

論文2000-37SP-2-3

움직임 벡터의 통계적 특성을 이용한 고속 움직임 추정

(Fast Motion Estimation Using the Statistical Characteristics of Motion Vector)

崔正鉉*, 朴大圭*, 李京桓*, 李法基*, 金德奎*

(Jung Hyun Choi, Dae Gyue Park, Kyeong Hwan Lee, Bub Ki Lee,
and Duk Gyoo Kim)

요약

고속 움직임 추정 방법들에서는, 움직임 추정 오차는 최적 탐색점에서 멀어질수록 단조 증가한다는 성질을 이용하여, 계산량을 줄인다.^{[2] [6]} 본 논문에서는 먼저, 참조 탐색점들의 MAE (mean absolute error) 차가 클 때에, 대부분의 움직임 벡터는 작은 MAE의 참조 탐색점 방향에서 나타난다는 통계적 특성을 조사하였다. 그러므로, 이 특성을 이용하여 탐색점 수를 줄일 수 있는 고속 움직임 추정 방법을 제안한다. 모의 실험 결과, 기존의 고속 움직임 추정 방법들에 비해서는 비슷한 성능을 유지하면서 계산량을 줄일 수 있었다.

Abstract

In Fast motion estimation algorithms, they reduce the computational complexity using the assumption that the matching error increases monotonically as the search moves away from the global minimum error^{[2] [6]}. In this paper, we first investigate the statistical characteristics of motion vector that the motion vector mostly occurs on the side of small MAE (mean absolute error) between the reference search points when the MAE difference of them is large. Therefore, we propose a fast motion estimation algorithm using this property and can reduce the number of search points. The computer simulation result shows that the proposed method reduces computational complexity compared with conventional fast algorithms.

I. 서론

디지털 전송 시스템에서는 정보를 제한된 대역의 채널을 통하여 전송하거나 저장 매체에 저장하기 위해서 데이터를 압축한다. HDTV, 디지털 TV 및 화상회의 등의 분야에서는, 많은 양의 영상 데이터를 압축하기

위해서 동영상 내에 존재하는 여러 가지 중복성들 (redundancy), 즉 프레임 내 (intraframe)의 화소들 사이에 존재하는 공간적 중복성, 인접 프레임 간 (interframe)에 존재하는 시간적 중복성, 그리고 발생부호의 확률적 중복성 등을 제거한다. 동영상 데이터에서는 시간 상관 계수가 공간 상관 계수보다 훨씬 크기 때문에 인접 프레임간의 시간적 상관성을 이용하는 움직임 보상 부호화가 효율적이다.^[1]

동영상 부호화에서의 움직임 추정 방법으로는, 알고리듬이 간단하고 보상 영상이 우수한 블록 정합 방법 (block matching algorithm; BMA)^[8] 널리 사용된다. 전역 탐색 블록 정합 방법 (full search BMA; FSBMA)

* 正會員, 慶北大學校 電子電氣工學科

(School of Electronic and Electrical Engineering,
Kyungpook National University)

接受日字 : 1999年9月15日, 수정완료일 : 2000年2月26日

은 탐색 영역내의 모든 탐색점들에 대해 정합을 행하는 방법으로서, 움직임 벡터를 정확하게 찾을 수는 있지만 많은 계산량을 필요로 한다. 따라서 움직임 추정의 계산량을 감소시키기 위해 2 차원 로그 탐색 (two dimensional logarithmic search) 방법, 교차 탐색 (cross search) 방법, 쌍방향 탐색 (conjugate search) 방법, 3 단계 탐색 (three-step search; TSS) 방법 등의 여러 고속 블록 정합 알고리듬들이 제안되었다.^{[2]-[5]}

TSS 방법^[3]에서는, 움직임 추정 오차는 움직임이 일어나는 방향으로 단조 감소한다고 가정하고, 전체 탐색 영역에서 참조 탐색점 (reference search point)을 정하여, 단계를 진행할수록 참조 탐색점의 간격을 좁혀가면서 움직임을 추정한다. Po 등은 움직임 벡터의 분포 특성을 이용한 4단계 탐색 (four-step search; FSS) 방법^[6]을 제안하였다. 이 방법에서는, 동영상의 움직임 벡터가 탐색 영역의 중심 부분에 많이 분포하므로, 초기 단계에서 탐색창의 크기를 줄여서 탐색함으로써, 성능을 향상시켰다.

