블록들로부터 많은 정보를 얻을 수 있다면 현재 블록의 모션 추정에 대한 성능을 높일 수 있고 또한 탐색 횟수를 줄임으로써 계산 복잡도를 크게 줄일 수 있다. 본 논문에서는 이전 프레임 블록과 주위 블록들의 모션 벡터로부터 예측된 모션 정보를 구하여, 이를 탐색 원점으로 사용하지 않고, 탐색 구간에 따라 적응적으로 해당 초기점으로 탐색 원점을 이동시켜 고속 탐색 패턴을 이용하여 블록 정합을 수행하는 블록 정합 모션 추정 방식을 제안한다.

실험 결과 제안된 방식은 기존의 예측 탐색 방식들에 비해 PSNR 값에 있어서 평균적으로 0.33~0.37[dB] 개선되고 영상에 따라 최고 1.05[dB] 정도 우수한 결과를 나타내었다. 또한 탐색 횟수에서는 기존의 탐색 알고리즘보다 29~97%를 줄일 수 있었고, 정확한 모션 벡터를 찾는 비교에 있어서도 월등히 우수한 결과를 나타내었다. 제안된 방식은 정량적인 결과뿐만 아니라 부호화후 복호화된 영상의 화질에 있어서도 다른 고속 탐색 알고리즘보다 월등히 우수한 화질을 제공한다.

키워드: 초기 탐색점, 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 the previous block. If we can obtain useful and enough information from the motion vector of the same coordinate block of the previous frame, the total number of search points used to find the motion vector of the current block may be reduced significantly. In this paper, we propose the block-matching motion estimation using an adaptive initial search point by the predicted motion information from the same block of the previous frame. And the first search point of the proposed algorithm is moved an initial point on the location of being possibility and the searching process after moving the first search point is processed according to the fast search pattern.

Simulation results show that PSNR(Peak-to-Signal Noise Ratio) values are improved up to the 1.05dB as depend on the image sequences and improved about 0.33~0.37dB on an average. Search times are reduced about 29~97% than the other fast search algorithms. Simulation results also show that the performance of the proposed scheme gives better subjective picture quality than the other fast search algorithms and is closer to that of the FS(Full Search) algorithm.

Key word : Initial Search Point, Block-Matching Algorithm

## 1. 서 론

인터넷과 이동 통신 등의 급격한 발전에 따라 컴퓨터 그래픽스 분야에서는 멀티미디어와 애니메이션과 같은 동영상 전송 수요가 확대되고 있으며, 다양한 서비스를 요구하는 사용자들의 요구를 만족시키기 위해 처리 시간의 단축이나 메모리의 효율적인 사용을 위한 동영상 분석 기법에

대한 관심이 고조되고 있다. 영상 부호화의 목적은 적은 양의 정보로 원영상을 충실히 표현하고 재생하는데 있다. 특히 만화의 형식에서 움직임을 첨가한 애니메이션은 형태를 변형하거나 물리적으로 움직이지 않는 대상을 움직여 공간을 재구성하므로 프레임간의 연관성이 매우 높다는 특징을 갖고 있다. 따라서 프레임 내의 정보와 프레임간의 정보에서 중복성(redundancy)을 줄이기 위해 모션 추정

(ME: Motion Estimation)을 이용함으로써 압축 효율을 높일 수 있다. 이러한 모션 추정에는 알고리즘의 단순성과 고속 처리를 위한 하드웨어적인 접근과 용이성으로 인해 MPEG 계열이나 H.261, H.263 등의 표준안으로 채택되어 사용되고 있으며, 컴퓨터 그래픽스 분야에서는 가상 회의를 위한 얼굴 표정(facial expressions) 시뮬레이션[1]과 계절풍 시뮬레이션(wind simulation)[2] 등에서도 이용되고 있다.

일반적으로 모션 추정 기법은 영상의 각 단위로 블록 또는 화소 단위로 적용되며, 이중 계산 복잡도 및 하드웨어 구현에 있어서 용이한 블록 단위의 모션 추정이 널리 사용되고 있으며, 블록 단위의 모션 추정은 동일한 블록 내의 화소들은 동일한 모션을 갖는다는 것과 블록들은 수평, 수직으로만 움직인다는 두 가지 전제 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 모션 벡터(motion vector)로 추정한다.

이와 같이 블록 단위로 모션을 추정하는 대표적인 알고리즘이 전역 탐색 블록 정합 알고리즘(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 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 오류가 가장 적은 모션 벡터를 찾을 수 있지만 많은 계산량이 필요한 단점이 있다.

