論文93-30B-6-5

평균 피라미드를 이용한 계층적 고속 이동벡터 추정

(A Fast Hierarchical Motion Vector Estimation Using Mean Pyramids)

南 權 文*, 金 俊 植**, 朴 來 弘**, 沈 英 錫***

(Kwon Mun Nam, Joon-Seek Kim, Rae-Hong Park and Young Serk Shim)

要 約

본 논문에서는 피라미드구조를 이용한 계층 이동벡터 탐색 알고리듬을 제안한다. TSS (three step search)에 근거한 제안한 알고리듬은 피라미드의 각 레벨에서 기존 방법에서의 정합 창틀보다 더 작은 창틀을 사용하므로, 계산량을 매우 줄이면서 TSS에 필적할만한 성능을 가진다. 피라미드의 다음 레벨에서 이동벡터 추정의 초기 탐색 지점이 되는 이동 벡터 후보의 수를 증가시킴으로써 성능을 더욱 향상시킨다. 이때 제안한 계층적 알고리듬의 계산량은 후보 이동벡터의 수에 의존하게 되고, PSNR (peak signal to noise ratio)은 TSS와 full search 기법의 사이값이 된다. 두가지의 다른 블럭 크기와 다양한 실험 시퀀스를 이용한 실험 결과를 보이고, 그 하드웨어 구현을 간략하게 보인다.

Abstract

In this paper, a hierarchical motion vector estimation algorithm using pyramidal structure is proposed. Using a smaller measurement window at each level of a pyramid than that of the conventional scheme, the proposed algorithm, based on the TSS (three step search), reduces the computational complexity greatly with its performance comparable to that of the TSS. By increasing the number of candidate motion vectors which are to be used as the initial search points for motion vector estimation at the next level, the performance improves further. Then the computational complexity of the proposed hierarchical algorithm depends on the number of candidate motion vectors, with its PSNR (peak signal to noise ratio) ranging between those of the TSS and the full search method. The simulation results with two different block sizes and various test sequences are given and its hardware implementation is also sketched.

Ⅰ. 서론

동영상 전송에 있어서 전송 데이타량을 줄이기 위한 방법으로 interframe 부호화가 효율적이다.[1,2] 특히, 연속한 영상간의 움직임을 고려한 MCI (motion compensated interpolation)와 MCP (motion compensated prediction)의 이동보상방법은 영상부호화의 효율을 증대시킨다.[3] 이러한 움직임 보상기법은 PRA (pel recursive algorithm)

*準會員, **正會員, 西江大學校 電子工學科
(Dept. of Elec. Eng., Sogang Univ.)
***正會員, 生産技術硏究院
(Korea Academy of Industrial Technology)
(※본 연구는 HDTV 수상기 개발 사업의 일환으로 생산 기술 연구원 주관하의 위탁연구비로 이루어진 것임.)
接受日字: 1992年 10月 29日

와 BMA (block matching algorithm)로 나뉘며, 후자인 BMA는 시간적으로 중복되는 영역들을 블럭 단위의 이동벡터를 이용하여 동영상을 압축하는 기술로서 블럭단위로 처리하기 때문에 PRA보다는 PSNR 성능이 떨어지는 반면, 계산량이 적고, 하드웨어 구성이 용이하여 널리 사용된다.

BMA의 대표적인 방법으로 탐색 영역내의 모든 점들을 후보점으로 하여 이동 벡터를 추정하는 FS (full search) 또는 brute force search 방식이 있다. 그러나 이 방법은 추정오차 측면에서는 최적이나 계산량이 방대하기 때문에 이것을 보상하기 위한 방법으로 제안된 것들이 DMD (direction of minimum distortion)[4], TSS (three step search)[5] 등의 고속 BMA이다. DMD는 중심점의 사방을 좁아가면서 탐색영역 내에서 최적의 이동을 찾는 방법이고,[4] TSS는 중심점으로부터 각 단계별로 3, 2, 1의 간격을 두고 9개의 지점에 대해서 최적의 움직임을 추정해 나가는 방식으로, DMD방식에 비해서 하드웨어의 구현이 용이하므로 많이 사용된다.[5] CCITT에서는 이 TSS를 처음의 이동변위를 4로 교체하여 보다 큰 움직임에도 이동 검출이 효율적인 변형된 TSS를 권고하고 있다. 이밖에도 계산시간 단축을 위해 많은 알고리듬이 제안되어 왔다. 또한 계산량의 문제를 떠나서 이동벡터 전송시 과중한 데이타량을 줄이기 위한 방법으로 많은 계산량에도 불구하고 비교적 구해진 이동벡터가 균일한, 계층적인 이동벡터 검출기법[6]도 제안되었으나 예측오차 감소의 효과를 갖지 못하고, 계산량도 줄이지 못하는 단점이 있다.

계산량을 줄이기 위한 알고리듬에서는 계산량을 줄인 만큼 성능저하가 나타나는 문제를 안고 있으며, 그 중 비교적 우수한 성능을 지니는 고속알고리듬인 TSS는 CCITT의 H.261에서 이동벡터 검출을 위한 기법으로 권고되고 있다. MPEG (moving picture expert group)에서는 계산량의 무리에도 불구하고 예측오차를 줄이기 위해 이동벡터의 검출에 FS를 사용하고 있다. 따라서 계산량을 줄이면서 이에 따른 성능저하를 극복할 수 있는 고속 알고리듬의 개발은 상용화될 시스템에서의 실시간 처리와 과중한 데이타량을 줄이기 위해 필수적이라고 하겠다.

본 논문에서는 CCITT에서 권고하고 있는 변형된 TSS와 같은 변위 즉, 이동벡터의 변위를 4, 2, 1로 유지하면서 고속으로 이동벡터를 검출하고, 예측 오차도 줄일 수 있는 기법을 제안하였다. 이 기법은 필요에 따라 계산량과 예측오차의 효율면을 조절할 수 있는 기법이라고 할 수 있다. 즉, 계산량은 TSS의 절반 수준에서 FS의 절반수준까지 조절되면서, 예측오차의 성능은 TSS 수준에서 FS 수준까지 조절되는 알고리듬이다.

Ⅱ장에서는 기존의 계층적 BMA 기법을 살펴보고, Ⅲ장에서는 피라미드구조를 이용한 제안한 계층적인 이동벡터 추정 방식을, Ⅳ장에서는 제안한 방법의 화질을 더욱 개선하기 위한 기법을 소개하고, Ⅴ장에서는 제안한 기법과 기존의 방법간의 계산량 비교에 대해, Ⅵ장에서는 제안한 기법의 하드웨어 구현에 대해, 그리고 Ⅶ장에서는 컴퓨터 모의 실험으로 기존의 방법과의 비교와 MPEG에의 적용실험에 대해 알아보고, Ⅷ장에서 결론을 맺는다.

Ⅱ. 기존의 계층적 블럭 정합 알고리듬

기존의 계층적인 BMA에는 Biering[6]이 제안한 기법과, 오차함수의 계층적 근사를 이용한 기법,[7] 그리고 인접 이동벡터간의 기하학적 연관성을 이용한 계층적 구조를 갖는 이동벡터 추정기법[8] 등이 있다. [7] 은 탐색영역 전체에 대해 이동벡터 후보를 설정하는 기법으로 오차함수를 단계별로 평균값으로 근사시켜서 FS에 비해 고속 알고리듬으로 제안됐으나, 파라메터의 조절이 영상에 의존적이다. 또한 [8] 은 단계별로 이동벡터를 보정하여 인접이동벡터간의 연관성을 이용하여 이동벡터를 보정하는 알고리듬으로 제안되었으나, 계산량이 과중한 단점이 있다. 본 장에서는 대표적인 계층기법인 [6] 에 대해 알아 본다.

Biering이 제안한 HBMA (hierarchical BMA)는 이동벡터의 탐색을 세 단계로 나누어 각 단계별로 다른 창틀 크기를 갖는 저역 필터를 사용하여 만들어진 영상을 바탕으로 처음 단은 8 x 8, 다음 단은 4 x 4, 마지막 단은 2 x 2 크기의 블럭단위로 이동벡터를 찾아가는 계층적인 이동벡터 추정법이다. 이 경우 각 단에서 오차를 계산하는 창틀의 크기는 각각 64 x 64, 28 x 28, 12 x 12이고, 상위단의 이동벡터는 다음 단에서 선형 보간법을 통해서 크기가 작아진 각 블럭의 초기 이동벡터를 결정하게 된다. 이 보간법을 통해서 마지막에는 한화소 단위까지 이동벡터를 찾아낼 수 있다. 또한 각 단에서의 이동벡터 탐색은 단계별로 TSS를 사용하면서 ±7, ±3, ±1씩 변위를 갱신하는 기법을 사용한다.

