DBpia

論文 96-21-5-3

시공간 상관성을 이용한 적응적 움직임 추정

正會員 김 동 욱*, 김 진 태**, 최 종 수*

Adaptive Motion Estimation Based on Spatio-Temporal Correlations

Dong-Wook Kim*, Jin-Tae Kim**, Jong-Soo Choi* *Regular Members*

※본 논문은 정보통신연구관리단의 대학기초연구지원사업의 연구비에 의해 연구되었음.

요 약

일반적으로 동영상은 배경 및 정지된 물체로부터 고속 움직임의 물체에 이르기까지, 움직임에 관한 여러 가지 요소들을 포함한다. 그러므로, 정확한 움직임 추정을 위해서는 여러가지 유형의 움직임을 고려해야 하며, 따라서 탐색 영역을 크게 설정해야 한다. 이 경우 매우 높은 계산 복잡도를 갖는 문제점을 안고 있다. 만일 움직임 추정을 하기 전에 블럭의 움직임을 예측할 수 있다면 이를 바탕으로 탐색 영역의 위치 설정 및 탐색 영역의 크기를 정의할 수 있다. 본 연구에서는 움직임의 높은 시공간적 상관성을 바탕으로 탐색 영역의 위치 및 크기 설정을 통해 적응적으로 움직임을 추정하는 새로운 기법을 제안한다. 제안된 알고리듬은 순방향 및 역방향 투영기법을 도입하여 초기 벡터를 예측하며, 이전에 얻어진 벡터들의 분포 및 예측 오차를 바탕으로 탐색영역의 크기를 적응적으로 결정한다. 모의 실험 결과, 제안된 알고리듬은 움직임 보상 화질 및 계산량의 감소에 있어서 높은 성능 향상을 보인다.

ABSTRACT

Generally, moving images contain the various components in motions, which range from a static object and background to a fast moving object. To extract the accurate motion parameters, we must consider the various motions. That requires a wide search region in motion estimation. The wide search, however, causes a high computational complexity. If we have a few knowledge about the motion direction and magnitude before motion estimation, we can determine the search location and search window size using the already-known information about the motion. In this paper, we present a local adaptive motion estimation approach that predicts a block motion based on spatio-temporal neighborhood blocks and adaptively defines the search location and search window

*중앙대학교 전자공학과
** 한서대학교 전산정보학과
論文番號:95343-1004
接受日字:1995年 10月 4日

www.dbpia.co.kr

size. This paper presents a technique for reducing computational complexity, while having high accuracy in motion estimation. The proposed algorithm is introduced the forward and backward projection techniques. The search window size for a block is adaptively determined by previous motion vectors and prediction errors. Simulations show significant improvements in the qualities of the motion compensated images and in the reduction of the computational complexity.

Ⅰ. 서 론

정보 전달의 수단으로서 영상 신호는 점점 더 그 중요성을 더해 가고 있다. 특히 통신 수단의 발달로 인하여 고속의 정보 전달 기능이 가능하게 되었으며, 최근에 와서는 여러 가지 정보, 특히 동영상 정보의 전송을 위한 표준 안으로서 MPEG-2가 소개되었다[1].

동영상은 그 특성상 시공간적으로 많은 상관성을 지니고 있으며, 고도의 압축율을 달성하기 위해 많은 기법들이 제안되었고, 특히, 그 중에서 블럭 정합법(block matching algorithm:BMA)은 계산량 및 하드웨어 구현의 용이성으로 인하여 널리 사용되고 있는 것이 현실이다[2][3][4].

동영상은 영상의 구성 요소로서, 움직임이 없는 배경으로부터 매우 빠르게 움직이는 물체에 이르기까지, 그리고 병진 이동으로부터 회전, 변형에 이르기까지 매우 복합한 움직임 요소들을 포함하고 있다. 따라서, 정확한 움직임 추정(motion estimation)을 위해서는 많은 고려 요소와 높은 계산량을 필요로 한다. 본 논문에서는 이러한 고려 요소 중 물체의 회전 및 변형 등에 관해서는 고려치 않고, 물체의 이동 방향 및 변이에 있어서 병진 이동을 한다는 가정으로부터 출발한다.

움직임의 추정 시 물체의 움직임의 방향 및 변이에 관한 선 지식이 없는 상황에서는 넓은 크기의 탐색 범위를 설정해야 한다. 따라서, 매우 높은 계산량이 필요하다. 일반적으로 계산량을 줄이기 위한 방안으로서 고속 알고리듬들이 제시되었다[5][6]. 그러나 고속 알고리듬의 경우 물체의 변이가 클 경우, 국부 최소치 문제에 봉착하기 쉬운 문제점을 지니고 있다.

동영상이 내포하고 있는 움직임의 특성, 즉 시공간적인 상관성을 바탕으로 이전에 구해진 움직임 정보를 다음 순서의 블럭 벡터를 구하기 위한 초기값으로 설정하는 기법들이 사용되었다[1][7][8][9]. 이전 프레임의 움직임 정보를 이용함으로써 적은 탐색 범위로 양호한 움직임 보상된 결과를 얻을 수 있다. 그러나, 이 경우 시간적 상관성을 고려함에 있어서 이전 프레임의 동일 좌표상의 움직임 벡터를 초기 벡터로 사용함으로써, 빠르게 움직이는 물체나 움직이는 물체, 그리고 움직임이 복잡한 영상 등의 경우 적용이 적절치 못하며, 특히 공간적 상관성과 시간적 상관성을 독립적으로 처리함으로써 문제점을 내포하고 있다[11].

만일 움직임 추정에 앞서 대상으로 하는 블럭 또는 영역에 대한 움직임 정보 즉, 움직임의 방향 및 크기를 예측할 수 있고 또 그 예측된 움직임 정보에 관한 신뢰성을 추정할 수 있다면 영상의 국부적 특성에 따라 탐색 위치 및 탐색 영역의 크기를 적응적으로 결정할 수 있다. 본 논문에서는 이전의 시공간적 움직임 정보를 바탕으로 초기 벡터를 설정하고 이를 바탕으로 움직임 특성에 맞게 적응적으로 움직임 정보를 추정하는 새로운 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 움직임 추정 기법에 대해 살펴보고, 3장에서 움직임의 시공간적 특성 및 투영 기법을 제시하며, 이를 바탕으로 4장에서 국부 적응적 움직임 추정 기법을 제안한다. 5장에서는 제안된 알고리듬의 성능 평가를 위한 모의 실험 결과를 제시하며, 마지막으로 6장에서 결론을 맺는다.

Ⅱ. 움직임 추정(motion estimation)

동 영상의 시간적 중복성 제거를 위해서 움직임 보상기법이 널리 사용된다. 이를 위한 움직임 추정기법은 영상의 각 영역 단위로 또는 블럭, 화소 단위로 적용이 된다. 이 중 계산적 복잡도 및 하드웨어의 구현에 있어서 용이한 블럭 단위의 움직임 추정 기법이 가장 널리 사용되고 있다. 블럭 정합법은 두장의 영상 사이에서 블럭들의 정합에 그 바탕을 두며 정합을

www.dbpia.co.kr

위한 불일치 기준을 최소화시키는 블럭이 선택된다.

블럭 정합법은 영상을 일정 크기의 블럭들로 나누고 블럭 내의 각 화소은 동일한 움직임을 가지며 각 블럭의 움직임 모델은 병진 운동만을 한다는 가정으로부터 출발한다. 움직임 추정은 이전 프레임으로부터 현재 프레임을 예측하고, 이를 위해 이동 변위, 즉 움직임 벡터를 생성한다. I_n를 현재 프레임, $I_n(x, y)$를 현재 프레임(t=n)의 좌표 (x, y)에서 영상의 밝기(image intensity)라 하고, I_{n-1}를 이전 프레임, 그리고 $I_{n-1}(x, y)$를 이전 프레임(t=$n-1$)의 좌표 (x, y)에서 영상의 밝기라고 하자. 현재 프레임의 $M\times N$ 크기의 기준 블럭(reference block) $O(i, j)$에 대한 움직임 벡터 $v_o(i, j)$는 아래 식 (1)에 의해 구해진다.

$$D(p, q)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\| I_n(x, y)-I_{n-1}(x+p, y+q)\|,$$

$$-W\le p, q\le W$$

$$v_o(i, j)=\arg\min_{-W\le p, q\le W}(D(p, q)) \qquad (1)$$

여기서, $D(p, q)$는 블럭 $O(i, j)$가 p, q변위를 가질 때의 불일치(disparity) 값이며, W는 탐색 영역(search window)의 크기이다. 식(1)의 $\|\cdot\|$는 불일치 값을 측정하기 위한 연산자이며, 예를 들어 차의 절대치 또는 차의 제곱이 사용될 수 있으며, 일반적으로 계산량을 고려하여 절대치 연산자가 널리 사용된다.