본 논문에서는, 움직임 벡터의 통계적 특성을 조사하고, 이를 이용한 새로운 고속 움직임 추정 방법을 제안한다. 점진적 움직임 추정 방법의 마지막 단계에서 세부 탐색을 행할 때, 최소 오차 참조 탐색점 (minimum error reference search point) 근방의 참조 탐색점들에 서의 정합 오차 크기에 따라, 세부 탐색 위치를 예측할 수 있다. 따라서 본 연구에서는, 인접 참조 탐색점들의 정합 오차 (matching error)의 크기에 따른 움직임 벡터의 분포 특성을 조사하였다. 그 결과, 이전 단계에서의 최소 오차 참조 탐색점 근방의 인접 참조 탐색점들의 정합 오차의 차가 클 경우, 마지막 단계에서 최소 오차 탐색점이 정합 오차가 작은 인접 참조 탐색점 근방에서 일어날 확률이, 거의 “1”로 나타났다. 또한 이전 단계에서 인접 참조 탐색점들의 정합 오차의 차가 아주 작을 경우에는, 그 방향에 상관없이 정합 오차의 차가 아주 작게 나타났다. 그러므로 이와 같은 움직임 벡터의 통계적 특성으로부터 문턱값 (threshold)을 정하여, 세 가지 탐색 패턴으로 분류하여 세부 탐색을 행함으로써, 기존 방법들에 비해 성능을 향상시킬 수 있었다.

본 논문의 구성은, II장에서 움직임 벡터의 통계적 특성을 보이며, III장에서 이를 이용한 제안한 알고리듬을 설명하고, IV장에서는 컴퓨터 모의 실험 결과를 나타내고, 그리고 V장에서 결론을 맺는다.

II. 움직임 벡터의 통계적 특성

점진적 움직임 추정 방법들에서는, 움직임 추정 오차는 최적 탐색점에서 멀어질수록 단조 증가하는 가정을 이용하여, 탐색 초기 단계에서는 참조 탐색점의 간격을 넓게 하고, 단계가 진행될수록 참조 탐색점의 간격을 점점 줄여서 탐색을 행함으로써 계산량을 줄일 수 있다. 마지막 단계에서는 이전 단계까지의 최소 정합 오차 탐색점을 중심으로 모든 8 개의 탐색점들을 세부 탐색하여, 이 중 최소 오차의 탐색점을 움직임 벡터로 결정한다.

점진적 움직임 추정 방법의 마지막 단계에서 세부 탐색을 행할 때, 최소 오차 참조 탐색점을 중심으로 한 인접 참조 탐색점들에서의 정합 오차 크기에 따라, 세부 탐색 위치를 예측할 수 있다.

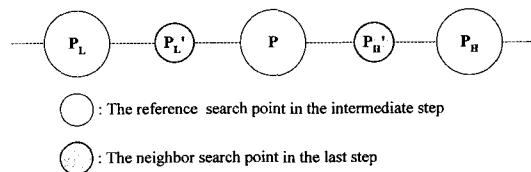


그림 1. 움직임 벡터의 확률 분포를 구하기 위한 일차 원 탐색

Fig. 1. The 1-D search to find the probability distribution of motion vector.

그림 1에서와 같이, 이전 단계에서의 최소 오차 탐색점을 P 로 두고, 이로부터 탐색 범위 (search distance) 가 2 만큼 떨어진 좌우 참조 탐색점들 중 정합 오차가 작은 쪽을 P_L , 큰 쪽을 P_H 라고 할 때, 마지막 단계에서 세부 탐색해야 할 두개의 탐색점 (P_L' , P_H')의 정합 오차를 비교하면, P_L' 의 정합 오차가 P_H' 보다 작을 것으로 예측할 수 있다. 그러므로, 세부 탐색을 행할 때에, 주의의 모든 8 점에 대해서 탐색하지 않고, 적절한 탐색점들에 대해서만 탐색함으로써, 움직임 추정 계산량을 감소할 수 있다.