이러한 전역 탐색법의 단점을 극복하기 위해 속도가 개선된 3단계 탐색(TSS: Three Step Search)[3], 새로운 3단계 탐색(NTSS: New Three Step Search)[4], 4단계 탐색(FSS: Four Step Search)[5], 다이아몬드 탐색(DS: Diamond Search)[6], 육각 패턴 탐색(HEXBS: Hexagon-based Search)[7] 등의 다양한 고속 블록 정합 알고리즘(FBMA: Fast Block Matching Algorithm)이 개발되었다. 이들 속도 개선 알고리즘은 주로 탐색 영역 내에서 탐색할 위치의 포인터 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)을 가지는 위치를 중심으로 다음 단계의 모션 탐색이 수행된다. 따라서 고속 블록 정합 모션 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 모션 벡터를 찾기 때문에 국부적인 탐색을 하게 되거나 해당 블록에 대한 모션 정보를 가지고 있지 않기 때문에 항상 탐색 영역의 원점에서부터 모션을 추정하는 과정에서 일부 탐색점들을 블록 정합 대상에서 제외시키거나 부정확한 방향으로 모션을 추정함으로써 복원된 영상의 화질을 저하시키는 단점이 있다.

그래서 기존의 고속 블록 정합 알고리즘의 단점을 개선하기 위해 이전 프레임의 같은 위치의 블록 또는 인접한 이전 블록들의 모션 벡터들의 정보를 이용하는 예측 탐색 알고리즘(PSA: Prediction Search Algorithm)이 제안되었다[8~10]. 이 방식은 이전 블록들의 모션 벡터 정보를 이용함으로써 보다 정확한 모션 벡터를 찾으려는 것이다. 그러나 이 방식은 참조한 블록의 모션 벡터들의 상관성이 떨어질 경우 압축 성능이 현저히 떨어지는 단점이 있다.

일반적으로 모션이 있는 비디오 영상의 경우 짧은 시간에 영상의 모션이 있더라도 그 시간에는 많은 모션이 있을 수 없으므로 인해서 모션 벡터는 중심 지향적 분포 특성을 갖게 된다. 또한 현재 매크로 블록의 모션 벡터는 이전 프레임의 같은 위치에 있는 블록들과 상관관계가 높으며, 비슷한 운동을 한다. 시간에 따라 변화하는 동영상은 시간 상관 계수가 공간 상관 계수보다 훨씬 크기 때문에 공간 상관성을 이용하는 프레임내의 부호화보다 프레임간의 높은 시간 상관성을 이용하여 프레임간의 시간적 여분을 감소시켜 압축 효율을 증가시키는 일이 효율적이다.

따라서 본 논문에서는 이전 프레임 블록의 인접 블록들의 모션 정보를 이용하여 현재 영상의 모션 벡터를 예측하여 적용적으로 해당 구간의 초기 탐색점에서부터 탐색을 수행함으로써 기존의 블록 정합 알고리즘들이 항상 탐색 영역의 원점에서 모션을 추정하는 비효율성을 개선하여 효과적으로 탐색점 수를 줄이고, 또한 기존의 탐색 패턴을 이용한 알고리즘의 국부적 최소(local minimum)로 인한 시각적인 화질 저하의 문제점을 개선하는 탐색 기법을 제안한다.

## 2. 기존의 예측 탐색 기법

최근에는 기존의 고속 탐색 알고리즘의 단점을 개선하기 위해 이전에 탐색된 모션 벡터들의 정보를 이용하는 예측 탐색 알고리즘이 제안되었는데, 이러한 예측 알고리즘은 현재의 매크로 블록의 모션 벡터가 주위 블록 또는 이전 프레임의 같은 위치에 있는 블록들과 상관관계가 높으며 비슷한 운동을 한다는데 착안하여 시간적으로 연속하거나 인접한 이전 블록들의 모션 벡터 정보를 이용함으로써 정

확한 모션 벡터를 찾는다. 그리고 이렇게 정의된 모션 벡터를 결정하여 블록 정합을 위한 탐색 패턴으로 TSS 또는 DS 방법을 적절히 적용하고 있다.