$v_o(i, j)=(d_i, d_j)$의 d_i, d_j를 얻기 위해서 만일 이전에 처리된 정보를 바탕으로 예측 벡터를 얻을 수 있다면 예측된 변이$(\hat{d}_i, \hat{d}_j)$만큼 탐색 영역의 중심을 이동시킴으로써 적은 계산량으로도 정확한 움직임을 추정할 수 있다. 즉,

$$v_o(i, j)=\arg\min_{-W\le p, q\le W}(D(\hat{d}_i+p, \hat{d}_j+q)) \qquad (2)$$

여기서, W′는 W′≪W의 관계를 가진다. 다음 장에서 움직임의 예측을 위하여 움직임의 시공간적 특성을 살펴보고 이를 바탕으로 예측 벡터를 얻고자 한다.

Ⅲ. 움직임의 시공간적 특성

3.1 공간적 상관성의 고려

영상 내의 움직이는 물체는 공간적으로 높은 상관도를 가지고 있다. 그림 1의 블럭 P는 움직임에 있어서, 인접한 블럭인 A, B, C,그리고 D와 높은 상관성을 지니고 있다.

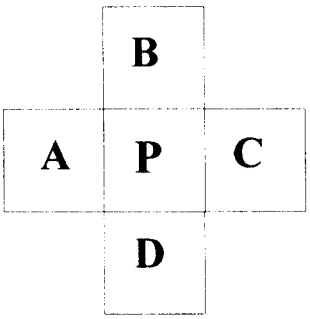

그림 1. 공간적 4-근방 블럭

따라서, 적은 계산량으로 블럭 P의 효율적인 움직임 추정을 위해 4-근방 블럭의 움직임 벡터가 탐색 영역의 중심 위치로 선택되기도 한다[9]. 만일 선택된 벡터가 부정확하거나 인접 블럭과의 움직임에 관한 상관성이 낮을 경우 인접 블럭의 벡터를 이용한 블럭 P의 움직임 추정의 결과는 매우 나쁘게 될 수 있기 때문에 초기치 선택을 위한 효과적인 방법이 요구된다. 그림 2는 football영상의 2번째 프레임에 대한 움직임 벡터들을 도식화한 것이다.

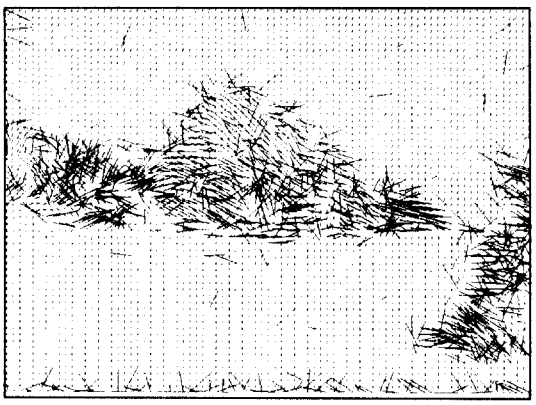

그림 2. 움직임 벡터(football 영상)

이 그림에서, 움직임 벡터는 배경 영역에서는 거의 영(zero)의 값을 가지며, 움직이는 물체 부분에서 공간적으로 인접 블럭의 벡터와 거의 비슷하나, 많은

www.dbpia.co.kr

부분에서 인접 블럭과 매우 다른 움직임 벡터를 나타내고 있음을 알 수 있다. 이러한 현상은 여러 가지 원인이 있을 수 있는데, 첫째, 블럭 정합법의 적용 시 최소 정합 오차를 만족하기만 하면 정합 블럭으로 선택되기 때문에 실제 움직임과 불일치하는 경우가 많이 발생된다. 둘째로, 서로 다르게 움직이는 물체 또는 물체와 배경 등이 동일한 블럭에 속해지는 경우이다. 그 외 드러난 배경(uncovered background) 영역 등의 경우에도 인접 블럭간 움직임의 상관도가 매우 떨어질 수 있다. 그림 3은 움직임 벡터의 크기와 그것에 대한 4-근방 블럭 움직임 벡터들의 편이량과의 관계를 나타낸 것이다(football 2번째 프레임, 블럭 크기 8×8, 탐색 범위 ±31). 여기서 움직임 벡터의 크기와 편이량은 식 (3)에 의해 구해진다.

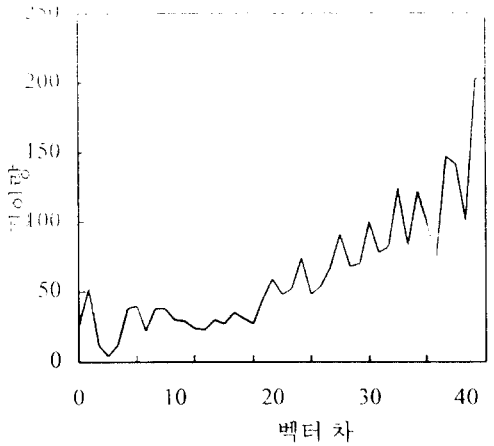

그림 3. 움직임 벡터의 크기와 편이량

$$\text{움직임 벡터의 크기} = \sqrt{v_x^2 + v_y^2}$$

$$\text{편이량} = |4v_x - \sum_{k=1}^{4} v_{xk}| + |4v_y - \sum_{k=1}^{4} v_{yk}| \qquad (3)$$

여기서 v_x, v_y는 블럭 벡터의 x성분, y성분을, 그리고 v_{xk}, v_{yk}는 그것의 4-근방 블럭 벡터들을 나타낸다. 그림 3에서 볼 수 있듯이 움직임이 적은 블럭의 경우 인접한 4근방 블럭과 편이량이 적은, 즉 유사한 값을 갖는 분포를 보이고 있다. 그러나 벡터의 크기가 커질수록 4근방 블럭간의 벡터의 유사도는 현저히 떨어짐을 알 수 있다. 따라서, 임의의 블럭에 대한 움직임 추정 시 탐색 영역의 위치 이동을 위해 초기 벡터로 인접 블럭의 벡터를 이용할 경우 해당되는 블럭의 움직임이 클수록 주변 블럭과 큰 차이를 가질 여지가 높아지기 때문에 적절치 못한 초기치 설정을 유발시킬 수 있다. 특히 배경 부분과 인접한 움직임 부분 및 드러난 배경(uncovered background) 영역 등에 있어서는 공간적 상관성이 현저히 떨어지게 된다. 따라서, 이러한 경우 공간적 상관성과 시간적 상관성과의 적절한 결합이 요구된다.

3.2 시간적 상관성의 고려

동영상에서 인접 프레임 간의 시간 간격은 매우 짧기 때문에 단위 프레임 시간당 움직임의 방향 및 크기의 변화량은 작은 범위 내로 제한된다고 볼 수 있다. 그런데, 그림 4에서 볼 수 있듯이 프레임 단위 시간 후 이동 물체가 겹쳐지기도 하며, 특히 실제 블럭 정합법의 적용 시 실제 움직임과 다른 값이 움직임 벡터로 선정되기도 한다. 따라서, 적절한 기법을 통해 임의의 블럭에 대해 얻어진 벡터 값이 실제의 움직임을 정확히 나타내고 있는지에 대한 고려가 필요하다.

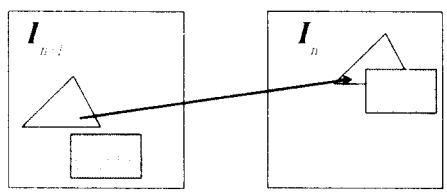

그림 4. 물체의 이동

시간적 상관성을 이용하기 위해 일반적으로 이전 프레임의 동일 좌표상에 위치한 블럭과 그것의 4-근방 블럭의 벡터들이 이용되며, 이는 다음 블럭의 초기 벡터로 이용된다. 그러나 물체의 경계 부분이나, 물체가 빠른 속도로 움직일 때 그 상관성이 매우 낮다. 따라서, 본 연구에서는 움직임 추정 시 인접한 프레임간의 움직임 요소의 상관성을 고려하기 위해 순방향 투영(forward projection) 및 역 방향 투영(backward projection), 2가지의 방법을 고찰하며, 이를 적용하고자 한다.