따라서, 본 논문에서는 먼저 움직임 벡터의 특성이 제안한 가정에 잘 맞는지를 조사하였다. 그림 1의 예를 2 차원으로 확장하여, 마지막 탐색 단계 이전까지의 최소 오차 참조 탐색점과 마지막 단계에서의 인접 탐색점들에 대한 정합 오차의 분포를 조사하였다. 인접 8점

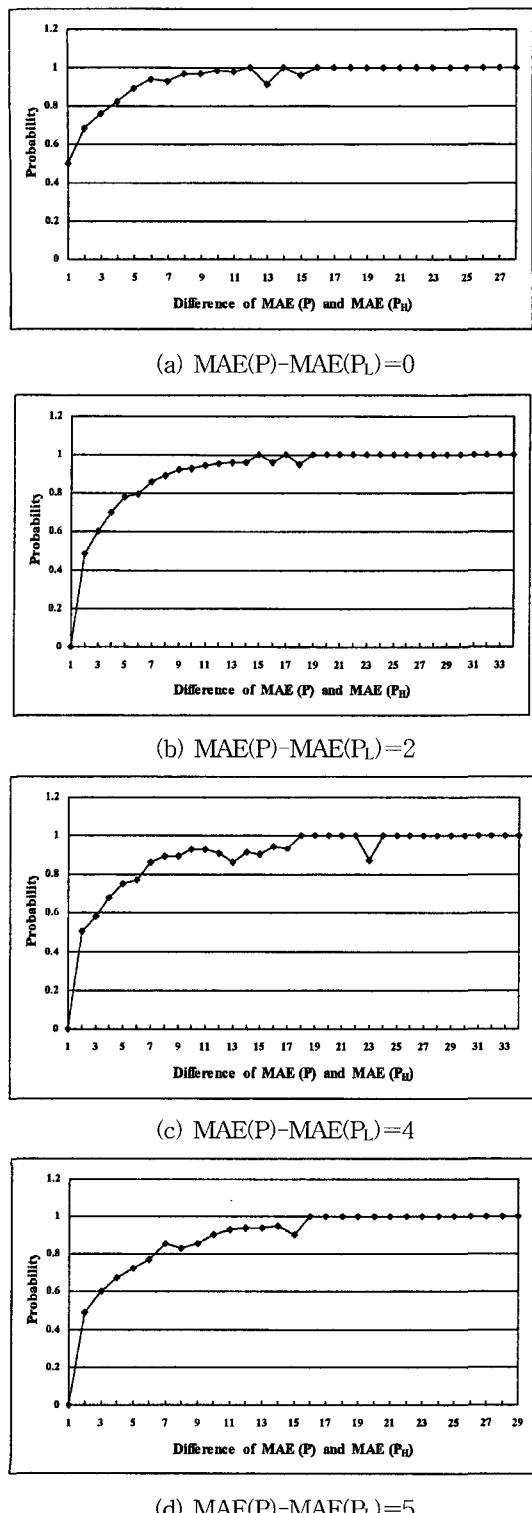


그림 2. FOOTBALL 영상에 대한 움직임 벡터의 확률 분포

Fig. 2. The probability distribution of motion vector to FOOTBALL sequence.

의 탐색점들 중 위의 가정에 맞는 탐색점의 개수를 조사하였으며, 실험 영상으로는 FOOTBALL, TABLE TENNIS, FLOWER GARDEN 및 MOBILE 영상 등의 움직임이 빠른 영상들과 움직임이 작은 SALESMAN 영상 각 100 프레임씩을 사용하였으며, 그 결과는 모두 비슷하게 나타났다.