인접한 이전 블록들의 모션 벡터 정보를 이용하는 예측 탐색 알고리즘은 모션 벡터가 일반적으로 공간적으로 인접한 블록들은 비슷한 속도로 거의 같은 방향으로 움직인다는 점을 고려하여 [그림 1]과 같이 수평(left), 수직(above) 그리고 대각(aboveRight) 방향에 있는  $MB_0$ ,  $MB_1$ ,  $MB_2$ 들을 참조하여 예측 모션 벡터를 구한다[8~10]. 이와 같은 주변 모션 벡터의 중간 값을 사용하여 그 차를 인코딩하는 예측 알고리즘의 현재 블록의 예측 모션 벡터  $MV_p$ 는 식 (1)과 같이 이들 인접 블록 각각의 모션 벡터  $MV_{left}$ ,  $MV_{above}$ ,  $MV_{aboveRight}$ 들의 중간 값으로 한다.

$$MV_p = \text{median}(MV_{left}, MV_{above}, MV_{aboveRight}) \quad (1)$$

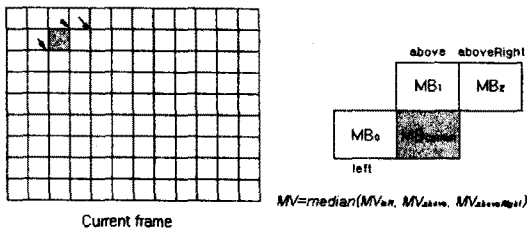


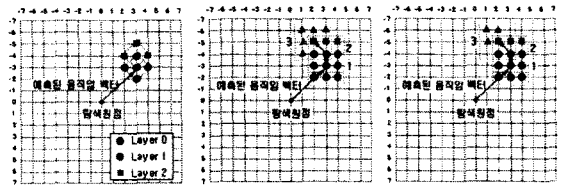
그림 1 공간적 상관성을 가지는 인접 매크로 블록

[그림 1]에서 현재 블록의 인접 블록(adjacent block)인  $MB_0$ ,  $MB_1$ ,  $MB_2$ 에서 그 중간 값을 취하여 예측 모션 벡터  $MV_p$ 를 결정할 수 있다. 인접 블록들의 모션 벡터의 중간 값이  $V_i = [\Delta c_i, \Delta r_i]$ 라고 할 때, 현재 블록  $MB_c$ 의 예측 모션 벡터는  $MV_p = [\Delta c_p, \Delta r_p]$ 가 된다. 따라서 현재 블록의 좌표가  $(m, n)$ 이면 탐색의 초기 중앙점의 좌표는  $(m + \Delta c_p, n + \Delta r_p)$ 이다. 즉, 탐색영역에서 탐색 원점(search origin)의 위치를 예측된 모션 벡터(predicted motion vector)만큼 이동시켜 새로운 탐색 원점을 설정한 후 블록 정합을 수행한다.

NNS(Nearest-Neighbors Search)[10] 탐색 과정은 [그림 2]의 (a)와 같이 순서적으로 후보 모션 벡터들의 집합으로 표현되는 각각의 탐색 다이아몬드 패턴 레이어로 구성된다. 이 알고리즘은 다른 중심들과 기껏해야 4개의 검사되지 않은 후보 모션 벡터들을 포함하는 연속적인 레이어(layer)들로 구성된 같은 크기의 레이어를 사용한다. 이때 레이어 1의 최소 SAD 점이 레이어 1+1의 중심이 된다. 즉, 연속되는 레이어들은 같은 다이아몬드 패턴을 적용하여 현재 탐색 중심과 그 중심과 인접하는 4개의 점을 갖

는다. NNS 알고리즘은 다음과 같이 요약된다.

먼저 1단계에서는 예측된 모션 벡터  $MV_p$ 를 얻어 현재 블록  $MB_c$ 에 적용하여 탐색 원점에서 예측된 모션 벡터 즉,  $(\Delta c_p, \Delta r_p)$ 만큼 이동시킨다. 그리고 2단계에서는 레이어 0의 위치점을 중심으로 이웃한 4개의 후보점을 추가하여 다이아몬드 패턴 상의 5개의 후보점(레이어 1)에서 최소 정합 오차를 계산하여 중심점이 최소 정합 오차면 탐색을 중단하고 4단계로 가고, 그렇지 않으면 3단계를 수행한다. 3단계에서는 이전 단계의 최소 정합 오차를 중심으로 새로운 레이어의 다이아몬드 패턴으로 탐색점을 추가하여 만약 중심점이 최소 정합 오차 점이면 4단계로 가고, 그렇지 않으면 3단계를 반복한다. 마지막으로 4단계에서는 탐색을 멈추고 이 값을 최종적으로 모션 벡터로 결정한다.



(a) NNS 탐색 경로 (b) PSA 탐색 경로 (c) GPS 탐색 경로

그림 2 기존 예측 탐색 알고리즘의 탐색 경로

PSA(Prediction Search Algorithm)[8]과 GPS(Gray Prediction Search)[9]의 탐색 경로는 각각 [그림 2]의 (b), (c)와 같으며, 이 두 기법들의 현재 블록은 좌표  $(m, n)$ 에 위치하는 것으로 전제로 하며, 단지 블록 정합시 탐색 패턴의 적용 방식이 다르다.