3.2.1 순방향 투영(forward projection)

www.dbpia.co.kr

일반적으로 블럭 기반으로 얻어진 움직임 벡터는 물체의 경계 부분에 걸쳐진 블럭의 경우 서로 다른 요소의 움직이는 물체를 포함하고 있다. 따라서, 이전 프레임 I_{n-1}에 걸쳐서 얻어진 블럭 벡터를 바탕으로 각 화소의 벡터 값을 부여하고자 할 때 공간적으로 인접한 블럭의 움직임 벡터를 이용하여 각 화소의 벡터를 결정할 수 있다[10]. 본 논문에서는 인접한 4근방 블럭들의 벡터만을 고려하고 이들 4-근방 블럭 벡터와 자신이 갖고 있는 블럭 벡터 중 최소 DFD(displaced frame difference)를 갖는 벡터를 블럭에 속한 화소의 화소 벡터로 지정한다. 이와 같은 방법으로 각 화소에 대한 화소 벡터를 결정하여 프레임 단위의 벡터 필드를 구성할 수 있는데 이를 I_{n-1}의 움직임 벡터 필드, V_{n-1}이라 칭하기로 한다. 만일 움직임 필드의 각 화소 벡터가 선형적이고 등속의 움직임을 가진다면 움직임 필드 V_{n-1}을 바탕으로 시간 Δt 후의 움직임 필드를 생성할 수 있으며, 생성되는 움직임 벡터 필드를 예측 움직임 필드(Predicted Motion Field, PMF), $\hat{V}_n$로 표기한다. 본 논문에서는 이와 같이 시간 변화에 따른 움직임 필드의 예측을 순방향 투영이라 정의한다. 그림 5는 V_{n-1}로부터 $\hat{V}_n$ 구성하기 위해 각 화소의 벡터를 순방향 투영하여 N, O, U로 분류하고 각각에 대한 최종적인 화소 벡터의 선정 과정을 나타낸 것이다. 이전 프레임의 벡터 필드 V_{n-1}의 각 화소 벡터를 순방향 투영시키고 이 때, 각 화소는 아래와 같이 분류된다.

(a) 1화소만 대응되는 점(N)

(b) 2화소 이상 대응되는 점(O)

(c) 하나도 대응되지 않는 점(U)

(a)의 경우가 가장 일반적인 1대1 대응 관계인 화소 벡터이다. (b)는 MSE를 바탕으로 최소치를 갖는 벡터가 화소 벡터로 선택되며, (c)의 경우는 인접한 화소의 벡터를 이용하여 보간된다. 다음으로, 구성된 $\hat{V}_n$을 I_n의 각 기준 블럭에 대응되는 블럭들로 구획을 나누고 나누어진 각 블럭에 대한 블럭 벡터, 즉 예측 블럭 벡터(predicted block vector : PBV)를 얻을 수 있다. 그런데, 임의의 블럭에 속하는 화소 벡터들은 움직이는 물체에 포함된 블럭의 경우 거의 동일한 벡터를 갖지만, 물체의 경계에 걸쳐진 블럭 등에서는 서로 다른 값을 가질 수 있다. 이러한 문제점을 해결하기 위해서 블럭에 속하는 $\hat{V}_n$의 화소 벡터들에 대해 메디언 필터링(median filtering)을 행하여 중간값을 얻고 이를 PBV로 선택한다. 얻어진 PBV를 이용하여 블럭 O(i, j)에 대한 움직임 추정 시 그 블럭과 동일한 좌표상에 위치한 $\hat{V}_n$상의 예측 블럭 벡터 v_o^f와 그것의 인접 블럭의 벡터들을 초기 벡터로 사용할 수 있다(그림 6).

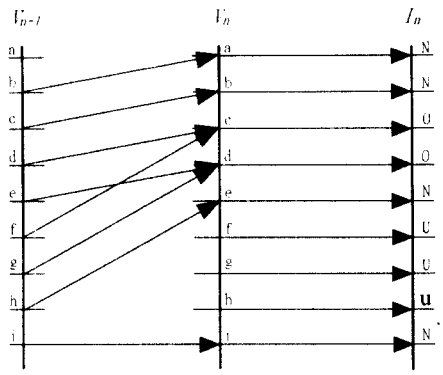

그림 5. 순방향 투영(화소 벡터의 결정)

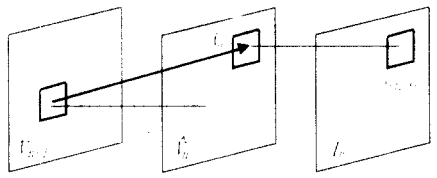

그림 6. 예측 블럭 벡터

3.2.2 역방향 투영(backward projection)

그림 7에서, 블럭 A의 움직임 추정이 이루어지고 블럭 O의 차례라고 하자. 이 때, 블럭 O의 이전 프레임 I_{n-1}에서의 위치를 예측하고자 한다. 만일 움직임 추정 결과, 블럭 A가 I_{n-1}의 p_a 위치로 부터 움직임 보상되고, 블럭 A의 움직임 벡터 v_a가 그 블럭에 대한 실제 움직임을 나타내고 있다고 하면, 블럭 A와 블럭 p_a는 시간적 차를 제외하면 동일 물체이다. 각각의 움직임 벡터는 만일 속도 변화가 크지 않다면 거의 유사한 값을 가질 것이다. 이와 같이 프레임 I_n의 블럭 벡터를 이용하여 I_{n-1}의 해당 블럭 벡터를 얻는 것을 역방향 투영이라 정의한다. 블럭 A에 대응되는 블럭 p_a의 블럭 벡터는 움직임 벡터 필드 V_{n-1}의 해당되는 화소 벡터들에 대해 메디언 필터링하여 중간값이 선

www.dbpia.co.kr

택된다. 한편, 현재 프레임 I_n에서 블럭 A와 블럭 O는 서로 인접하고 있기 때문에 큰 겹쳐짐 없이 병진운동을 통해 프레임 단위 시간 Δt동안 이동해 왔다면 이전 프레임 I_{n-1}에서도 역시 서로 인접해 있다. 따라서, 만일 블럭 A가 I_{n-1}의 p_a 위치로 부터 움직임 보상되었다면 블럭 O은 블럭 p_a, p_b, p_c, p_d, 그리고 p_o의 움직임과 매우 유사할 것이다. 이와 같이 블럭 O의 움직임을 예측하기 위해 이전 프레임 I_{n-1}로부터 선택되는 블럭들을 시간적 인접 블럭이라고 명명한다.

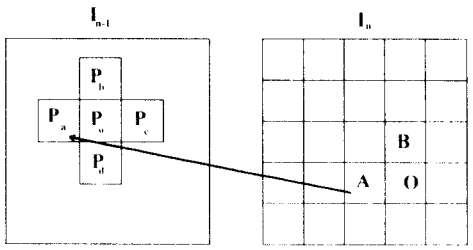

그림 7. 역방향 투영

Ⅳ. 국부 적응적 움직임 추정

본 장에서는 전 장에서 기술한 움직임 벡터의 시공간적 특성을 바탕으로 임의의 블럭에 대한 초기 벡터(initial vector)를 구하고, 이를 중심으로 탐색 영역의 크기(search window size)를 결정하기 위한 새로운 알고리듬을 제안한다. 현재 프레임의 각 블럭에 대한 초기 벡터는 이전에 구해진 공간적 4-근방의 인과적인 블럭의 움직임 벡터가 실제 움직임을 나타내는지에 대한 신뢰도를 바탕으로 시간적 인접 블럭과 결부되어 결정된다. 또한 결정된 초기 벡터는 탐색을 위한 중심 위치로 되는데, 이 때 고정 크기의 탐색 영역을 정하고 움직임 추정을 행할 수 있다. 이 경우 초기 벡터를 가지지 않는 경우보다 빠른 움직임을 적은 탐색 범위로 추정할 수 있다. 또한 탐색 영역의 크기를 적응적으로 결정할 수 있는데, 이 경우 초기 벡터와 후보 벡터들의 차 및 초기 벡터에 대한 DFD를 고려하여 탐색 영역의 크기가 정해진다.