HBMA에서는 저역 필터와 선형보간법에 따라서 최종결과가 달라지게 되며, 저역 필터링된 영상위에서 이동벡터를 검출하기 때문에 이동벡터들이 비교적 균일한 장점은 있으나, 각 블럭의 세밀한 움직임을

놓치는 단점과, 블럭정합을 위한 계산량이 큰 단점등이 있다. 이 기법을 CCIR 601 표준영상에 대해 실험한 결과를 표 1에 FS, TSS의 결과와 함께 실었다. 이때, 저역 필터링에는 [6] 에서 고속알고리듬으로 소개하고 있는 단순 평균법을 사용했다.

표 1. 블럭정합 알고리듬의 평균 PSNR(8×8)
Table 1. Average PSNR of block matching algorithms (8×8).

(단위 : dB)

방 법	Table T.	Flower G.	Susie	Popple	Foothall	Mobile
TSS	26.91	26.45	37.71	26.86	25.73	23.35
HBMA	25.41	27.72	37.92	26.85	24.50	25.70
FS	27.39	28.13	38.94	29.12	26.83	27.04

표 1에서 보듯이 그 성능이 전체적으로 FS 보다 떨어지며 TSS에 비해서도 못한 성능을 보이기도 한다. 결과를 살펴본 바에 의하면 영상전체가 균일한 움직임을 갖는 "Mobile"과 "Flower Garden" 영상에서는 전역적인 (global) 움직임이 잘 찾아지므로 TSS보다 좋은 성능을 보인 것을 알 수 있으며 움직임이 적은 "Susie" 영상에서도 각 계층별로 이동벡터의 refinement가 효과를 보인 것을 알 수 있다. 그러나 모든 영상이 균일한 움직임을 갖지는 않으므로 작은 영역에서 다양한 움직임을 갖는 "Popple", "Football" 그리고 "Table Tennis" 영상에서는 성능을 개선시키지 못하는 문제가 있다. 또한 저역 필터링과 정합을 위한 계산에서 기존의 BMA가 사용하는 창틀보다 큰 창틀을 사용하기 때문에 추가되는 계산량이 큰 점과 PSNR 성능이 개선되지 못한 점 등의 문제가 있다. 따라서 Ⅲ장에서는 PSNR 성능을 개선하며 계산량 또한 감소시킬 수 있는 TSS에 근거한 계층적인 움직임 검출 기법을 제안한다.

Ⅲ. TSS에 근거한 제안한 계층적 블럭 정합 알고리듬

계층구조를 이용하는 영상처리 기법은 다양한데 피라미드 구조를 이용하여 상위계층에서 하위계층으로 움직임을 찾아나가는 기법은 계층구조의 특성상 상위계층에서 영상의 크기가 작아지므로 이동벡터 탐색계산량을 줄일 수 있다. 여기서 제안하는 알고리듬은 이러한 피라미드구조를 이용하여 이동벡터를 찾아나감으로써 계산량을 줄이고, 기존의 알고리듬에 필적할만한 PSNR 성능을 갖도록 구현하였다. 더우기, TSS의 탐색과정이 각 단계마다 이동변위가 4, 2, 1로 결정되어 있으므로 이러한 특징을 이용하여 피라미드를 구성하면 피라미드의 각 계층에서의 이동벡터가 다음 계층의 초기값으로 주어질 때, TSS에서의 각 단에서 다음 단으로 이동벡터가 전달되는 것과 같은 효과를 얻을 수 있고, 최대 이동변위도 결국 ±7로, 변형된 TSS의 탐색영역내에서의 이동벡터 추정시의 최대 이동변위와 같다.

본장에서는 먼저 피라미드를 구성하는 방식에 대해서 살펴보고, 다음으로 이동벡터의 검출기법 및 half-pel로의 확장에 대해 살펴본다.

1. 계층적 피라미드의 구성

블럭정합을 위해 M x N 크기의 부블럭을 사용할 경우, TSS는 각 단계마다 최적의 정합점을 찾기 위해 M x N 블럭의 모든 화소들을 사용하여 각 블럭의 오차를 계산하며 각 단계별로 9번의 계산이 필요하므로 한 블럭의 이동벡터를 찾기 위해서는 중복되는 계산을 피하면 25번의 M x N 블럭의 계산이 필요하다. 이러한 계산을 피라미드 영상을 이용하여 각 단계별로 피라미드 영상에서 정합계산을 할 경우, 피라미드의 상위계층에서는 블럭을 구성하는 화소수가 줄어들게 되므로 그 계산량을 줄일 수 있다.

피라미드 영상을 단순히 부표본화하여 구성할 경우 부표본화된 화소가 원래 해상도의 전역적인 특성을 나타내기에는 부족하며 이러한 잡음의 영향으로 상위계층에서 정확한 이동벡터의 검출이 이루어지지 않아 추정오차 측면에서 TSS에 비해 큰 성능저하를 보인다. 따라서 상위계층에서 coarse한 이동벡터를 더 정확히 검출할 수 있도록, 저역필터를 사용하여 피라미드 영상을 구성하는데, Gaussian 저역필터[9]를 사용할 경우 추정오차를 어느 정도 줄이기는 하나 피라미드 영상을 구성하는데 있어서 많은 계산량이 필요하므로 고속알고리듬을 위한 본 연구의 목적에 어울리지 않는다. 따라서 다음에 설명하는 바와 같이 단순 평균법을 사용하여 피라미드 영상을 구성하는데, L번째 계층의 (x, y) 위치에서의 화소값을 $g^L(x, y)$라고 할 때, $g^0(x, y)$는 원래 해상도의 화소값을 의미하며, 피라미드 영상은 식 (1) 및 그림 1에서와 같이 하위계층의 4화소를 non-overlapping방법으로 평균값을 truncation하여 상위계층의 한화소를 구성하는 간단한 저역필터를 사용하여 구성된다. 따라서 제일 상위계층에서의 해상도가 가장 떨어지지만 저역성분만이 남으므로 부근의 전역적인 움직임을 찾기가 용이하고 잡음으로 인한 오류가 줄어들게 되어 탐색블럭의 coarse한 이동벡터를 구하는데 용이하고, 또한 탐색 계산량을 줄일 수 있다.

$$g^L(x,y)=\frac{1}{4}\sum_{m=0}^{1}\sum_{n=0}^{1}g^{L-1}(2x+m,\ 2y+n),\ 0<L\le 2 \quad (1)$$

이때, L을 2까지 제한한 것은 피라미드를 3계층까지만 구성하여 최대 이동변위를 ±7까지 제한하기 위해서이다. 이렇게 구성된 피라미드에서 역으로 계층 2로부터 계층 0으로 이동벡터를 찾아 나간다. 이때 각 계층마다 찾아진 이동벡터를 다음의 하위계층으로 전송하여 하위계층의 이동벡터 초기값으로 사용하는데 제안한 피라미드구조를 이용한 이동벡터 추정은 다음과 같다.

2. 각 계층에서의 이동벡터 추정

그림 1과 같은 피라미드 영상이 구성되면 탐색과정은 구성된 피라미드의 계층 2로부터 역으로 계층 0으로 이동벡터를 찾아 나간다. 이때 그림 2와 같이 각 계층마다 찾아진 이동벡터를 다음의 하위계층으로 전송하여 하위 계층의 이동벡터 초기값으로 사용하는데, 다음과 같이 이동벡터를 추정하게 된다. 그림에서 숫자는 계층에 따라 구해진 이동벡터를 나타낸다.

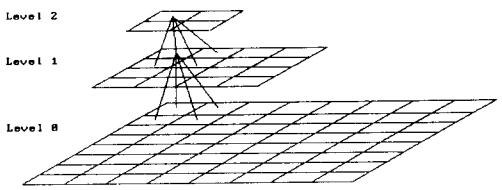


그림 1. 영상 피라미드 계층 구조

Fig. 1. Hierarchical structure of an image pyramid.