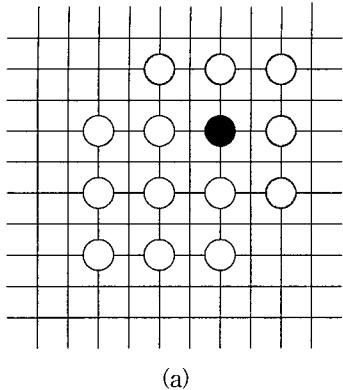
그림 2에서는 FOOTBALL 영상에 대한 정합 오차의 통계적 확률 분포를 나타내었는데, 가로축은 P 와 P_H 의 MAE (mean absolute error)의 차를 나타내며, 세로축은 제안한 가정을 만족하는 확률을 나타내었다. 예를 들면, 그림 2(d)는 P 와 P_H 의 블록 MAE 차가 5일 경우이며, 따라서 가로축의 값이 3인 것은 P 와 P_H 의 블록 MAE의 차이가 3인 것을 나타낸다. P 와 P_H 의 MAE 차 만큼 그래프가 그려질 수 있으며, 각 그래프마다 P 와 P_H 의 MAE 차만큼의 가로축 눈금이 생기게 된다. 그림 2에서 확인할 수 있듯이, P 와 P_H 의 MAE 차가 클수록 제안한 가정이 맞을 확률이 커짐을 알 수 있다. 또한 P 와 P_H 의 MAE 차가 작은 경우는, 확률이 다소 낮더라도 전체적인 MAE의 차가 작으므로 성능에는 큰 영향을 미치지 않는다.

III. 제안한 고속 움직임 추정 방법

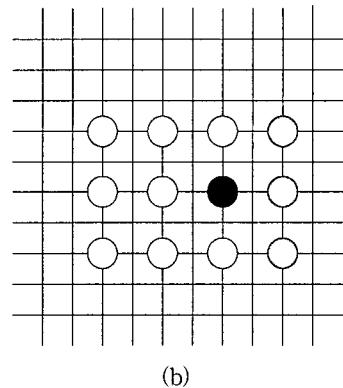
본 논문에서는, II장에서의 조사한 움직임 벡터의 통계적 특성을 이용하여, 고속 움직임 추정 방법을 제안한다. 제안한 방법은 크게 두 과정으로 이루어지는데, 첫번째는 탐색 범위를 일정하게 유지하면서, 탐색 영역의 중앙에서 가장자리로 점진적 탐색을 행하여 대략적인 움직임을 추정한다. 두번째 과정에서는, 이전 단계까지의 최소 오차 참조 탐색점을 중심으로, 움직임 벡터의 통계적 특성을 이용하여 제한된 인접 탐색점에 대해서만 탐색하여 최종 움직임 벡터를 결정한다.

탐색 영역의 크기를 -7~7까지로 두었을 때를 예로 제안한 방법을 설명하면 다음과 같다. 탐색 초기 단계에서부터 마지막 단계의 직전 단계까지는 탐색 범위를 2로 하고, 점진적으로 최소 정합 오차 탐색점을 찾는다. 그림 3에서는, 중간 단계에서의 탐색을 나타내었다. 그림 3(a)와 (b)에서는 각각 이전 단계에서의 최소 오차 탐색점의 위치에 따라, 현재 단계에서 탐색할 참조 탐색점의 위치를 달리함을 나타내었다. 마지막 세부적 탐색과정으로 넘어가기 전까지는 탐색 범위가 2로 고정된 5×5크기의 탐색창으로 대략적인 움직임을 따라

가게 된다. 이 때 이전 단계와 현재 단계에서 서로 중첩되는 영역에 대해서는, 현재 단계에서는 탐색을 하지 않음으로써 계산량을 줄일 수 있다. 이전 단계의 최소 오차 참조 탐색점의 위치에 따라 수평 및 수직과 대각선 방향으로 달리 진행함을 알 수 있다.



(a)



(b)

- : The searched point at the previous step
 - : The minimum error search point at the previous step
 - : The search point at the present step
- 그림 3. 중간 단계에서의 탐색
- Fig. 3. The search at the intermediate step.