4.1 후보 벡터 및 초기 벡터의 결정

현재 프레임의 임의의 블럭에 대한 초기 벡터는 공간적으로 또는 시간적으로 인접한 주변 블럭들이 가지고 있는 움직임 벡터를 바탕으로 구해진다. 이 때 초기 벡터 결정을 위한 후보 벡터들(candidate vectors)의 선택이 먼저 선행되어야 하는데, 이 후보 벡터들의 선택 과정은 2 가지의 단계를 가진다.

첫 번째 단계는 공간적으로 인접한 블럭 중 최소 오차를 갖는 블럭 벡터의 선택이다. 즉, 공간적으로 인접한 2개 블럭 중의 하나를 선택한다. 그림 7에서 공간적으로 4근방 블럭은 4개이나 인과적 블럭인 A, B만이 대상이 된다. 먼저 블럭 A, B가 갖고 있는 벡터 v_A, v_B를 이용하여 블럭 P에 대한 각각의 DFD (displaced frame difference)를 구하고 그 중 최소값을 갖는 블럭 벡터 v_S를 얻는다. 즉,

$$v_S = \arg\min\left(|DFD(v_N)|\right) \tag{4}$$

으로 나타낼 수 있다. 여기서, $v_N \in (v_A, v_B)$이며, $\arg\min(\cdot)$은 $(\cdot)$의 인자들 중 최소 $DFD(v_N)$를 주는 벡터를 되돌린다.

두 번째 단계는 시간적으로 인접한 이전 프레임 I_{n-1}의 시간적 인접 블럭의 블럭 벡터들 중 후보 벡터의 선택 단계이다. 전 장에서 언급되었듯이, 시간적 인접 블럭의 벡터는 4-근방만 고려한다면 순방향 투영 및 역방향 투영을 통해 각각 5개씩 모두 10개의 후보 벡터가 존재한다. 계산량을 고려하여 순방향 및 역방향 모드 중 하나를 택하고, 선택된 모드에 대해 각기 해당되는 후보 벡터들이 결정된다. 여기서, 순방향 및 역방향 투영을 이용한 예측의 정확도에 대해 고찰해 보자. 순방향 투영을 이용한 예측의 경우 Δt의 시간이 경과된 후의 움직임을 예측하기 때문에, 만일 움직임이 선형적이지 못할 때 잘못된 예측이 발생될 수 있다. 이에 반해 역방향 투영을 이용한 예측의 경우 동일한 시간 축에 위치한 인접 블럭의 벡터를 이용하기 때문에, 만일 이동 물체의 공간적인 분포 관계가 유지된다면 순방향 투영에 의한 결과보다 더욱 정확한 결과를 기대할 수 있다. 따라서, 역방향 투영을 이용한 예측을 중심으로 하고 선택적으로 순방향 투영의 결과를 이용하고자 한다. 블럭 벡터 v_S를 이용한 순방향 및 역 방향에 대한 모드 선택은 아래 식(5), (6)을 바탕으로 이루어진다. 만일 식(4)의 결과 $v_S = v_A$이고, 그림 7에서 v_A를 역방향 투영 후 블럭 p_a와 대응

www.dbpia.co.kr

된다고 할 때, 얻어지는 벡터를 v_{p_e}라 놓으면,

$$v_{p_e}(i', j') = v_A^b(i, j) = \arg.MED_{[s,t]\in p_e}(V_{n-1}(k+s-d_A(i,j)_x,\ l+t-d_A(i,j)_y)) \quad (5)$$

$$v_{diff}(i, j) = |v_A(i, j) - v_{p_e}(i', j')| \quad (6)$$

여기서 $MED_{[s,t]\in p_e}$는 블럭 움직임 벡터 에 의해 역방향 투영되어 생성된 블럭 내부의 화소 벡터 중 중간 값을 선택하는 연산자(median filter)이며, (i', j')는 투영 후 대응되는 블럭 p_a의 좌표를, (k, l)는 i행, j열의 블럭 좌표에 대한 벡터 필드 V_{n-1}의 좌표를 의미하며, $d_A(i, j)_x$, $d_A(i, j)_y$는 각각 $v_A(i, j)$의 x, y 성분 즉, 변이이다. 식 (6)의 연산 결과, 만일 v_{diff}가 적은 값을 갖는 임의의 문턱치 T보다 크지 않다면 이것은 대응되는 블럭의 벡터들이 움직임을 정확히 나타내고 있는 것을 의미한다. 이것은 블럭 정합법을 적용할 때 나타나는 실제 움직임과 다른 벡터 값을 얻게 되는 문제점을 배제하기 위한 것이다. 즉, 이것은 v_{diff}가 큰 경우는 투영 전후의 벡터 중 실제 움직임과 상이한 값을 갖고 있다는 가정으로부터 출발한다. 이 경우 역방향 투영을 통해 후보 벡터가 결정되며 3장에서 제시되었던 바와 같이 역방향 투영 후 얻어지는 블럭 p_a, p_b, p_c, p_d, p_o에 해당되는 벡터가 선택되며(그림 7) 얻어지는 후보 벡터들은 아래 식(7)과 같다.

$$\begin{aligned}
v_{p_a} &= \arg\ MED_{[s,t]\in p_a}(V_{n-1}(k+s+B-d_A(i,j)_x,\ l+t-B-d_A(i,j)_y))\\
v_{p_c} &= \arg\ MED_{[s,t]\in p_c}(V_{n-1}(k+s+2B-d_A(i,j)_x,\ l+t-d_A(i,j)_y))\\
v_{p_d} &= \arg\ MED_{[s,t]\in p_d}(V_{n-1}(k+s+B-d_A(i,j)_x,\ l+t+B-d_A(i,j)_y))\\
v_{p_e} &= \arg\ MED_{[s,t]\in p_e}(V_{n-1}(k+s+B-d_A(i,j)_x,\ l+t-d_A(i,j)_y))
\end{aligned} \quad (7)$$

여기서 B는 블럭 크기이다. 블럭 O에 대한 후보 벡터는 시간적 인접 블럭인 식(7)의 4개와 식(5)의 1개, 그리고 식(4)의 v_s, 즉 v_A를 합하여 모두 6개가 얻어지며, 이 중 최소의 DFD를 갖는 벡터가 초기 벡터로 결정된다. 즉, 초기 벡터는

$$\hat{v}_o(i, j) = \arg\min\ (|DFD\ (v_o^{candidate}(i, j))|) \quad (8)$$

이고, 여기서 $v_o^{candidate} \in (v_{p_a}, v_{p_b}, v_{p_c}, v_{p_d}, v_{p_A})$이다.

한편, 만일 식 (6)의 값 v_{diff}가 문턱치 T보다 크다면 그 이유로는 v_A 또는 대응되는 I_{n-1}의 움직임 벡터 v_{p_e}가 실제의 움직임과 다른 값을 가지고 있을 때, 또는 Δt동안 속도 변화가 클 때이다. 그런데 Δt는 매우 짧은 시간이기 때문에 속도의 변화는 크지 않으며, v_{diff}가 크다면 이것은 해당 요소들의 움직임 벡터가 실제의 움직임과 다른 경우로 간주된다. 따라서, 이 경우에는 시간적 인접 블럭의 후보 벡터 선정은 v_s에 의존할 수 없으며 후보 벡터들은 순방향 투영 후 얻어지는 예측 블럭 벡터를 v_o^f라 두면 v_o^f와 그것의 4-근방 블럭 벡터, 그리고 v_A 중에서 선택된다. 따라서, 최종적인 초기 벡터는

$$\hat{v}_o(i, j) = \arg\min\ (|DFD\ (v_o^{candidate}(i, j))|) \quad (9)$$

여기서, $v_o^{candidate} \in (v_A \cup v_o^f \cup v_o^N)$이고 v_o^N는 v_o^f의 4-근방 블럭, 즉 시간적 인접 블럭의 벡터들이다.

이상의 과정을 통해 얻어진 초기 벡터를 이용하여 탐색 영역의 중심 위치를 이동시킬 수 있다.

4.2 국부 적응적 탐색 크기

본 절에서는 전 절에서 얻어진 초기 벡터를 바탕으로 움직임 추정을 위한 탐색 크기를 국부 적응적으로 결정한다. 탐색 크기의 결정은 초기 벡터와 후보 벡터들 간의 차, 초기 벡터에 대한 DFD를 바탕으로 한다.