이들 알고리즘의 1단계는 예측된 모션 벡터  $MV_p$ 를 얻어 현재 블록  $MB_c$ 에 적용되며, 탐색 패턴의 초기 중심점의 좌표는  $(m + \Delta c_p, n + \Delta r_p)$ 로 이동된다. 그리고 2단계에서는 3x3의 정사각형 블록 패턴상의 9개의 후보점들 간의 최소 정합 오차 점을 찾아 만약  $S > \text{Count}$  또는 최소 정합 오차 점이 주어진 탐색 패턴의 중앙에 있다면 다음 단계로 가고, 그렇지 않으면 S는 1 증가한 후 다음의 두 개의 탐색 패턴에 따라 이 단계를 반복 수행한다. 만약 최소 정합 오차 점이 주어진 탐색 패턴의 경계 행/열의 중앙에 있다면, [그림 2] (b), (c)의 2와 같이 수직/수평 방향으로 인접한 탐색 후보점 3개가 추가된 다음 탐색 패턴의 중앙은 최소 정합 오차 점으로 이동하며, 최소 정합 오차 점이 주어진 탐색 패턴의 가장자리(코너)에 있다면, PSA는 [그림 2] (b)의 3과 같이 대각 방향으로 인접한 탐색 후보점 5개가 추가된 다음 탐색 패턴의 중앙은 최소 정합 오차 점으로 이동하고, GPS는 [그림 2] (c)의 3과 같이 대각 방향으로 인접한 탐색 후보점 3개가 추가된 다음 탐색 패턴의 중앙

은 최소 정합 오차 점으로 이동한다. 마지막으로 4단계에서는 탐색을 멈추고 이 값을 최종적으로 모션 벡터로 결정한다.

이러한 예측 탐색 기법은 공간적인 상관관계를 적용하여 속도뿐만 아니라 PSNR 및 시각적인 측면에서 좋은 결과를 보인다고 알려져 있다[11]. 그러나 예측된 모션 벡터를 탐색 원점으로 이용하여 빠르게 탐색을 수행함으로써 속도 면에서는 좋은 결과를 보일 수 있으나, 현재 프레임의 블록과 인접된 이전 블록만으로 모션 벡터를 예측함으로써 모션 벡터의 상관성이 떨어져 영상의 화질이 크게 저하될 수 있다.

따라서 본 논문에서는 현재 프레임의 인접 블록으로 예측 모션 값을 구하지 않고, 더 정확한 모션 벡터를 추정하기 위해 이전 프레임에서 구해진 모션 정보를 이용하여 상하좌우의 주위 블록까지 포함하여 모션을 예측함으로써 블록 내에 공존하는 유사한 정합 오차로 인해 잘못된 추정을 하는 문제점을 개선하고자 한다.

### 3. 시공간 모션 벡터와 초기점에 의한 예측 알고리즘

본 논문에서 제안하는 예측 탐색 알고리즘은 영상의 모션은 급격히 변화하기보다는 완만하게 변화한다는 연속성 제약 사항(motion continuity constraint)[12]에 근거하여 이전 프레임의 같은 위치에 있는 블록의 모션 벡터를 참조하여  $t-1$ 시점에서 추출한 모션 벡터를 이용하여  $t$ 시점에서의 모션 벡터의 위치를 예측한다. 즉, 현재 블록의 모션 벡터를 찾기 위하여 [그림 3]과 같이 이전 프레임( $t-1$ )의 같은 위치에 있는 블록  $MB_{p_{msl}}$ 와 좌우상하의 인접 블록  $MB_0, MB_1, MB_2, MB_3$ 들을 포함하여 5개의 평균 모션 벡터를 현재 프레임 블록의 예측 모션으로 이용한다.

$$MV_p = \frac{MV_{p_{msl}} + MV_{left} + MV_{right} + MV_{up} + MV_{down}}{5} \quad (2)$$

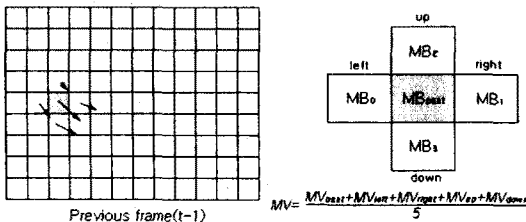
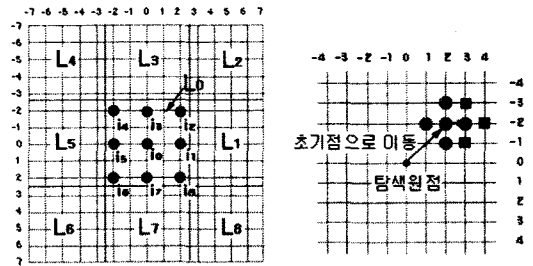