블럭 정합 알고리듬에서 사용되는 일반적인 정합기준으로는 MSE (mean square error), NCCF (normalized cross correlation function), 그리고 MAD (mean absolute difference)등이 있으나, 일반적으로 계산량이 가장 적으면서 오차 측정이 정확한 MAD를 많이 사용한다. MAD는 탐색 부블럭의 크기를 M×N이라고 할 때 식 (2)로 표현된다.

$$MAD(m,n)=\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}|g_k(i,j)-g_{k-1}(i+m,\ j+n)| \quad (2)$$

단, m, n은 각 단계별로 탐색영역내에서 제한된 변위이고, $g_k(i,j)$는 k번째 영상의 (i,j) 위치의 화소값이다. 제안한 방법도 MAD를 사용하는데, Ⅲ. 1에서 전술한 바와 같이 피라미드에서 계층 2의 한 점은 계층 0의 4×4 블럭, 계층 1의 2×2 블럭에 해당하고, 계층 1의 한 점은 계층 0에서의 2×2 블럭에 해당하므로, 계층 2에서의 2×2 블럭과 계층 1의 4×4 블럭은 계층 0의 8×8 블럭에 해당된다. 따라서 식 (2)의 M, N은 각 계층에서 각각 $M/2^L$, $N/2^L$ (L = 0, 1, 2)으로 대체되어 식 (3)으로 표현된다. 즉,

$$MAD^L(m,n)=\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}|g^L{}_k(i,j)-g^L{}_{k-1}(i+m,\ j+n)| \quad (3)$$

로 주어지며 여기서 L은 계층을 나타내고, M = $M/2^L$, N = $N/2^L$, m, n = 0, ±1이다. 따라서 각계층에서는 식 (3)을 이용하여 각 계층의 이동벡터를 찾게 된다. 이때 상위 계층에서 탐색된 이동벡터는 하위 계층으로 전달되어 하위 계층의 이동벡터의 초기값이 되며 하위 계층에서 추정된 이동벡터는 k번째 영상의 L번째 계층에서의 이동벡터를 $MV^L{}_k(x,y)$라고 할 때, 식 (4)와 같이 일반화된다.

$$MV^L{}_k(x,y)=2*MV^{L+1}{}_k(x,y)+\triangle MV^L{}_k(x,y) \quad (\text{단, } L=1,0) \quad (4)$$

여기서 (x, y)는 탐색블럭의 위치이고, $\triangle MV^L{}_k(x,y)$는 L번째 계층에서의 이동변위가 되며, 이 경우 계층 2의 한 점 간격은 최하위 계층에서의 4 점 간격, 계층 1의 한 점은 2 점 간격이 된다. 그리고 계층 0의 1 점 간격을 포함하여, 4, 2, 1의 이동벡터 변위가 되어 변형된 TSS의 이동벡터 변위 추정 방식을 따르게 되고, 따라서 $MV^0{}_k(x,y)$의 최대 변위는 ±7 화소가 된다.

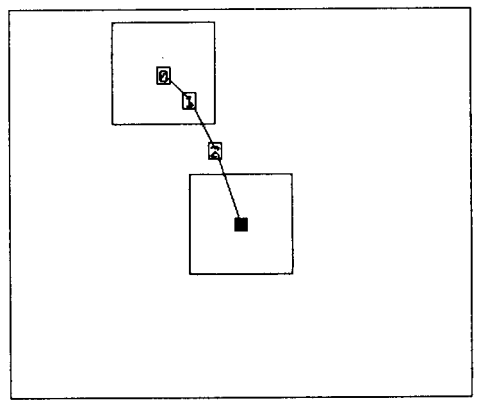


그림 2. 단일 후보의 계층적 이동벡터 추정

Fig. 2. Hierarchical motion vector estimation with one candidate.

그림 2의 각 후보블럭과 탐색과정을 1차원적으로

나타내면 그림 3과 같다. 즉, 각 계층에서 초기 이동벡터를 중심으로 이동벡터를 찾고 하위 계층으로 이동벡터를 전달하게 된다. 이때 각 계층에서의 탐색영역과 유사도 계산에 필요한 영역을 나타내면 그림 4와 같은데 하위 계층으로 갈수록 그 크기가 증가하는 것을 알 수 있다.

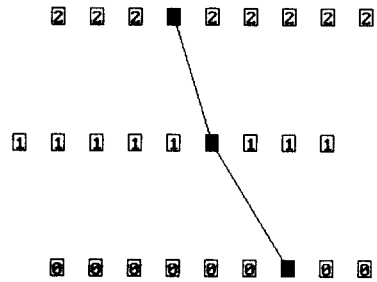


그림 3. 그림 2의 일차원적 표현
Fig. 3. One-dimensional representation of Fig. 2.

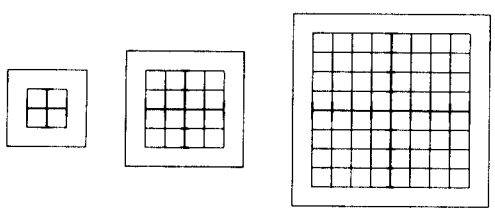


그림 4. 각 계층에서의 탐색영역과 오차 측정 블럭
Fig. 4. Search area and error measurement block at each level.

3. Subpel 탐색

BMA의 결과, 나타나는 예측오차의 성능을 개선하기 위한 방안으로 subpel탐색기법이 연구되고 있다. 이 기법은 BMA로부터 구한 이동벡터를 중심으로, 한 화소보다 작은 , ±1/2, ±1/4등의 변위로 이동벡터의 변위를 확장시키면서 PSNR 성능을 개선시키기 위한 방법이다. 여기서는 크기의 ±1/2이동벡터 변위의 확장으로 성능개선을 살펴본다. Subepl 탐색에서는 먼저 아래 식 (5)의 보간 방법을 사용해 half-pel 단위를 갖는 영상을 만들고 그림 5의 순서로 탐색을 하며 최소의 MAD를 갖는 이동벡터를 찾는다.

$$S(x+0.5, y) = (S(x, y) + S(x+1, y)) // 2$$
$$S(x, y+0.5) = (S(x, y) + S(x, y+1)) // 2$$
$$S(x+0.5, y+0.5) = (S(x, y) + S(x+1, y) + S(x, y+1) + S(x+1, y+1)) // 4 \quad (5)$$

여기서 // 연산은 나눗셈후 반올림을 나타낸다. 이렇게 찾아진 이동벡터는 half-pel 단위를 갖기 때문에 이동벡터 전송시 overhead가 증가하는 단점이 있으나, 예측오차 감소로 인한 성능향상과의 trade off로 MPEG SM3 (simulation model three)에서 사용되는 기법이다. MPEG SM3에서는 FS와 half-pel 탐색을 함께 사용하므로 제안한 기법에 half-pel 탐색 기법을 접목하여 이에 따른 예측오차의 향상정도를 실험, 비교한다.

Ⅳ. 성능개선을 위한 제안한 기법의 변형

Ⅲ. 2에서 설명한 탐색 기법은 그 성능에 있어서는 TSS에 필적할만한 예측오차를 갖고, 계산량은 TSS의 절반 수준의 계산량을 갖는다. 그러나 여전히 FS와 고속 알고리듬 간의 성능차이를 줄이지는 못하며 계산시간만의 감소는 영상의 화질 향상에 도움을 주지 못하므로 예측오차 측면에서 FS에 가까운 성능을 가지면서 계산량이 많이 줄어든 알고리듬이 필요하다. 이러한 목적에 맞도록, 제안한 방법의 문제점들을 해결하여 계산량이 약간 증가하더라도 예측오차를 많이 줄일 수 있도록 다음과 같이 알고리듬을 수정 제안한다.

제안한 기법이 TSS에 비해 우수한 성능을 보이는 부분도 있는데 이것은 잡음의 영향을 감소시키는 저역필터를 사용하여 피라미드 영상을 형성했기 때문이다. 그러나, FS와 TSS를 비교할 때 나타나는 차이만큼의 성능저하가 예측오차 측면에서 나타난다. TSS의 경우에 제한된 탐색 영역내의 모든 화소에 대해 정합을 하지 않기 때문에 처음 단계에서 local minima에 빠졌을 경우 이 지역에서 벗어나지 못하기 때문으로 풀이된다. 제안한 기법도 이와 같이 피라미드 영상을 구성할 때의 truncation연산으로 인해 최상위 계층에서 local minima에 빠지면 단계별로 후보벡터가 하위계층으로 전달될 때, 이 지점을 벗어나지 못하는 문제가 발생하며 이로 인해 전체적으로 예측오차가 FS의 수준에 도달하지 못하고 TSS의 수준에 머무는 것을 볼 수 있다. 이러한 local minima 문제를 해결하면 전체적으로 예측오차의 성능을 개선시킬 수 있을 것으로 기대되며 따라서 다음과 같은 방법으로 이러한 local minima 문제를 해결한다.