그림 3에서, 움직임이 클수록 인접한 블럭간의 움직임 벡터의 유사도가 떨어짐을 알 수 있다. 즉, 이것은 초기 벡터와 후보 벡터들 간의 차의 변화가 클수록 더 큰 탐색 크기가 필요함을 의미한다. 한편, 만일 물체의 변형이 적다면 최종적으로 결정된 초기 벡터에 대한 DFD는 정합 정도에 따라 그 값을 달리한다. 즉, DFD가 적다면 초기 벡터로 결정된 벡터가 움직임 추정 후 얻어지는 벡터와 매우 가까우며, 적은 탐색 크기만으로도 움직임 추정이 가능하다. 그러나 DFD가 클 경우 부정확한 초기 벡터 설정으로 인하여 더 넓은 범위의 탐색을 요한다. 따라서, 얻어진 초기 벡터를 중심으로 정확한 움직임 벡터를 얻기 위한 탐색 영역의 크기는 벡터 $\hat{v}_o$의 신뢰도에 의존하며, 벡터 $\hat{v}_o$

www.dbpia.co.kr

의 신뢰도는 위의 고찰로부터 DFD와, 인접 블럭 벡터들간의 유사도에 의존할 것으로 예상할 수 있다. 본 논문에서는 이를 위해서 사후 확율(a posteriori probability)을 구하고, 이를 바탕으로 탐색 크기를 결정하고자 한다. 인접 벡터간의 상호 의존성(interdependency)은 Gibbs/Markov 랜덤 필드에 의해 모델링될 수 있다[12][13]. 주어진 영상 I_{n-1}에서 I_n으로의 벡터 필드 V에 대한 사후 확율을 $p(V|I_{n-1}, I_n)$으로 놓을 수 있다. 이 때, $V=\hat{v}_o$일 때의 사후 확율 $p(V|I_{n-1}, I_n)$이 높을수록 벡터 $\hat{v}_o$로 결정될 확율이 높음을 의미하며, 반면에 사후 확율이 낮을수록 벡터 $\hat{v}_o$로 결정될 확율이 낮음을 의미한다. 즉, 사후 확율과 탐색 영역의 크기는 반비례 관계로 근사시킬 수 있다. 임의의 블럭에 대해 $\hat{v}_o$를 초기 벡터로 한 움직임 추정시 탐색 영역의 크기 $S(\hat{v}_o)$는 아래와 같은 근사화시킬 수 있다.

$$S(\hat{v}_o) \propto \frac{1}{p(\hat{v}_o|I_{n-1}, I_n)} \tag{10}$$

식(10)은 Bayes 법칙에 의해 아래 식으로 놓을 수 있다.

$$S(\hat{v}_o) \propto \frac{1}{p(I_{n-1}, I_n|\hat{v}_o)\cdot p(\hat{v}_o)} \tag{11}$$

식(11)에서 오른쪽 분모에서 첫째 요소 $P(I_{n-1}, I_n|\hat{v}_o)$는 벡터 필드 $\hat{v}_o$에 대한 영상 I_{n-1}, I_n의 유사도(likelihood)를 나타내며, 둘째 요소 $p(\hat{v}_o)$는 벡터 필드 모델에 의해 근사화된 변이 벡터들의 상호 의존성을 나타낸다. 먼저 $P(I_{n-1}, I_n|\hat{v}_o)$를 유도하는데 있어서 참고 문헌[12]의 가정과 그로부터 유도된 결과를 적용한다. 즉,

$$P(I_{n-1}, I_n|\hat{v}_o) \propto \frac{1}{\sqrt{2\pi\cdot Max(DFD(\hat{v}_o), N_C)}^M} \tag{12}$$

여기서 M은 화소의 갯수, 그리고 N_c는 잡음 전력을 나타낸다.

한편, 벡터 필드의 상호 의존성을 얻기 위해 벡터 필드 모델로서 Gibbs/Markov 랜덤 필드를 이용하며, Gibbs/Markov 랜덤 필드[13]는

$$p(\hat{v}_o) \propto \exp(-\sum_{k=1}^{M} C_K) \tag{13}$$

로 주어진다. 여기서, M은 클럭(clique)의 수를 나타내며 하나의 클럭은 두개의 블럭 벡터로 구성되는데, 하나는 $\hat{v}_o$를, 다른 하나는 주변 블록 벡터 중 하나이다. C_k는 k번째 클럭에 할당되는 cost이다. 이 때, 초기 벡터 $\hat{v}_o$와 주변 블럭 벡터와의 상호 의존성을 구하기 위해 현재 프레임의 인과적 주변 블럭과 이전 프레임의 시간적 인접 블럭 벡터를 이용한다. 현재 프레임의 경우 그림 1에서, 인과적 블럭인 A, B를 이용하며, 시간적 인접 블럭의 경우 초기 벡터를 구하기 위해 얻어진 후보 벡터가 적용된다. 따라서, 적용되는 클럭은 1차 Gibbs/Markov 랜덤 필드가 사용된다. 이것은

$$C_i = \frac{c}{l_s}\cdot\|\hat{v}_o - v_i\| \tag{14}$$

$$C_j = \frac{c}{l_s + l_T}\cdot\|\hat{v}_o - v_j\| \tag{15}$$

로 나타낼 수 있다. 여기서, c는 상수 이며, l_s는 공간적 거리를 나타내며 4근방일 경우 1, 8 근방일 경우 $\sqrt{2}$로 된다. l_T는 프레임간 시간 간격(time sampling factor)을 나타내며, 시간 간격이 클수록 값이 커진다. 본 논문에서는 30Hz 프레임율일 경우 1로 놓았다.

얻어지는 $S(\hat{v}_o)$는

$$S(\hat{v}_o) \propto \ln(Max(DFD(\hat{v}_o), N_C)) + \sum_{i=1}^{p}\frac{c}{2l_s}\cdot\|\hat{v}_o - v_i\| + \sum_{j=1}^{q}\frac{c}{(l_s+l_T)}\cdot\|\hat{v}_o - v_j\| \tag{16}$$

따라서, 탐색 영역의 크기 $S(\hat{v}_o)$는

$$S(\hat{v}_o) = C_1\ln(Max(DFD(\hat{v}_o), N_C)) + C_2\left(\sum_{i=1}^{p}\frac{c}{l_s}\cdot\|\hat{v}_o - v_i\| + \sum_{j=1}^{q}\frac{c}{(l_s+l_T)}\cdot\|\hat{v}_o - v_j\|\right) \tag{17}$$

4.3 움직임 벡터의 최대값을 고려한 초기 벡터의 재설정

전 절에서 얻어진 초기 벡터를 바탕으로 정의된 탐색 영역내의 정합 위치를 찾음으로서 움직임 벡터를 얻게된다. 이 때 탐색 범위가 클수록 실제 움직임과 다른 블럭을 정합 블럭으로 찾는 오정합의 발생 가능

www.dbpia.co.kr

성이 높아진다. 이것은 블럭 정합법의 문제점 즉, 블럭 정합 결과 얻어진 움직임 벡터가 물체의 실질적인 움직임을 언제나 표현하고 있다고는 볼 수 없기 때문이다. 이것은 특히 블럭의 크기가 적을수록 잘못된 벡터를 얻을 확율이 높으며, 드러난 배경이나 평활한 영역일수록 발생하기 쉽다. 따라서, 만일 잘못된 벡터를 초기 벡터로 얻을 경우 이로 인하여 그 결과는 계속 누적되므로 매우 잘못된 결과를 가져오는 블럭이 발생하는 경우가 있다. 본 논문에서는 이러한 문제점을 해결하기 위하여 얻어지는 움직임 벡터의 값을 제한된 값 이내에서 갖도록 하고자 한다.