그림 3 이전 프레임의 인접 매크로 블록

제안한 알고리즘은 먼저 식 (2)에서 구해진 예측 모션 벡터 값이 속하는 [그림 4]의 (a)와 같이 9등분된 해당 탐

색 구간에 따라 초기점을 적응적으로 결정한다. 각 초기점의 위치는 이전 모션 벡터의 위치 오차와 모션 벡터가 탐색창의 중심점을 중심으로 반경 2화소 이내에 분포할 확률이 약 56.72%~98.70%[6]이라는 중심 지향적 분포 특성(center-biased property)을 고려하여 탐색창의 중심점의  $\pm 2$ 화소 이내에 설정한다. 초기점은 이전 시점의 블록에서 추출한 모션 벡터 정보에 의한 분할된 위치의 대표점으로 정의되며, 매크로 블록의 블록 정합을 위한 탐색 원점으로 이용된다. 이것은 각 구간의 적응적인 초기점을 해당 구간 내의 중심점으로 설정할 수 있으나, 실제 모션 벡터는 중심 지향적 분포 특성을 가진다는 점과 예측된 모션 벡터의 오차 범위 내의 모션 존재 가능 구간을 고려할 때, 구간  $L_0$ 와 각 구간의 경계에 있는 점을 초기점으로 하는 것이 효율적이므로 탐색 영역의 원점과 예측 모션 벡터 위치로 나가는 방향에 존재하는 점들 중에서  $L_0$ 내의 끝점을 각 구간의 초기점으로 한다.



(a) 탐색 구간별 초기점 (b) 탐색 과정의 예  
그림 4 적응적인 위치별 초기점과 탐색 과정

[그림 4]의 (a)는 각 탐색 구간별 초기점 위치를 나타낸다. 즉, 블록 정합을 위한 탐색 영역의 크기를  $(2p+1) \times (2p+1)$ 로 설정하고, 이전 프레임의 블록의 모션 벡터를  $MV_p$ 이고 이 모션 벡터가 속하는 탐색 구간이  $L_i$ 라 하면 현재 프레임 블록의 블록 정합의 첫 번째 탐색 후보점은 탐색 구간  $L_i$ 의 대표점인  $i_j$ 가 된다. 이때 각 모션 가능 구간의 초기점 위치를  $(i, j)$ 라 할 때, 모션 벡터를 추정하기 위한 탐색 원점의 위치는 식 (3)과 같다.

$$i, (i, j) \leftarrow L_i(MV_p), \quad -p \leq i \leq p, \quad -p \leq j \leq p \quad (3)$$

[그림 5]는 제안한 알고리즘의 순서를 간단히 나타낸 흐름도로 제안한 알고리즘에서는 참조 프레임의 모션 벡터 값에 따라 탐색 구간  $L_i$ 를 정하여 해당 초기 탐색점  $i_j$ 을 중심으로 [그림 4]의 (b)와 같이 추가된 4개를 포함한 5개의 탐색 후보점에 대하여 다음과 같은 알고리즘을 수행하여 모션 벡터를 추정한다.