최상위 계층에서 계산된 9개의 MAD값은 비교적

작은 블럭에 대한 값으로서 같은 수치를 갖는 지점이 한개 이상 나타나는 경우가 있다. 이때의 후보선정 기준 순위는 TSS에서와 같은 순서로 결정되는데, 이때에 local minima에 빠지는 확률이 큰 것을 알 수 있다. 따라서 local minima 문제를 해결하기 위한 방법으로 각 계층에서 하위계층으로의 이동벡터의 초기화에서 후보의 개수를 증가시키는 방법을 제안한다. Ⅲ장에서 언급한 원래 제안된 방법은 각 계층에서 계산된 MAD값중 최소인 값에 해당되는 이동벡터 하나만을 하위단으로 전송하여 이동벡터의 초기값을 결정하나, 본 장에서 제안하는 후보의 개수를 늘리는 방법은 임의의 개수로 후보의 수를 제한하고, 제한된 개수에 따라 최상위단에서 결정된 후보벡터들을 각각 독립적으로 하위단으로 전송하고, 하위단에서 각각에 대해 최소의 MAD값을 갖는 이동벡터를 찾아 다음단으로 이동벡터를 전송하며 최종단에서 정해진 후보 개수만큼의 이동벡터에 대한 MAD값이 최소인 이동벡터를 기준 블럭의 이동벡터로 결정하게 된다. 이때 선정되는 후보의 개수는 최상위단에서의 탐색영역의 크기에 의해 제한되는데, 최상위단은 근본적으로는 중심화소로부터 4화소 떨어진 지점에 대한 이동벡터 탐색의 단계이고, 최상위단의 한 화소가 최하위단의 16화소에 해당하며, 이에 따라 탐색하는 지점도 9개로 제한되므로 결과적으로 선정될 수 있는 후보의 개수는 최대 9개가 된다.

1	2	3
4	0	5
6	7	8

그림 5. Subpel 단위탐색 순서
Fig. 5. Order of Subpel Search.

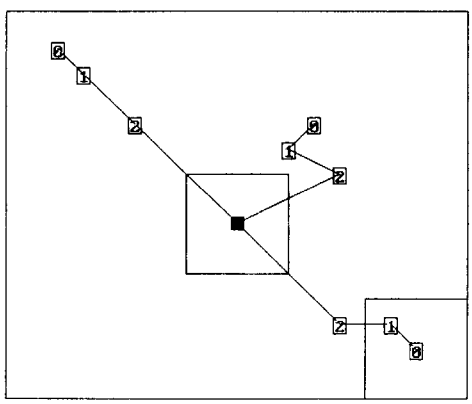


그림 6. 후보가 셋일 때의 탐색과정
Fig. 6. The search procedure with three candidates.

이러한 제한을 벗어날 경우 7의 최대변위를 넘는 탐색방법이 되므로 후보의 개수는 최대 9개로 제한했다. 이러한 방법은 제안된 원래 방법에 비해서는 많은 화질의 향상을 꾀할 수 있는 반면, 계산량이 선형적으로 증가하는 단점이 있다고 할 수 있으나, 결과적으로는 FS의 PSNR 성능에 다가가면서, FS에 비해서는 훨씬 적은 계산량을 갖는다. 이를 그림으로 나타내면 그림 6, 그림 7과 같다. 그림 6은 후보를 셋으로 제한했을 경우 각 계층마다 독립적으로 탐색하여 세개의 후보를 찾아나가는 과정으로 세 후보에 대해 독립적으로 탐색한 후 최하위 계층에서 이동벡터를 결정한다. 그림 7은 후보를 둘로 제한했을 경우 각 계층에서의 후보탐색의 일차원적 표현을 보이고 있다.

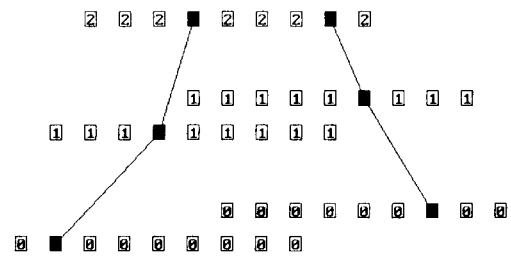


그림 7. 후보가 둘일 때의 일차원적 표현
Fig. 7. One-dimensional representation with two candidates.

Ⅴ. 제안한 기법의 계산량 비교

Ⅲ장에서 제안한 바와 같이 후보개수를 하나로 제한할 경우, 16 x 16 탐색블럭에 대한 계산량을 고려하면 다음과 같다. TSS에 대해서는 256번의 뺄셈과 256 번의 절대치 연산, 255번의 덧셈연산이 25개의 탐색지점에 대해 행해지므로 총 6,400번의 뺄셈, 6,400번의 절대치 연산과 6,375번의 덧셈연산이 필요하게 되며, 제안된 알고리듬에 대해서는 탐색 후보를 하나로 했을 때, 다음과 같은 계산량이 요구된다.

계층 2에서 16번의 뺄셈과 16번의 절대치 연산, 그리고 15번의 덧셈 연산,

계층 1에서 64번의 뺄셈과 64번의 절대치 연산, 그리고 63번의 덧셈 연산,

계층 0에서 256번의 뺄셈과 256번의 절대치 연산,

그리고 255번의 덧셈 연산이 각각 9개의 탐색 지점에 대해서 필요하다. 따라서 결과적으로는 3,024번의 뺄셈과 3,024번의 절대치 연산, 그리고 2,997번의 덧셈연산이 이동벡터를 찾는데 필요한 계산량이 되며, 각 계층 피라미드 영상을 구성하는데 있어서 16 x 16 크기의 최하위 블럭이 8 x 8 크기의 중간 단계의 블럭을 형성하고, 이것이 다시 4 x 4 크기의 최상위 계층의 블럭을 형성하게 되므로 전체적으로 80 x (3번의 덧셈과 1 번의 나눗셈)이 필요하게 된다. 이 나눗셈은 균등하게 4로 나누는 연산으로 shift right로 처리가능하므로 덧셈 등의 연산에 비해 overhead가 크다고 할 수 없다. 따라서, 제안한 방법은 TSS의 55%정도의 계산량을 갖는 알고리듬이라고 할 수 있다. 위의 16 x 16 블럭 탐색의 경우 subpel까지의 탐색을 고려할 때는 모든 BMA에 아래와 같은 동일한 계산량이 추가된다. 먼저 식 (5)의 subpel단위의 화소를 보간하기 위해서 상하나 좌우의 한방향으로 보간하는 경우 16 x 17 (또는 17 x 16) 크기의 half-pel 영상이 필요하고 사방으로 보간하는 경우 17 x 17의 half-pel 영상이 필요하므로 식 (6)과 같은 계산량이 추가되며, 8개 지점에 대해서 MAD를 계산하기 위해서는 식 (7)과 같은 계산량이 추가된다.

16x17x(1 덧셈, 1 나눗셈)x2 + 17x17x(3 덧셈, 1 나눗셈) = 1,411 덧셈, 833 나눗셈. (6)

8 x (256 뺄셈, 256 절대치, 255 덧셈) = 2,048 뺄셈, 2,048 절대치 2,040 덧셈. (7)

이러한 추가되는 계산량을 표 2에 나타냈다. 또한 Ⅳ장에서 설명한 이동벡터의 후보를 증가시키는 경우에는 후보가 하나 증가할 때마다, 후보가 하나일 때의 계산량이 동일하게 증가된다고 볼 수 있으므로 후보가 n개라면 n x (후보가 하나일 때의 이동벡터 탐색시 계산량)의 계산량을 가지므로 선형적으로 계산량이 증가한다고 할 수 있으며 피라미드 영상을 만들 때 추가되는 계산은 물론 한번만의 계산으로 충분하다.

표 2. 제안한 기법의 계산량 비교 (16×16)
Table 2. Comparison of the computational complexity of the promethod (16×16).

방 식	뺄 셈	절대치	덧 셈	추가 영상 구성
FS	57,600	57,600	57,375	
TSS	6,400	6,400	6,375	
제안	3,024	3,024	2,997	80×(3덧셈, 1나눗셈)
subpel	2,048	2,048	2,040	1,411 덧셈, 833 나눗셈

Ⅵ. 제안한 기법의 하드웨어 구현

제안된 알고리듬은 계층별로 독립된 연산을 수행하므로 병렬연산의 효과를 얻을 수 있다. 이때 한 블럭에서의 이동벡터를 찾기 위해서는 상위계층에서의 이동벡터가 하위계층으로 전송되어 그 단계에서의 초기벡터를 형성해야 하므로 처음 2단계의 계산은 모든 계층이 동시에 동작하지는 않는다. 그러나 3번째 블럭에 대해 최상위단의 이동벡터를 검출할 때에는 모든 단계의 연산이 독립적으로 이루어질 수 있으므로 병렬처리가 가능해진다. 그림 8에 나타낸 블럭선도는 제안된 기법중 후보벡터의 수를 하나로 선정했을 경우를 중심으로 설계한 것이다.