만일 얻어진 초기 벡터가 $\hat{v}_o(x, y)=(\hat{d}_{ox}, \hat{d}_{oy})$일 때 최대 탐색 범위를 수평, 수직 방향으로 각각 $\pm W_x$, $\pm W_y$로 할 경우 얻어질 수 있는 움직임 벡터 $v(x, y)=(d_x, d_y)$의 범위는

$$-W_x+\hat{d}_{ox} \le d_x \le \hat{d}_{ox}+W_x \tag{18}$$
$$-W_y+\hat{d}_{oy} \le d_y \le \hat{d}_{oy}+W_y \tag{19}$$

를 갖는다. 만일 얻어지는 움직임 벡터 d_x, d_y의 최대값을 각각 M_x, M_y, 최소값을 각각 $-M_x$, $-M_y$로 제한하면 얻어지는 움직임 벡터 $v(x, y)$는

$$|d_x| \le M_x, \quad |d_y| \le M_y \tag{20}$$

를 만족해야 한다. 따라서, 초기 벡터의 값은 수평, 수직 성분에 대해 아래와 같이 각각 수정되어 적용된다.

$$d'_x=\begin{cases} M_x-W_x, & \text{if } \hat{d}_{ox}>0 \text{ and } \hat{d}_{ox}+W_x>M_x \\ W_x-M_x, & \text{if } \hat{d}_{ox}<0 \text{ and } \hat{d}_{ox}-W_x<-M_x \\ \hat{d}_{ox}, & otherwise \end{cases} \tag{21}$$

$$d'_y=\begin{cases} M_y-W_y, & \text{if } \hat{d}_{oy}>0 \text{ and } \hat{d}_{oy}+W_y>M_y \\ W_y-M_y, & \text{if } \hat{d}_{oy}<0 \text{ and } \hat{d}_{oy}-W_y<-M_y \\ \hat{d}_{oy}, & otherwise \end{cases} \tag{22}$$

이러한 고려에 있어서, 제안된 알고리듬은 적은 탐색 범위로 큰 움직임을 추정하고자 하는것이기 때문에 일반적으로 $M_x \gg W_x$, $M_y \gg W_y$의 관계를 갖는다. 따라서, 위에서 언급된 에러의 누적 현상은 거의 발생되지 않는다.

V. 모의 실험

제안된 알고리듬의 성능 평가를 위한 실험 영상으로는 720×480 크기의 football, flower 영상을 대상으로 하였으며, 블럭 크기는 8×8과 16×16, 두 가지에 대해 실험을 행하였다. 그 외의 여러가지 영상에 대해서는 움직임이 적기 때문에 실험에서 제외시켰다. 움직임 추정 시 이전 프레임은 복호화된 영상 대신 원 영상을 그대로 입력하여 사용하였는데, 이것은 움직임 추정에 대한 성능만을 정확히 비교하기 위해 행해진 것이다.

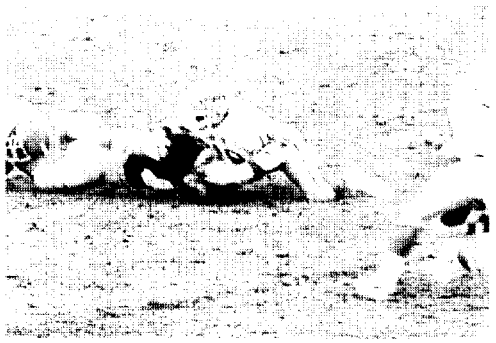

그림 8. football 원 영상(2번째 프레임)

실험에서 제안된 알고리듬의 적용을 위해 탐색 범위를 설정함에 있어서, 대상 영상별로 각 영상의 최대 움직임 거리를 고려하고, 탐색 범위는 고려된 최대 움직임 범위의 $\frac{1}{2}$ 을 초과하지 않도록 설정하였다. 즉, football영상의 경우 약 최대 움직임은 화소 단위로 ±31, flower영상 ±15로 놓고, 이에 따라 적용된 탐색 범위(W)는 각각 football의 경우 ±15, ±7, flower는 ±7로 설정하고 실험을 하였다. 이와 함께 움직임 추정 후 가질 수 있는 움직임 벡터의 최대, 최소값은 2가지 영상의 최대 움직임 값으로 설정된다.

제안된 알고리듬의 실험은 탐색 영역의 크기(W)를 고정시킨 제안 방법-A (method-A), 그리고 탐색 영역의 크기를 적응적으로 가변시키는 제안 방법-B (method-B), 두 가지에 대해 행하였다. 또 제안된 알고리듬의 성능 평가를 위해 2가지의 다른 움직임 추정 기법들 즉, Full search BMA(FS-BMA)와 기존의 방법[9](con-

www.dbpia.co.kr

ventional[9])과 비교했다. FS-BMA는 가장 일반적으로 사용되는 전역 탐색 기법을 의미하며, 각 블럭에 대한 움직임 추정은 공간적으로나 시간적으로 인접한 블럭들과는 독립적으로 처리된다. 기존의 방법[9]은 후보 벡터를 공간적인 인접한 블럭 2개와 이전 프레임의 동일 좌표를 갖는 블럭 및 그것의 4근방 블럭 4개, 그리고 영(zero)을 초기 치로 한 것 등 모두 8개의 후보 벡터를 바탕으로 초기 벡터를 결정하고, 이를 중심으로 움직임 추정을 하는 방법이다. 본 논문에서는 탐색 영역의 크기를 고정시킨 경우를 고정 크기 탐색 기법으로, 그렇지 않은 경우를 적응 크기 탐색 기법으로 명명한다. 4장에서 사용된 파라미터 T는 $\frac{M}{3}$ 으로 고정시켰으며, 여기서 M은 설정된 움직임 벡터 최대값이다.

그림 9-14는 고정 탐색 기법을 적용하여 얻어진 결과들이다. 그림 9는 탐색 범위의 크기(W)가 ±7, 블럭 크기가 8×8인 경우 football영상에 대한 결과로서 제안방법-A는 FS-BMA에 비해 움직임 보상된 화질의 평균이 4.1dB, 기존 방법[9]에 비해 1.1dB 더 높은 결과를 가져왔다. 그림 11은 블럭 크기 16×16에 대해 탐색 범위를 ±7로 적용한 것인데, 제안방법-A는 FS-BMA에 비해 2.1dB에서 4.7dB, 평균 3.2dB의 화질 향상을 가져왔으며, 기존 방법[9]에 비해 0.5dB에서 1.9dB, 평균 1.0dB의 화질 향상을 가져왔다(football 영상).

제안 알고리듬(제안방법-A)과 기존방법[9]과의 계산량을 비교할 때, 제안된 알고리듬이 약간의 부가적인 계산을 필요로 한다. 즉, 순방향 투영을 위한 계산 및 역방향 투영시 블럭 벡터를 결정할 때 소요되는 계산량이 그것이다. 그러나 이 계산량은 매우 적으며, 특히 순방향 투영의 경우 매 프레임에 대해 움직임 추정 전 프레임 단위로 한번 이루어지기 때문에 계산적 부담은 적다고 할 수 있으며, 이러한 약간의 계산량 증가에 비해 얻어지는 성능 개선효과는 실험 결과에서 볼 수 있듯이 큼을 알 수 있다.

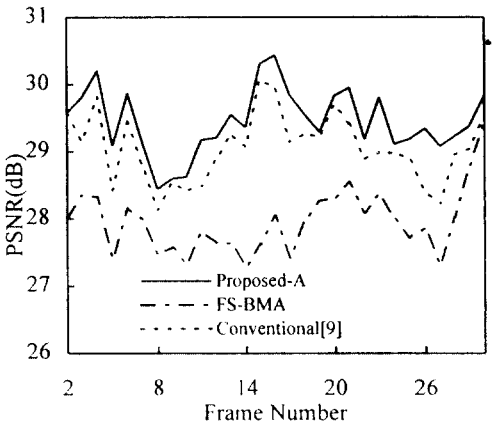

그림 10. 움직임 보상된 영상(football)의 화질(블럭 크기 = 8×8, W = ±15)

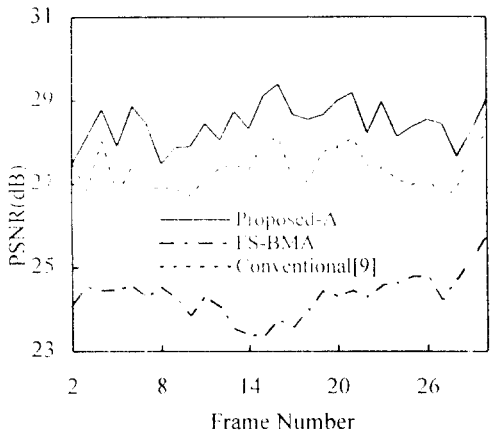

그림 9. 움직임 보상된 영상(football)의 화질(블럭 크기 = 8×8, W = ±7)