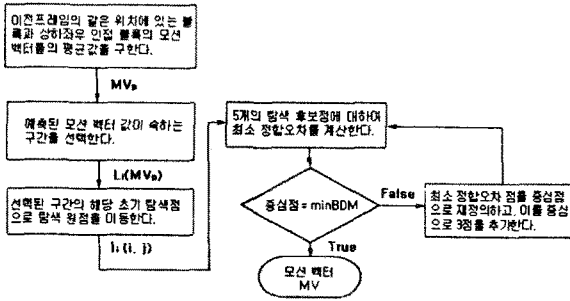


그림 5 제안된 모션 벡터 추정 방식의 흐름도

- 1단계: 이전 프레임의 인접 블록들의 모션 벡터 정보를 이용하여 예측된 모션 벡터 값을 구한다.
- 2단계: 이전 단계에서 구한 모션 벡터 값이 속한 탐색 구간을 찾아 해당 초기점으로 탐색 원점을 이동시킨다.
- 3단계: 이동된 탐색 원점을 중심으로 작은 다이아몬드 패턴 내에 존재하는 5개의 후보 탐색점에 대하여 최소 정합 오차를 계산한다. 이때 중심점이 최소 정합 오차면 5단계로 가고, 그렇지 않으면 새로 구한 최소 정합 오차 점을 중심점으로 하여 4단계로 간다.
- 4단계: 이전 단계에서 발견된 최소 정합 오차 점을 작은 다이아몬드 패턴의 중심점으로 지정하여 매 단계마다 최소 정합 오차는 재정의되고, 새로 계산된 최소 정합 오차가 중심점에 위치하면 5단계로 가고, 아니면 4단계를 반복한다. 단, 탐색 영역( $w=±7$ )을 벗어나는 모든 점들은 무시한다.
- 5단계: 이전 단계에서 구한 최소 정합 오차 점이 모션 벡터의 최종 해가 된다.

통계적으로 모션이 탐색 중심에 많이 분포하기 때문에 탐색 영역의 중심으로부터 가까운 거리에 있는 탐색점을 우선적으로 고려하기 쉽다. 상관도나 정합 기준 값이 비슷한 경우에 탐색 영역 중심으로부터 거리가 가까운 탐색점을 최종 모션 벡터로 결정하지만 우선순위는 정합 기준 값이 되어야 한다. 다시 말하면, 정합 기준 값이 비슷하다고 예상되는 탐색점 위치들 중에서 모션 벡터의 후보를 결정하는 것이 타당하다. 상관도의 관점으로 볼 때, 정사각형 블록을 사용하는 블록 정합의 경우에 탐색점들의 탐색 영역 중심으로부터 거리와 상관도의 관계는 일치하지 않으며, 실제의 경우에도 거리적으로 동일한 위치에 있을 때 상관도가 같다고 확신할 수 없다.

따라서 제안된 기법은 모션 존재 가능 위치를 고려한 초기점으로부터 적응적으로 탐색함으로써, 탐색 대상 영역이 줄어들게 되어 모션 추정에 대한 계산량이 상대적으로 많

이 줄어들게 된다. 즉, 초기점을 중심으로 상대적으로 축소된 탐색 영역 내에서 미소 변위 추정을 행하게 된다. 또한 이전 프레임에서 추정된 모션 정보를 이용함으로써 효율적으로 더 정확한 모션을 추정할 수 있다는 장점이 있다.

#### 4. 실험 결과

제안된 기법의 성능을 평가하기 위하여 CIF(Common Intermediate Format: 352×288 pixels)의 Coastguard, Susie, SIF(Source Input Format: 352×240 pixels)의 Flower Garden, Football, QCIF(Quarter-CIF: 176×144 pixels)의 Claire, Mother and Daughter, Table Tennis의 7개 영상에 대해 각각 80프레임씩을 대상으로 실험하였고, 비교 탐색 기법으로는 FS와 예측 탐색 알고리즘인 NNS, PSA, GPS, 그리고 제안한 이전 프레임의 시공간 모션 정보를 이용한 탐색 기법을 사용하였다. 그리고 모션 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 (4)의 평균 제곱 오차(MSE: Mean Squared Error), 식 (5)의 평균 절대 값 오차(MAD: Mean Absolute Difference)와 정합 오차 측정 함수로는 식 (6)의 절대 값 오차의 합(SAD: Sum of Absolute Difference)을 이용하였다. 또한 제안하는 기법의 성능 향상을 측정하기 위해 블록 당 탐색점의 개수를 기존 방법들과 비교하였다.

$$MSE(i, j) = \left( \frac{1}{N^2} \sum_{k=1}^N \sum_{l=1}^N |I_i(k, l) - I_{i-1}(k + i, l + j)|^2 \right) \quad (4)$$

$$MAD(i, j) = \left( \frac{1}{N^2} \sum_{k=1}^N \sum_{l=1}^N |I_i(k, l) - I_{i-1}(k + i, l + j)| \right) \quad (5)$$

$$SAD(i, j) = \sum_{k=1}^N \sum_{l=1}^N |I_i(k, l) - I_{i-1}(k + i, l + j)| \quad (6)$$

여기서 N은 영상의 가로와 세로의 각각의 크기이며,  $I_i(k, l)$ 은 원영상의 화면을 나타내고,  $I_{i-1}(k + i, l + j)$ 은 모션 추정 화면을 나타내며, 이들 정합 기준들은 최소 값을 가지는 위치를 모션 벡터로 결정한다. 그리고 화질의 평가를 위한 PSNR은 식 (7)과 같다.

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) \quad (7)$$

실험 영상에 대한 실험 결과는 <표 1>과 <표 2>에 각각 나타내었다. <표 1>에서는 각각의 실험 영상별로 기존의 탐색 기법과 제안하는 탐색 기법에 대한 평균 PSNR[dB]을 나타내었고, <표 2>에는 블록당 평균 탐색 수에 대한 실험 결과를 나타내었다.