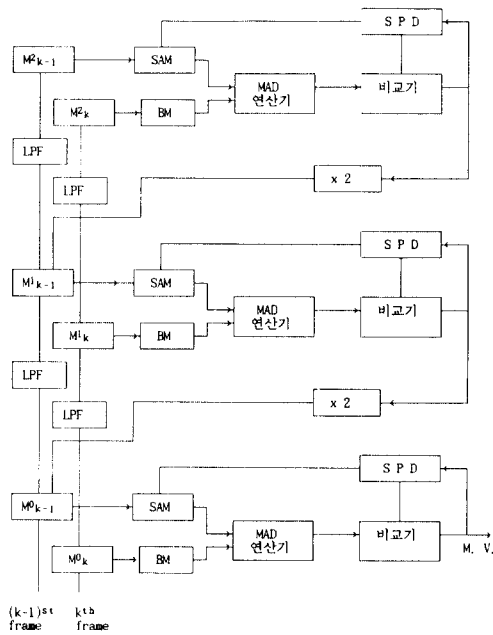


그림 8. 계층적 이동벡터 검출기의 구성도
Fig. 8. Structure of a hierarchical motion vector estimator.

k번째 frame과 (k-1)번째 frame이 입력되면 각각의 frame을 저역필터를 사용하여, 크기가 전체적으로 1/4 (가로 1/2, 세로 1/2)인 영상을 앞에서 설명한 바와 같이 구성하여 메모리에 저장한다. 이때 입력영상이 K x L의 크기를 갖는다고 하면, M^2_k, M^1_k, M^0_k을 위해 각각 KL / 16과 KL / 4, KL의 메모리가 필요하게 되므로 영상피라미드를 구성하여 각 frame의 처리가 끝나기까지는 2 x (KL / 16 + KL / 4 + KL)의 메모리가 필요하다. 이렇게 구성된

피라미드 영상으로부터 탐색할 블럭에 대한 정보를 바탕으로, BM (block memory)과 SAM (search area memory)에 각각 현재 frame의 탐색할 블럭과 이전 frame의 탐색영역이 저장되고, SPD (search point determination) unit에 의해 계산된 SAM내의 블럭과 BM의 화소값들이 MAD연산기를 거쳐 MAD를 계산하고 이 값은 비교기에 저장되어 있던 최소의 MAD값과 비교하여 최소의 MAD값과 그 값을 갖는 지점의 주소를 비교기에 저장하게 된다. [10] 이러한 탐색이 한 블럭에 대해서 끝나면 결정된 이동벡터의 두배에 해당되는 지점의 주소를 하위단의 $M^{l}k\text{-}1$에 전달하여 이점을 중심으로 하는 탐색영역이 SAM으로 저장되고, 동시에 BM에도 현재 frame에서의 값이 저장되어 위와 같은 연산을 통해 하위단으로 이동벡터를 전송하게 되며, 최종단에서는 같은 연산을 통해 탐색블럭의 이동벡터가 검출된다. 이때, 각각의 계층이 상위계층에서 주어지는 초기 탐색지점 (최상위 계층은 중심지점)을 중심으로 이동벡터를 검출하므로 처음 상위 두계층의 탐색이 이루어진 후, 각 계층이 동시에 이동벡터의 탐색이 가능하다. 즉, 처음의 두 블럭에 대한 최상위 계층의 탐색이 끝난 후에는 동시에 세 계층에서의 이동벡터 검출이 가능해지므로 한 블럭에 대한 이동벡터 검출을 위해서는 최고의 해상도인 최하 계층에서 9개의 지점에 대해서 MAD계산을 통해 이동벡터를 검출하는 것과 같은 계산량만이 요구되므로, 제안된 계층적인 이동벡터 검출기를 이와 같은 병렬 처리가 가능한 형태로 구성할 경우 더욱 빠른 이동벡터 검출이 가능해진다.

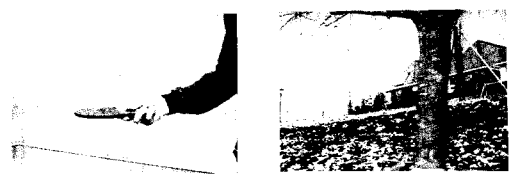
탐색의 복잡도 면에서 TSS와 비교하면 TSS는 각 단계별로 4,2,1의 화소 간격의 이동벡터를 탐색하므로 탐색간격을 단계별로 조정할 필요가 있다. 그러나, 제안된 기법은 먼저 피라미드 영상을 구성한 후 각계층별로 1 화소 간격의 이동벡터 탐색을 수행하므로 탐색 간격의 조정이 필요없고, 후보가 증가할 경우 각 계층의 후보를 기억할 필요는 있다.

Ⅶ. 실험 및 결과

먼저 제안한 기법의 성능을 기존의 방법과 비교하고, 이 기법을 MPEG에 적용하여 성능을 알아본다. 실험에는 그림 9에 나타낸 720 x 480 크기의 6가지 CCIR 601 표준영상을 이용한다. 즉, "Table Tennis"와 "Susie" 영상은 각각 29장, "Flower Garden" 영상은 31장, "Football"과 "Popple" 영상은 각각 150장, 그리고 "Mobile" 영상은 136장을 사용하는데, 이것은 720 x 240 크기의 luminance 신호의 odd field만을 사용하여 보간 [11] 한 영상이다. "Mobile"과 "Flower Garden" 영상은 움직이고 있는 물체 위에서 배경을 관찰한 것으로, 영상전체의 움직임이 거의 같은 방향이고, "Popple" 영상은 회전과 확대의 복잡한 움직임을, "Football" 영상은 움직임이 매우 큰 영상으로 blurring이 많고, "Susie" 영상은 움직임이 거의 없는 영상이다. 성능의 객관적인 평가를 위해서 식 (8)로 정의되는 PSNR (peak signal to noise ratio)을 사용한다. 또한 제안한 기법이 TSS에 근거한 변위방식을 갖고, H.261에서 TSS를 권고하며, MPEG에서는 FS를 이동벡터의 탐색에 권고하고 있으므로 제안된 기법의 성능 비교를 위해 이 두가지 BMA를 함께 실험한다.

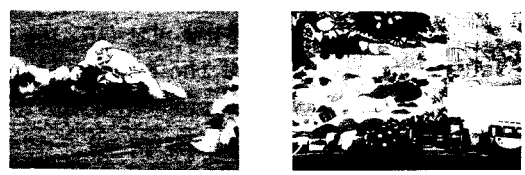
$$PSNR = -10\log_{10}\frac{\sum_{m}\sum_{n}\left[g(m,n)-g_r(m,n)\right]^2 / KL}{255^2} \qquad (8)$$

여기서 K L 은 영상의 크기, g(m, n)은 (m, n) 위치의 원영상 화소값, g_r(m, n)은 복원영상의 화소값이다.



(a) "Table Tennis" 영상 (b) "Flower Garden" 영상

(c) "Susie" 영상 (d) "Popple" 영상



(e) "Football" 영상 (f) "Mobile" 영상

그림 9. 실험 원영상
Fig. 9. Original test images.

일반적으로 작은 탐색 블럭을 사용할 경우 예측오차의 성능은 큰 탐색블럭을 사용할 때보다 향상되나, 이동벡터 전송시 overhead가 많아지는 단점이 있다. 대개는 8 x 8이나 16 x 16블럭을 사용하므로, 본 실험에서도 이 두가지 탐색블럭에 대한 실험을 함께 고려하며, 탐색 변위는 최대 ±7로 제한하며, forward search만을 고려한다. 또한 subpel 단위까지의 변위 탐색은 각 블럭크기에 대해 동일하게 적용하여 이때의 PSNR 성능 개선을 함께 살펴보며, MPEG SM3와 제안한 기법을 결합하여 그 성능을 분석하였다.

1. 8 x 8 블럭의 실험결과

각 영상에 대한 평균 PSNR을 표 3에 종합해 놓았다. 이때 제안 n은 후보의 개수를 n으로 제한했을 때의 제안한 기법을 말한다. 표 3에서 보듯이 제안 1은 TSS와 비교할 때 영상에 따라 -0.5 ~+0.5 dB의 차이를 보이므로 TSS와 근사한 성능을 가지며, 계산량은 앞서 서술한 바와 같이 TSS의 절반 수준이다. 영상에 따라서는 TSS보다 우수한 성능을 보이기도 하는데, 이러한 현상을 보이는 "Mobile" 영상은 전체 영상이 규칙적인 이동을 갖는 영상이며 "Popple" 영상은 물체들이 회전하는 영상이다.

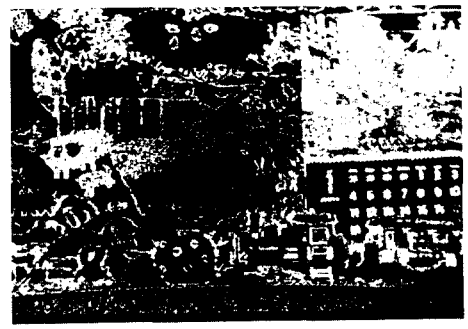
표 3. 제안한 기법의 평균 PSNR (8 x 8)
Table 3. Average PSNR of the proposed method (8 x 8).