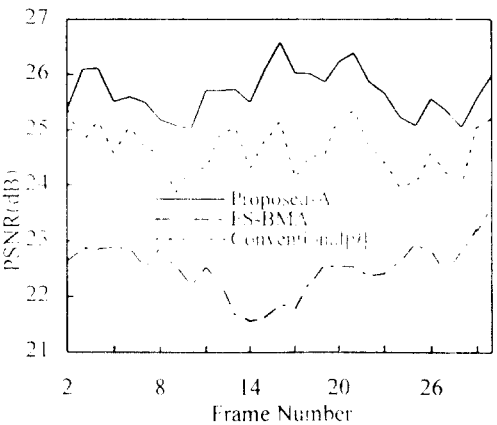

그림 11. 움직임 보상된 영상(football)의 화질(블럭 크기 = 16×16, W = ±7)

www.dbpia.co.kr

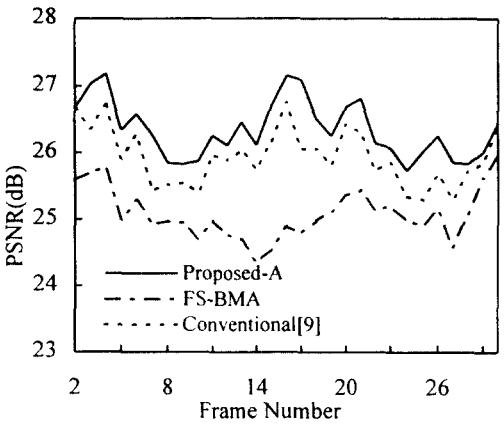

그림 12. 움직임 보상된 영상(football)의 화질(블럭 크기 = 16×16, W = ±15)

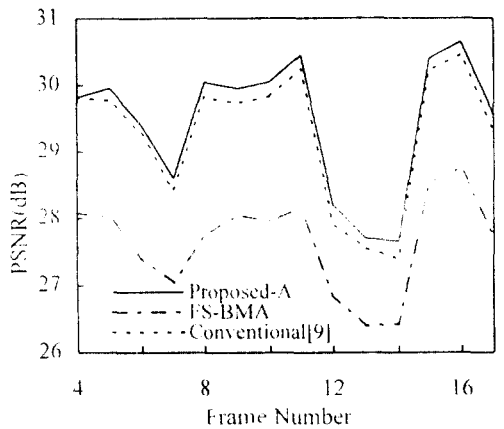

그림 13. 움직임 보상된 영상(flower)의 화질(블럭 크기 = 8×8, W = ±7)

football 및 flower 영상에 대한 실험 결과를 표1에 제시하였다. 실험 결과에서 알 수 있듯이 탐색 영역의 크기가 적을 수록 성능향상이 커짐을 알 수 있다. 이것은 움직임 추정 시 탐색 영역의 위치를 이전에

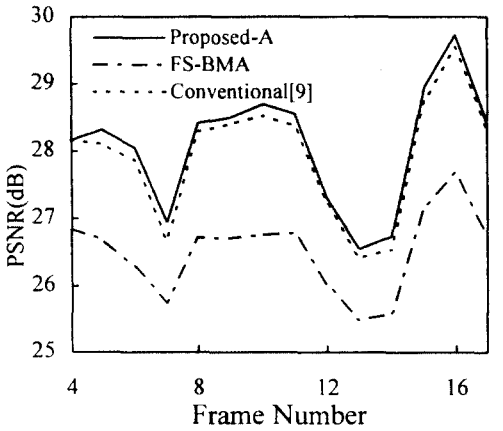

그림 14. 움직임 보상된 영상(flower)의 화질(블럭 크기 = 16×16, W = ±7)

얻어진 초기 벡터를 이용하여 이동시킴으로써 적은 계산량으로도 움직임 추정이 가능하기 때문이다. 또한, 제안방법-A를 기존방법[9]의 결과와 비교할 때 football영상이 flower영상에 비해 화질 개선 폭이 큼을 알 수 있다. 이것은 football의 경우 움직임이 빠르고 복잡해서 시공간적 인접 블럭과 상관도가 떨어지며 기존방법[9]으로는 효율적인 결과를 얻을 수 없다는 사실을 알 수 있다. 반면에 flower영상의 경우 방기존의 전역 탐색법에 비해서는 많은 성능향상을 가져오지만 기존방법[9] 역시 제안방법-A보다는 떨어지나 성능개선 결과는 좋다. 이것은 flower 영상의 경우 카메라의 이동으로 나타나는 움직임 특성으로서, 이러한 움직임 특성은 매우 선형적이며, 따라서 예측하기 쉽다.

제안 방법-B의 성능을 평가하기 위해서 FS-BMA와 비교하였는데, FS-BMA의 경우 탐색 크기가 ±15일 때 평균 화질이 약 25dB이며, 제안방법-B의 적용 시 동일한 평균 화질 값을 갖도록 C_1, C_2를 선택하였다. 그 때 C_1, C_2는 각각 0.3, 0.5로 선택되며 이 경우

표 1. 여러가지 고정 탐색 기법의 움직임 보상된 화질(평균, dB)

구 분	football(W = 7)		football(W = 15)		flower(W = 7)	
	8×8	16×16	8×8	16×16	8×8	16×16
Proposed-A	28.45	25.70	29.55	26.44	29.36	27.97
FS-BMA	24.32	22.50	27.95	25.08	27.65	26.52
Conventional[9]	27.34	24.65	29.05	25.94	29.16	27.80

www.dbpia.co.kr

각각의 계산량을 그림 15에 나타내었다. 제안방법-B는 전역 탐색법 즉, FS-BMA에 소요되는 계산량의 평균적으로 약 5%만이 소요됨을 알 수 있다. 여기서, 제안방법-B의 경우 초기 벡터 설정 시 소요되는 계산량은 미미하기 때문에 제외시켰다.

그림 16은 초기 벡터의 결정시 공간적 인접 블럭과 시간적 인접 블럭 중 시간적 인접 블럭이 초기 벡터로 선택되는 비율을 도식화 한 것이다. W = ±15일 때 블럭 크기 8×8 및 16×16에 대해 각각 조사하였으며, 움직임 벡터의 크기가 적은 블럭은 제외시켰는데 이것은 배경 영역 및 움직임이 적은 블럭의 경우 넓은 탐색을 필요로 하지 않으며 실제 넓은 탐색을 요하는 빠른 움직임에 대한 움직임 추정이 큰 의미를 갖기 때문이다. 움직임 벡터의 크기는 식(3)에서 정의한 관계식을 이용하였으며, 그 때 얻어지는 벡터의 크기가 7.0 이상인 벡터만을 대상으로 했다. 이러한 실험 환경 하에 조사 결과, 초기 벡터는 블럭 크기 88에 대해 평균 약 36%, 16×16에 대해 45%가 시간적 인접 블럭으로부터 선택됨을 알 수 있다(대상 영상: football 영상). 이 때, 제안된 알고리듬의 화질 평균은 블럭크기 8×8에 대해 약 29.55dB, 16×16에 대해 26.44dB이다.

시간적 인접 블럭을 제외한 공간적 인접 블럭만을 후보 벡터로 사용하고, 이로 부터 최소 DFD를 갖는 벡터를 초기 벡터로 사용했을 때 얻어지는 화질평균은 실험 결과, 8×8블럭에 대해 28.93dB, 16×16에 대해 약 25.92dB의 화질을 얻었다. 따라서, 시간적 인접 블럭을 적용한 후 화질 개선은 각각 0.62dB, 0.52dB에 달한다. 한편, 시간적 인접 블럭의 선택율은 블럭 크기가 클수록 높아짐을 알 수 있다. 이것은 다음과 같이 해석될 수 있다. 즉, 블럭의 크기가 크질 수록 동일한 블럭 속에서는 서로 다르게 움직이는 물체를 블럭 크기가 적을 경우보다 많이 포함하게 되며, 따라서, 인접 블럭간의 상관도가 떨어진다. 이 경우 이전 프레임의 시간적 인접 블럭의 움직임 정보를 다음 블럭의 움직임을 추정 시 이용하면 더욱 정확한 추정이 가능하게 됨을 알 수 있다.