표 1 각 실험 영상에 대한 평균 PSNR

Sequence	FS	NNS	PSA	GPS	제안
Coastguard	29.642	28.347	28.347	28.347	29.392
Susie	34.722	33.754	33.971	33.968	34.325
Flower Garden	23.994	23.523	23.505	23.490	23.750
Football	22.413	21.399	21.534	21.491	21.665
Claire	43.455	43.453	43.454	43.448	43.455
Mother&Daughter	41.160	40.666	40.586	40.586	41.052
Table Tennis	28.197	27.237	27.271	27.238	27.340
평균	31.940	31.197	31.238	31.224	31.568

표 2 모션 벡터 추정시 평균 탐색 횟수

Sequence	FS	NNS	PSA	GPS	제안
Coastguard	204.28	5.91	9.49	9.43	6.26
Susie	204.28	7.02	11.00	10.41	8.37
Flower Garden	202.05	6.40	10.28	10.04	7.30
Football	202.05	8.66	12.76	11.63	9.25
Claire	184.56	4.68	7.91	7.91	4.66
Mother&Daughter	184.56	5.05	8.36	8.28	5.04
Table Tennis	184.56	7.13	10.62	10.17	7.44
평균	195.19	6.41	10.06	9.70	6.90

<표 1>과 <표 2>에서와 같이 제안된 탐색 기법은 모션 추정 속도면이나 모션 추정 정확도면에서 기존의 예측 탐색 기법보다 더 나은 성능을 보였다. 전반적으로 제안된 탐색 기법은 FS에 비해 평균 탐색점 수가 약 97% 정도 감소하였고, 탐색 속도면에서는 약 25~40배 정도의 성능 향상을 나타내었으며, 기존의 탐색 기법에 비해 탐색점 수가 감소하여 탐색 속도가 향상되었으며 평균 PSNR 값은 FS 다음으로 우수하게 나타났다. 그리고 실험에 사용한 영상별로 볼 때, 모션이 작은 Claire와 Mother & Daughter 영상은 프레임간의 시간적 상관성이 많이 존재하는 제안된 기법이 FS에 비해 탐색점 수를 약 97% 감소시키면서도 PSNR 값이 거의 근접하였으며, 또한 다른 모든 기법에 비해 모션 추정에 필요한 탐색점 수가 가장 작게 나타나 속도면에서 높은 향상을 보였고 또한 PSNR 값도 가장 우수하였다.

그보다 모션이 조금 있는 Susie, Coastguard 및 Table Tennis 영상의 PSNR 값에서는 제안된 기법이 FS보다 떨어지나 탐색점 수에서 약 24~33배의 많은 탐색점 수를 사용하여 모션을 추정한 것에 비해 거의 근접한 값을 얻을 수 있었다. 모션이 큰 Flower Garden과 Football 영상에서는 FS가 제안된 기법보다 각각 0.23[dB], 0.52[dB] 정도 우수하나 역시 탐색점 수를 약 22~28배의 많은 탐색

점 수를 사용하여 모션 벡터를 추정하기 때문이다.

이상의 실험 결과에 의하면, PSNR 측면에서 FS를 제외한 모든 기법에 비해 제안된 방식이 가장 우수한 것으로 나타났다. 속도면에서는 모션이 적은 경우는 제안된 방식이 가장 우수하였으며, 모션이 큰 경우는 NNS 기법이 우수하게 나타났다. 그러나 모션이 있는 영상에서 제안된 기법보다 탐색점 수에서 약 10% 적게 나타난 NNS 기법은 탐색점 수가 적음에도 불구하고 PSNR 값이 제안된 기법보다 평균 0.37[dB] 정도 나쁘게 나타난 것은 인접 블록의 모션 벡터의 상관성이 떨어져 국부적 탐색에 의해 조기 수렴하는 최소화 문제에 빠지기 때문으로 예측의 정확성에서 시간적인 상관관계를 적용한 기법보다 효율적인 방식이 아님을 알 수 있었다. 즉, 시간에 따라 변화하는 비디오 영상은 시간 상관 계수가 공간 상관 계수보다 훨씬 크기 때문에 공간 상관성을 이용하는 프레임 내의 모션 추정보다 프레임간의 높은 시간 상관성을 이용하는 것이 효율적임이 실험 결과로 알 수 있었다.