(단위: dB)

방 법	Table T.	Flower G.	Susie	Popple	Football	Mobile
TSS	26.91	26.45	37.71	26.86	25.73	23.35
제안 1	26.57	25.90	37.61	27.10	25.53	24.01
제안 2	26.96	27.21	38.33	27.91	26.16	25.65
제안 3	27.17	27.66	38.53	28.26	26.42	26.21
제안 4	27.23	27.85	38.63	28.53	26.56	26.48
제안 5	27.26	27.92	38.64	28.63	26.63	26.60
제안 6	27.28	27.99	38.72	28.70	26.68	26.66
제안 7	27.30	28.04	38.74	28.82	26.72	26.70
제안 8	27.30	28.06	38.77	28.85	26.75	26.73
제안 9	27.31	28.08	38.79	28.87	26.77	26.75
FS	27.39	28.13	38.94	29.12	26.83	27.04

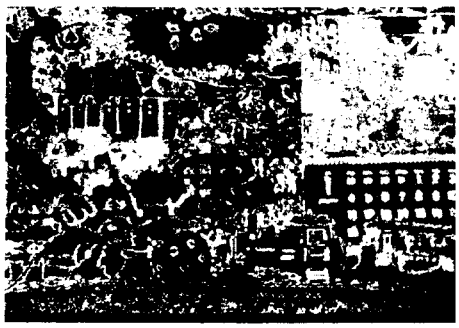
그림 10은 복원영상과 원영상과의 차이를 8배한 차영상으로 TSS의 경우 제대로 복원되지 못한 지점에서도 밝기차는 다른 방법과 큰 차이를 보이지 않으나 실제 복원 영상은 글자가 깨어진 것으로 보아 local minima의 결과임을 알 수 있다. 반면에 제안한 기법은 피라미드 영상이 구성된 후 저주파 성분만이 남은 상위계층에서의 coarse한 움직임 추적이 전역적인 이동을 더 잘 검출해냈으며, 이로 인해 local minima를 더 잘 극복한 것을 알 수 있으며, 탐색 후보의 개수를 증가시킴에 따라 이러한 local minima의 영향을 잘 제거하여 세밀한 움직임을 잘 찾는 것을 알 수 있다.

(a) FS



(b) TSS

(c) 제안한 방법 (후보갯수 = 1)

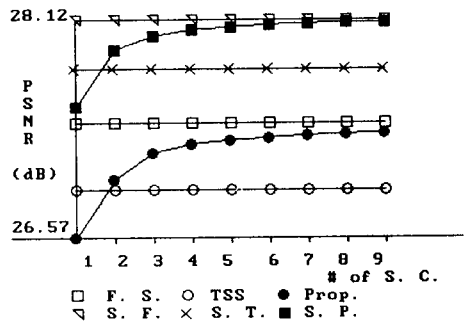


(d) 제안한 방법 (후보갯수 = 9)

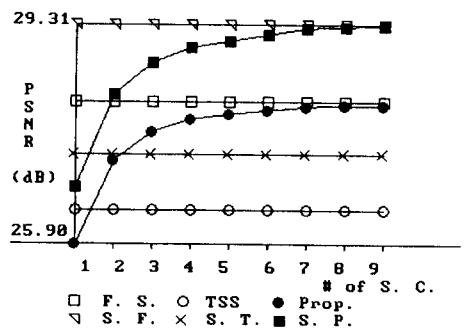
그림 10. 복원영상과 원영상과의 차영상 ("Mobile" 영상)

Fig. 10. Difference image between the original and reconstructed images ("Mobile" image).

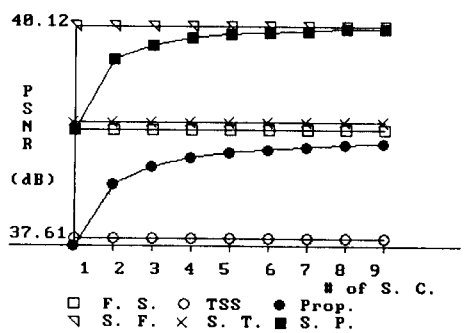
탐색 후보의 개수가 증가할수록 TSS의 수준에서 FS의 성능으로 다가감을 알 수 있는데 이는 FS와 TSS사이의 차이가 큰 영상일수록 두드러지는 것을 볼 수 있다. 그림 11은 제안한 기법에서 후보의 개수를 증가시킴에 따른 추이를 보이기 위한 그래프이다. 그래프상의 약자인 F. S.은 FS를 나타내고 Prop.는 제안한 기법을 의미한다. 또한 S. F., S. T., S. P.은 각각 subpel단위의 탐색을 함께 고려했을 경우의 FS, TSS, 제안한 기법을 의미하며, 가로 축의 # of S. C.은 제안된 기법에서의 후보의 개수를 의미한다. FS와 TSS 방법은 후보의 개수와 무관하므로 후보개수에 따른 그래프에서 일정한 수평 성분으로 표현된다. 그래프에서 보듯이 후보의 개수가 증가함에 따라 제안한 기법의 성능이 TSS 수준에서 FS 수준으로 지수함수적으로 증가하는 것을 볼 수 있으며, 후보의 최대 개수인 9개까지 증가시켰을 때는 모든 영상에서 거의 FS에 가까운 PSNR을 갖는 것을 알 수 있으며, 이때의 계산량은 후보를 하나로 했을 때의 약 9배에 가까운 계산량을 가지므로, FS에 비해서 절반 정도의 계산량이라고 할 수 있다. 탐색 후보의 개수를 적당히 증가시킬 경우에 TSS에 비해 더욱 향상된 예측오차 성능과 함께 FS에 비해 커다란 계산량의 감소를 얻을 수 있음을 알 수 있는데, 후보의 개수를 2개로 제한하였을 경우에 모든 실험 영상에서 TSS와 FS의 중간정도의 PSNR을 가지며, 이때의 계산량은 후보가 하나일 때의 2배에 조금 못 미치는 계산량을 갖게 되므로 TSS와 거의 동일한 계산량에서 매우 향상된 예측오차 성능을 갖는 이동벡터의 검출을 이루는 장점을 갖는다.



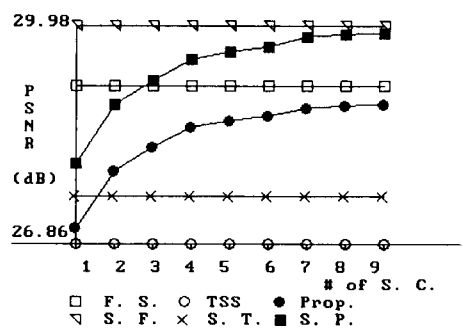
(a) "Table Tennis" 영상



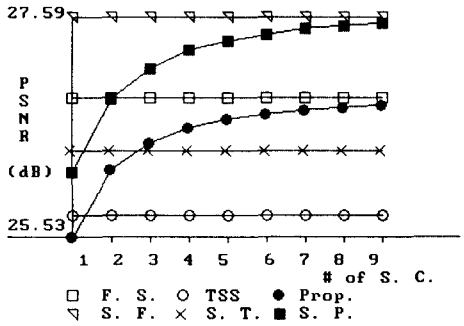
(b) "Flower Garden" 영상



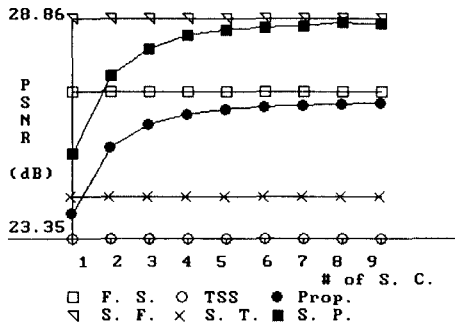
(c) "Susie" 영상



(d) "Popple" 영상



(e) "Football" 영상



(f) "Mobile" 영상

그림 11. 제안한 기법의 후보벡터 증가에 따른 평균 PSNR 그래프 (8 x 8)

Fig. 11. Graphs of the average PSNR of the proposed method as a function of the number of candidate motion vectors (8 x 8).

제안한 기법에 subpel 단위까지의 이동벡터를 탐색했을 때의 결과는 표 4에 나타나 있다.

표 4. 제안한 기법의 subpel 단위 탐색의 평균 PSNR (8 x 8)

Table 4. Average PSNR of the proposed method with subpel search (8 x 8).