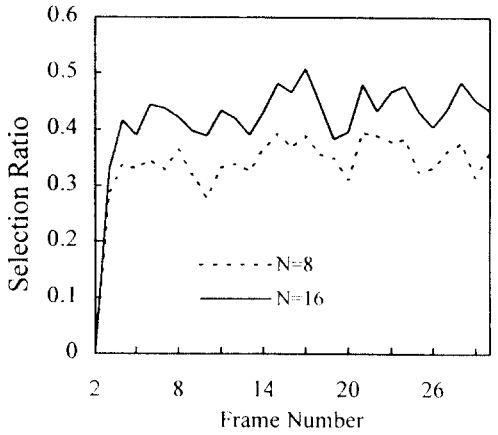

그림 16. 시간적 인접 블럭의 선택율

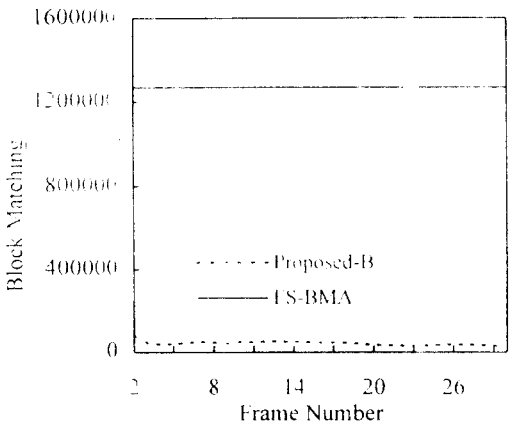

그림 15. 계산량의 비교(블럭 정합 횟수)

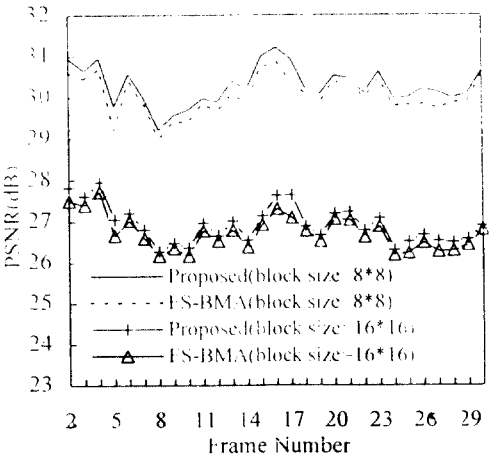

그림 17. 탐색범위가 클 경우 수렴 특성

www.dbpia.co.kr

마지막으로, 제안된 알고리듬의 수렴성을 살펴보기로 한다. 즉, 물체의 움직임을 충분히 따라갈 수 있을 정도의 탐색 범위에 대해 제안된 알고리듬이 수렴성을 가지는지의 여부에 대한 실험을 행하였다. 실험 조건은 움직임 벡터가 가질 수 있는 최대값을 탐색 범위의 2배로 설정하였다. football 영상의 경우 앞에서 언급되었듯이 가장 빠른 움직임이 거의 ±31화소를 넘지 않으며, 따라서 탐색 범위를 W=±31로 설정하고, 이 때의 움직임 벡터가 가질 수 있는 최대값 $|M_x|$, $|M_y|$를 각각 62로 설정하였다. 그림 17에 얻어진 결과를 제시하였다. 이 때, 8×8 블럭에 대한 평균화질은 전역 탐색(FS-BMA)이, 약 30.04dB, 제안방법-A가 30.28dB이며, 16×16 블럭에 대해서 각각 26.73dB, 26.94dB를 얻었다. 평균화질과 그림 17에서 알 수 있듯이 탐색 범위를 최대 움직임과 같게 했을 때도 전역 탐색보다 화질이 떨어지지 않음을 알 수 있으며, 수렴성을 가짐을 알 수 있다.

Ⅵ. 결 론

움직이는 물체는 시공간적으로 높은 상관성을 유지한다. 본 논문에서는 움직임 보상을 위해서 국부적응적으로 블럭 움직임 벡터를 추정하는 효율적인 기법이 제안되었다. 제안된 알고리듬은 계산적 복잡도를 줄이고 예측 효율을 높이기 위해 움직임 추정을 위한 초기 벡터를 선정하고, 이를 중심으로 움직임 추정 시 거의 동일한 계산량으로 높은 화질 향상을 기할 수 있다. 또한 인접 블럭 벡터의 차 및 DFD를 바탕으로 탐색 영역의 크기를 국부 적응적으로 결정하는 기법을 제안하였다. 제안된 알고리듬은 football 영상에 대해 기존의 기법에 비해 고정 탐색 기법의 경우 1.0 dB에서 4.1dB에 이르는 화질 향상을 가져왔으며, 적응 탐색 기법의 경우 비슷한 화질로 높은 계산량 단축을 가져왔다.

참 고 문 헌

1. ISO-IEC/JTC1/SC29/WG11 *MPEG* 92/175, Mar. 1992.
2. A. N. Netravali and J. D. Robbins, "Motion Compensated Television Coding," *The Bell System Technical Journal*, vol.58(3), pp. 631- 670, Mar. 1979.
3. N. Netravali and B. G. Haskell, *Digital pictures representation and compression*, New York: AT&T Bell Lab., 1988.
4. K. M. Yang, M. T. Sun, and L. Wu, "A familiy of VLSI design for the motion compensation block matching algorithm," *IEEE Trans. on Circuits and Syst.*, vol. 36, no.10, pp.176-184, May, 1994.
5. B. Liu and A. Zaccarin, "New Fast Algorithms for the Estimation of Block Motion Vectors," *IEEE Trans. on circuits and sys. for video tech.*, vol.3, no. 2, Apr. 1993.
6. H. Brusewitz, "Motion compensation with triangles," *Proc. 3rd Inter. Conf. on 64 kbit Coding of Moving Video*, Rotterdam, Netherland, Sept. 1990.
7. C. Horne, "Improving Block Based Motion Estimation by the Use of Global Motion," *Proc. SPIE VCIP 93*, Cambridge, MA, pp. 576-587, 1993.
8. S. OHalek, K. Sauer, "Temporal prediction of block motion vectors with local ambiguity-based adaptivity," *Proc. SPIE VCIP 94*, Chicago, pp.1818-1825, 1994.
9. C.Hsieh and *et al.*, "Motion estimation algorithm using interblock correlation," *Electrical Letters*, vol. 26, no.26, pp.276-277. Mar. 1990.
10. H. Chan and Y. B. Yu, "Variable size block matching motion compensation with application to video coding," *Proc. IEEE*, vol. 137, pp. 205-212. Aug. 1990.
11. D. W. Kim, J. S. Choi, and Y. S. Choi, "Adaptive Determination of Search Region for Motion Tracking," *Proc. second Asia-Pacific Conf. on Communications*, Osaka, Japan, Jun. pp.663-666, 1995.
12. C. Stiller, "Motion-Estimation for Coding of Moving Video at 8kbit/s with Gibbs Modeled Vectorfield Smoothing," *Proc. of SPIE VCIP* vol. 1360, 90, pp. 468-476, 1990.
13. Derin, Cole, Segmentation of textured Image using Gibbs Random Fields, *Computer Vision, Graphics and Image Processing* 35, pp. 72-98, 1986.

www.dbpia.co.kr

김 동 욱(Dong-Wook Kim) 정회원

1987년 2월 : 성균관대학교 전자공학과 졸업(공학사)
1992년 2월 : 중앙대학교 대학원 전자공학과 졸업(공학석사)
1996년 8월 : 중앙대학교 대학원 전자공학과 졸업예정(공학박사 학위 취득 예정)
vice Projector)에 적용되고 있으며, 처리방식은 인가
1993년 3월~현재 : 중앙대학교 대학원 전자공학과 박사과정 재학중
※주관심분야 : 영상 처리, 영상 부호화 및 통신, 컴퓨터 비전 등임

김 진 태(Jin-Tae Kim) 정회원

1963년 9월 28일생
1987년 2월 : 중앙대학교 전자공학과 졸업(공학사)
1989년 2월 : 중앙대학교 대학원 전자공학과 졸업(공학석사)
1993년 8월 : 중앙대학교 대학원 전자공학과(공학박사)
1993년 9월~1995년 2월 : 중앙대학교 기술과학연구소 선임연구원
1995년 3월~현재 : 한서대학교 전산정보학과 전임강사
※주관심분야 : 영상 부호화, MPEG, 영상통신 등임.

최 종 수(Jong-Soo Choi) 정회원

1981년 9월~현재 : 중앙대학교 전자공학과 교수
한국통신학회 논문집 제21권 2호 참조

www.dbpia.co.kr