전체적으로 제안한 기법이 모션 추정에 필요한 탐색점 수를 NNS를 제외한 기존 탐색 기법에 비해 29~97% 이상 줄임으로써 탐색 속도면에서 높은 성능 향상을 보였고, 영상의 화질을 평가하는 PSNR 값에서는 FS를 제외한 기법에 대해 0.33~0.37[dB] 개선되어 화질면에서도 안정된 예측 정확도를 얻을 수 있었다.

[그림 5]와 [그림 6]에서는 실험 영상의 프레임별 블록당 평균 탐색점 수와 프레임별 화소당 MAD 값을 각각 그래프로 나타내었다.

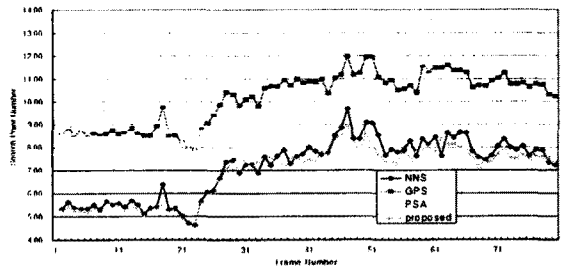


그림 6 "Table Tennis" 영상의 프레임별 블록당 평균 탐색점 수

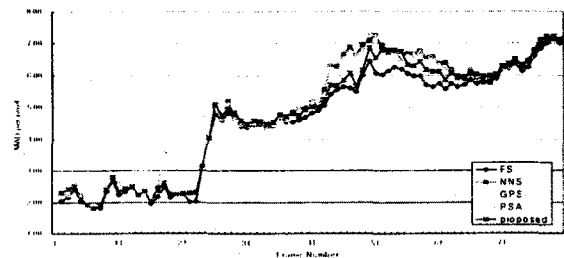


그림 7 "Table Tennis" 영상의 프레임별 화소당 절대값 오차

[그림 6]의 프레임별 블록당 평균 탐색수에 있어서 제안한 기법이 가장 작아 속도면에서 우수하게 나타났으며, [그림 7]의 프레임별 MAD 값에서는 기존의 예측 탐색 기법이 제안된 기법보다 떨어짐으로써 화질면에는 좋지 않게 나타났다.

## 5. 결론

제안한 알고리즘은 이전 프레임의 여러 후보 벡터에 의해 강인한 매크로 블록 단위의 모션 벡터를 생성하였으며, 여기에 제안한 매크로 블록 탐색 위치에 따른 첫 번째 탐색점을 이전 모션 정보를 이용함으로써 적은 탐색 수만으로도 전역 탐색에 근접한 성능을 나타내었다. 실험을 통하여 알 수 있듯이 제안된 기법은 모션 추정에 필요한 탐색 점수를 29~97% 줄임으로써 속도면에서 높은 향상을 보였고, 화질면에서도 우수한 성능을 보였다. 이 경우 모션 예측면에서도 평균적으로 0.33~0.37[dB] 정도의 성능 향상을 보였다.

모션 추정시 모션 벡터의 시공간적 상관성을 이용하여 현재 프레임의 매크로 블록 탐색 영역을 재정의하여 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. 그리고 제안된 기법을 보완함으로써 가상 회의를 위한 얼굴 표정 분석을 위한 카메라 프레임(camera frame)과 합성 프레임(synthetic frame)간의 모션 추정도 가능하리라 본다.

## 참고 문헌

[1] P. Eisert, B. Girod, "Analyzing Facial Expressions for Virtual Conferencing", IEEE Computer Graphics & Applications, Vol. 18, No. 5, pp.70-79, Sep./Oct., 1998.  
 [2] W. Lefer, B. Jobard, C. Leduc, "High-Quality Animation of 2D Steady Vector Fields", IEEE Transactions on Visualization & Computer Graphics, Vol. 10, No. 1, pp.2-14, Jan./Feb., 2004.  
 [3] 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.  
 [4] 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.  
 [5] 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.  
 [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] 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.  
 [8] L. Luo, C. Zou, X. Gao, Z. He, "A New Prediction Search Algorithm for Block Motion Estimation in Video Coding", IEEE Transactions on Consumer Electronics, Vol. 43, No. 1, pp.56-61, Feb., 1997.  
 [9] J. Jou, P. Chen, J. Sun, "The Gray Prediction Search Algorithm for Block Motion Estimation", IEEE Transactions on C & S for Video Tech., Vol. 9, No. 6, pp.843-848, Sep., 1999.  
 [10] 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.  
 [11] 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.  
 [12] S. Hwang, S. Vincent, "Tracking Feature Points in Time-Varying Image using an Opportunistic Selection Approach", Pattern Recognition, Vol. 22, pp.247-256, 1989.