(단위: dB)

방 법	Table T.	Flower G.	Susie	Popple	Football	Mobile
TSS	27.71	27.30	39.03	27.55	26.34	24.40
제안 1	27.50	26.80	38.95	28.02	26.14	25.50
제안 2	27.90	28.25	39.75	28.86	26.83	27.45
제안 3	28.00	28.74	39.92	29.20	27.11	28.12
제안 4	28.05	28.97	40.01	29.51	27.29	28.46
제안 5	28.07	29.07	40.04	29.62	27.37	28.59
제안 6	28.08	29.17	40.06	29.69	27.43	28.65
제안 7	28.09	29.26	40.08	29.83	27.49	28.70
제안 8	28.10	29.29	40.10	29.86	27.52	28.73
제안 9	28.10	29.31	40.10	29.88	27.54	28.74
FS	28.12	29.31	40.12	29.98	27.59	28.86

제안한 기법의 선정된 후보의 개수에 무관하게 최하위 단에서 얻어진 이동벡터를 중심으로 half-pel단위의 영상을 구성하여 이동벡터를 찾는 기법으로 이때는 TSS나 FS와 동일한 계산량의 증가를 보인다. 표 3과 비교하면 subpel 탐색을 하지 않았을 때와 비교하여 평균적으로 1 dB가량의 성능개선을 볼 수 있으며, 후보의 개수를 증가시킴에 따라 지수함수적으로 성능이 향상되어 FS의 PSNR에 다가가는 것을 볼 수 있다. 후보의 개수가 9일 때는 FS에 0.001~0.1 dB정도의 차이로 성능이 근접하는 것을 알 수 있다.

2. 16 x 16 블럭의 실험 결과

탐색 부블럭을 16 x 16으로 하였을 때의 PSNR의 평균치를 표 5에 나타냈다. 이 때의 성능 역시 표 3의 결과와 같은 추이를 갖는 것을 알 수 있다. 후보의 개수를 2개로 했을 경우 TSS와 FS의 중간 정도의 PSNR을 가지면서, 계산량은 TSS와 비슷하고, 후보의 개수를 3개로 제한했을 경우에는 TSS와 FS의 차이의 3/4정도를 극복하는 것을 알 수 있으며, 후보의 개수를 최대로 했을 경우에는 FS에 비해 계산량은 절반 수준이면서 PSNR은 거의 FS 수준임을 알 수 있다.

표 5. 제안한 기법의 평균 PSNR (16 x 16)

Table 5. Average PSNR of the proposed method (16 x 16).

(단위: dB)

방 법	Table T.	Flower G.	Susie	Popple	Football	Mobile
TSS	26.29	26.13	37.41	26.03	24.22	23.89
제안 1	26.42	26.25	37.56	26.46	24.26	25.05
제안 2	26.56	26.63	37.87	26.92	24.49	25.50
제안 3	26.61	26.78	37.95	27.07	24.57	25.59
제안 4	26.62	26.82	37.97	27.20	24.61	25.63
제안 5	26.63	26.83	38.01	27.24	24.63	25.64
제안 6	26.64	26.84	38.03	27.27	24.65	25.65
제안 7	26.64	26.85	38.04	27.31	24.66	25.65
제안 8	26.64	26.85	38.05	27.32	24.67	25.66
제안 9	26.64	26.85	38.05	27.33	24.67	25.67
FS	26.69	26.87	38.16	27.48	24.70	25.73

16 x 16 블럭을 사용한 경우 대부분의 영상에서 후보의 개수를 하나로 한 경우의 성능이 TSS보다 우수함을 알 수 있는데, 이것은 8 x 8 부블럭을 사용한 경우와 비교할 경우 최상위단의 블럭이 16개의 화소로 이루어져 계산된 MAD^2 값이 같게 되는 경우가 줄어들게 되므로 8 x 8 블럭을 사용한 경우보다 더욱 정확한 전역적인 이동벡터를 구하기 때문으로 풀이된

다. 전체적인 성능은 8 x 8 블럭을 사용한 때보다 2dB에서 0.5dB정도의 성능저하를 보이고 있는데, 영상내의 모든 영역에 움직임이 고루 퍼져 있을 때가 성능저하가 심하고, 움직임이 적은 영상은 성능저하의 정도가 덜한 것을 알 수 있다. 이것은 탐색 블럭이 커짐에 따라 세밀한 움직임을 무시하게 되는 블럭이 증가함에 따른 전체적인 성능저하의 결과로 앞서 설명한 바와 같이 이동벡터 전송 데이타의 양과 trade off관계에 있다.

표 6은 16 x 16의 탐색블럭을 이용했을 때, subpel단위까지 이동벡터를 찾은 결과를 보이고 있다. 전체적인 성능 및 이에 따른 추세는 8 x 8 블럭을 사용한 경우와 같다.

표 6. 제안한 기법의 subpel 단위 탐색의 평균 PSNR (16 x 16)

Table 6. Average PSNR of the proposed method with subpel search (16 x 16).

(단위: dB)

방 법	Table T.	Flower G.	Susie	Popple	Football	Mobile
TSS	27.08	27.07	38.97	26.90	24.76	25.22
제안 1	27.35	27.21	39.13	27.47	24.82	26.72
제안 2	27.44	27.60	39.40	27.89	25.06	27.14
제안 3	27.47	27.76	39.43	28.03	25.14	27.23
제안 4	27.47	27.81	39.44	28.18	25.20	27.27
제안 5	27.48	27.82	39.45	28.22	25.22	27.28
제안 6	27.48	27.84	39.46	28.26	25.24	27.29
제안 7	27.48	27.85	39.47	28.30	25.25	27.29
제안 8	27.48	27.85	39.47	28.31	25.26	27.30
제안 9	27.48	27.86	39.47	28.32	25.26	27.30
FS	27.50	27.88	39.50	28.38	25.28	27.32

후보의 개수를 증가시키는 기법은 TSS에 적용시켜 TSS의 성능을 개선시킬 수도 있다. 예를 들어 TSS의 처음 단계에서 가장 작은 MAD값을 갖는 지점을 여러 개로 선택해서 다음 단계로 전달하여 각각을 독립적으로 진행시킬 경우가 여기에 해당하는데, 이러한 후보의 개수를 2개로 사용할 경우, "Table Tennis" 영상은 26.45 dB, "Flower Garden" 영상은 26.49 dB, "Susie" 영상은 37.80 dB, "Popple" 영상은 26.61 dB, "Football" 영상은 24.45 dB, 그리고 "Mobile" 영상은 24.80 dB의 성능을 갖는다. 이를 표 5와 비교하면 기존의 TSS보다 성능이 개선되는 것을 알 수 있다. 그러나 피라미드를 이용한 제안한 기법 (후보의 개수가 2인 표의 제안 2)보다 성능이 떨어지는 것을 알 수 있는데 이는 피라미드를 이용하는 기법이 후보를 선정하는데 있어서 보다 전역적인 특성을 갖는 후보를 선정하기 때문으로 풀이된다.

3. 제안한 기법의 MPEG [11-13] 적용 실험

제안한 기법중 후보의 개수를 하나로 제한한 경우의 half-pel까지의 움직임을 검출하는 계층적 움직임 추정기법의 성능을 분석하기 위해 MPEG SM3와 제안한 기법을 결합시켜 성능분석을 한다. 실험에 사용한 영상은 CCIR 601 format (luminance resolution: 720 x 480, chrominance resolution: 360 x 240)으로부터 부표본화 필터를 사용하여 MPEG SM3의 SIF (source input format, luminance resolution: 352 x 240, chrominance resolution: 176 x 120)로 변환한 4종류의 영상을 사용하였다.

이 4종류의 영상은 각각 145 frame의 "Edited" 영상과 "Football" 영상 그리고 "Popple" 영상과 133 frame의 "Mobile" 영상으로, head-and-shoulder 실험영상에 비해 상대적으로 움직임이 크고 복잡하다. "Edited" 영상은 5종류의 서로 다른 영상이 처음 4가지 영상은 30장 그리고 마지막 영상은 25장이 연결된 실험영상으로 매 30번째 영상마다 scene change가 발생한다.

실험에서 하나의 GOF (group of frames)는 한 장의 intraframe과 3장의 predicted frame 그리고 8장의 interpolated frame의 총 12장의 frame들로 구성되어 있다고 가정하였으며, 움직임 추정시 half-pel까지 움직임을 추정하였고, 고정 전송채널의 전송율은 1.15 Mbps이고 전송 버퍼의 용량은 120 Kbits로 하였다. 또한 초기의 버퍼상태는 30 Kbits로 가정하였다.

표 7. MPEG SM3와 제안된 기법과 결합된 MPEG SM3의 평균 PSNR

Table 7. Average PSNR of the MPEG SM3 and the MPEG SM3 combined with the proposed method.

(a) intra mode Y

	FS	TSS	제안 1	제안 2	제안 5	제안 9
Edited	28.44	28.36	28.07	28.34	28.38	28.42
Football	26.83	26.69	26.71	26.75	26.80	26.82
Popple	31.13	30.86	30.77	31.09	31.15	31.10
Mobile	21.64	21.58	21.51	21.57	21.59	21.61

(b) intra mode I

	FS	TSS	제안 1	제안 2	제안 5	제안 9
Edited	34.24	34.18	33.93	34.14	34.20	34.21
Football	31.04	30.94	30.96	30.98	31.03	31.03
Popple	30.00	29.62	29.57	29.92	30.03	29.94
Mobile	25.07	25.03	24.98	25.02	25.04	25.05

(c) intra mode Q

	FS	TSS	제안 1	제안 2	제안 5	제안 9
Edited	34.23	34.16	33.95	34.13	34.21	34.21
Football	34.28	34.20	34.22	34.24	34.27	34.28
Popple	31.17	30.78	30.76	31.08	31.19	31.10
Mobile	24.43	24.40	24.36	24.40	24.41	24.42

(d) predicted mode Y

	FS	TSS	제안 1	제안 2	제안 5	제안 9
Edited	26.86	26.60	26.75	26.79	26.84	26.85
Football	23.47	23.43	23.42	23.48	23.45	23.46
Popple	27.25	26.99	27.02	27.18	27.22	27.24
Mobile	20.82	20.69	20.71	20.78	20.79	20.78

(e) predicted mode I

	FS	TSS	제안 1	제안 2	제안 5	제안 9
Edited	33.44	33.45	33.43	33.45	33.45	33.43
Football	29.11	29.14	29.12	29.16	29.11	29.11
Popple	26.79	26.78	26.82	26.81	26.72	26.76
Mobile	25.64	25.66	25.61	25.66	25.68	25.65

(f) predicted mode Q

	FS	TSS	제안 1	제안 2	제안 5	제안 9
Edited	33.50	33.62	33.53	33.51	33.54	33.53
Football	33.39	33.35	33.40	33.43	33.38	33.40
Popple	27.96	28.10	28.05	27.99	27.96	27.96
Mobile	24.98	24.92	24.93	24.97	25.00	24.96

(g) bidirectional mode Y

	FS	TSS	제안 1	제안 2	제안 5	제안 9
Edited	26.38	26.09	26.13	26.26	26.32	26.33
Football	22.99	22.89	22.86	23.00	22.95	22.99
Popple	26.48	26.17	26.23	26.39	26.40	26.46
Mobile	20.55	20.44	20.41	20.50	20.53	20.53

(h) bidirectional mode I

	FS	TSS	제안 1	제안 2	제안 5	제안 9
Edited	32.33	32.34	32.24	32.24	32.30	32.28
Football	28.54	28.54	28.52	28.58	28.61	28.61
Popple	25.64	25.50	25.66	25.69	25.67	25.64
Mobile	26.38	26.15	26.19	26.38	26.33	26.47

(i) bidirectional mode Q

	FS	TSS	제안 1	제안 2	제안 5	제안 9
Edited	32.57	32.64	32.46	32.50	32.59	32.60
Football	32.75	32.69	32.71	32.79	32.83	32.82
Popple	26.81	26.78	26.88	26.85	26.82	26.79
Mobile	25.43	25.20	25.31	25.48	25.41	25.53

MPEG SM3의 조건하에서 성능을 비교한 실험결과는 표 7에 나타나 있다. 표에서 제안 n은 제안한 기법의 후보의 개수를 n으로 했을 때의 결과이다. 표 7은 MPEG SM3의 조건하에서 이동벡터의 탐색에 FS를 사용한 결과와 TSS를 사용한 결과, 그리고 제안한 기법을 MPEG SM3에 결합한 결과의 평균 PSNR을 나타낸 것이다. 표에서 보듯이 TSS를 사용한 결과는 FS를 사용한 결과보다 성능이 떨어짐을 알 수 있다. 반면에 제안한 기법을 MPEG SM3에 결합한 결과는 후보의 수가 증가함에 따라 FS를 사용한 결과에 근접함을 알 수 있다.

또한 표 7에서 보듯이 FS를 사용한 결과와 비교할 때, 움직임 추정을 위한 계산량은 상당히 감소되면서 거의 성능차이가 없음을 표를 통해서 알 수 있다.

Ⅷ. 결론

본 연구에서는 TSS의 삼단계 탐색기법의 변위방식을 유지할 수 있는 피라미드 형태의 계층적 구조를 이용한 이동벡터 추정기법을 제안하였다. 각 계층의 영상은 하위계층의 2 x 2 크기의 블럭을 한 화소로 만드는 단순평균법을 이용하여 1/4로 줄어든 영상이고, 상위 계층에서부터 단계 탐색을 행하기 때문에 계산량을 많이 줄일 수 있는 알고리듬이며, 상위계층에는 저주파성분만 남으므로 전역적인 움직임을 찾는데 유리하고, 잡음에 의한 local minima에 빠지는 문제를 해결할 수 있는 장점이 있다.

TSS나 제안한 기법이 FS의 PSNR 성능에 못미치는 것은 local minima의 문제이므로 이 local minima의 문제를 해결하기 위해 각 단에서 후보벡터의 수를 증가시킴으로써 local minima에 빠질 가능성을 줄이는 동시에 상위계층에서의 탐색을 위한 계산량이, 탐색개수가 증가하더라도 큰 overhead가 되지 않는 점을 이용해 적은 계산량으로도 PSNR 성능면에서 FS에 근접하는 결과를 얻었다.

제안된 기법은 후보 벡터의 개수를 2개로 할 경우 TSS와 같은 계산량으로도 TSS와 FS의 중간정도의 PSNR 성능을 갖고, 후보벡터의 개수가 9개로 선정될 경우에는 FS의 절반 정도의 계산량으로 FS에 필

적할 만한 PSNR 성능을 보이는 것을 알 수 있다. 또한 MPEG SM3와 결합한 경우는 이동벡터 추정의 계산량을 크게 줄이면서 PSNR 성능은 거의 차이가 없음을 알 수 있다. 더우기 이 알고리듬은 반복적인 탐색기법을 사용하므로 하드웨어의 구현 또한 여타 고속 블럭 정합 알고리듬에 비해 용이하다. 따라서 적은 계산량으로 좋은 예측오차를 얻을 수 있는 이동벡터 검출기법으로서 기존의 방법들의 단점을 개선시킨 면에서 기존의 고속 BMA나 많은 계산량을 요구하는 FS와 대체되어 사용이 가능하리라 기대된다.

또한 변위가 큰 움직임이 나타났을 때를 위해 subpel기법 대신 탐색변위를 증가시키는 방법들이 사용되는데, 제한된 이동변위를 넘는 큰 움직임에 대한 성능의 개선을 위해 계층의 수를 증가시켰을 때의 성능분석과 성능개선이 앞으로의 연구과제가 될 것이다.

參 考 文 獻

[1] A. K. Jain, "Image data compression: A review," *Proc. IEEE*, vol. 69, pp. 349-389, Mar. 1981.

[2] A. N. Netravali and J. O. Limb, "Picture coding: A review," *Proc. IEEE*, vol. 68, pp. 366-406, Mar. 1980.

[3] N. S. Jayant and P. Noll, Digital Coding of Waveforms. Prentice-Hall, New Jersey, pp. 252-338, 1984.

[4] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans.* Commun., vol. COM-29, pp. 1799-1808, Dec. 1981.

[5] T. Koga et al., "Motion-compensated interframe coding for video conferencing," *in Proc. Nat. Telecom. Conf.*, pp. G 5.3.1-G 5.3.5, Nov. 29-Dec. 3, 1981.

[6] M. Bierling, "Displacement estimation by hierarchical blockmatching," in Proc. *SPIE Conf.* Visual Communications and Image Processing '88, vol. 1001, pp. 942-951, Cambridge, Mass., Nov. 1988.

[7] 천 강욱, 나 종범, "오차함수의 계층적 근사를 이용한 고속 블럭 정합 알고리즘 에 의한 움직임 벡터의 추정," 제 4회 신호처리합동학술대회논문집, 제 4권 제 1호, pp. 5-10, 1991년 9월.

[8] 서 강수, 김 재균, "인접 이동벡터간의 기하학적 연관성을 이용한 계층적 구조를 갖는 이동벡터 추정 기법," 제 4회 신호처리합동학술대회논문집, 제 4권 제 1호, pp. 11-14, 1991년 9월.

[9] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, pp. 532-540, Apr. 1983.

[10] 김 재균 외, 영상정보의 부호화기술 개발, 과학기술처, 1988.

[11] ISO CD 11172-2 rev 1 "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s," Nov. 1991.

[12] ISO/IEC JTC1/SC2/WG11 "MPEG video simulation model three," MPEG90/041, July 1990.

[13] CCITT SGXV Working Party XV/4, Description of Ref. Model 7 (RM7), CCITT SGXV Doc. #446, 1988.

著 者 紹 介

南 權 文(準會員) 第 29卷 B編 第 1號 參照
현재 삼성전자 연구원

朴 來 弘(正會員) 第 23卷 第 6號 參照
현재 서강대학교 전자공학과 부교수

金 俊 植(正會員) 第 29卷 B編 第 4號 參照
현재 서강대학교 전자공학과 박사과정

沈 英 錫(正會員) 第 30卷 B編 第 4號 參照
현재 생산기술연구원 근무