그림 4에서는 마지막 단계에서 세부 탐색을 위한 세 가지 탐색 방법을 나타내었다. 점진적 탐색과정에서 결정된 최적 탐색점 주위의 인접 8점중 움직임 벡터의 통계적 특성으로 구해진 TH_H 와 TH_L 의 임계값에 따라 탐색 방법을 달리함을 알 수 있다. 즉, P_H 와 P_L 의 MAE 차가 TH_L 보다 작을 경우에는 예측 확률이 크지 않지만, 인접 탐색점 간의 MAE 차가 크지 않을 것으로 해석하여, P_H 와 P_L 사이의 한 점만을 탐색한다. 만약 P_H 와 P_L 의 MAE 차가 TH_H 와 TH_L 사이일 경우에는, 예

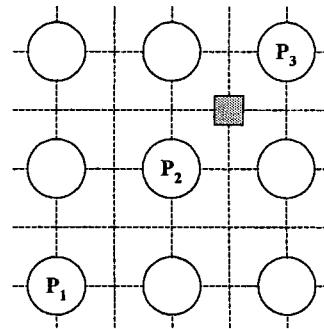
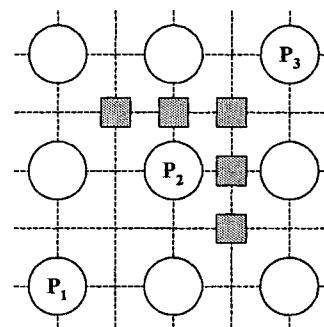
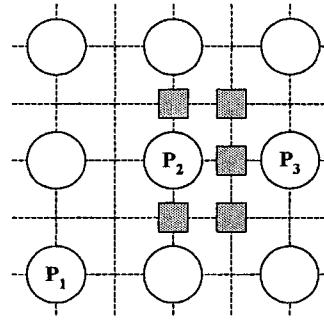
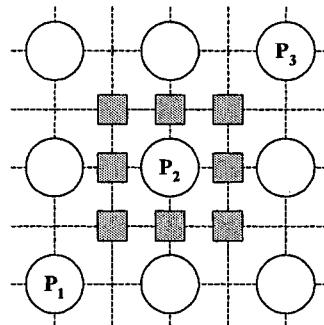
(a) $(MAE(P_1)-MAE(P_3)) < TH_L$ (b) $TH_H < ((MAE(P_1))-MAE(P_3))$ (c) $TH_H < ((MAE(P_1))-MAE(P_3))$ (d) $TH_L < ((MAE(P_1))-MAE(P_3)) < TH_H$

그림 4. 마지막 단계에서 MAE차에 따른 세부 탐색 패턴

Fig. 4. The fine search pattern according to MAE difference at the last step.

측 확률이 크지 않고, 예측이 잘못될 경우 움직임 추정 오차에 큰 영향을 미칠것이므로, 인접 8점 모두를 탐색 한다. 그리고 TH_H 보다 클 경우에는 MAE 차가 커서 성능에 많은 영향을 미치지만 확률값이 거의 1에 가까우므로 P_L 방향의 다섯 점만 탐색한다.

제안한 알고리듬을 단계별로 설명하면 다음 식 (1)과 같다.

Step 1 : Coarse search in the middle of the search area and find P_1 if ($P_1 = (0,0)$)

fine search around P_1 and find MV

else go to step 2

Step 2 : Coarse search with the search pattern of Fig. 3 and find P_2 if ($MAD(P_2) > MAD(P_1)$)

go to step 4

else go to step 3

Step 3 : Coarse search with the search pattern of Fig. 3 and find P_3 go to step 4

else go to step 3

Step 4 : if ($MAD(P_3) < MAD(P_2)$)

fine search around P_3 and find MV

else fine search with the search pattern of Fig. 4 and find MV

Coarse search : Search with the search distance as 2
fine search : Search with the search distance as 1

그림 5에서는, 움직임 벡터가 (4, -5)인 경우일 때, 제안한 방법으로 움직임을 추정한 예를 나타내었다. 이 그림은 P_1 , P_2 , P_3 중 P_2 의 MAE가 가장 낮고, P_1 의 MAE가 가장 클 경우에 대한 예다.

IV. 실험 결과 및 고찰

제안한 고속 움직임 추정 방법의 성능을 평가하기 위해 전역 탐색 방법, TSS 방법^[3], 그리고 FSS 방법^[6]과 비교하여 컴퓨터 모의 실험을 행하였다. 블록의 크기는 8×8 및 16×16 , 탐색 범위는 $-7 \sim 7$ 로 하였고 정합 척도는 MAE를 사용하였다. 객관적인 성능을 비교하기 위해 화질의 척도로는 PSNR을 사용하였고, 계산

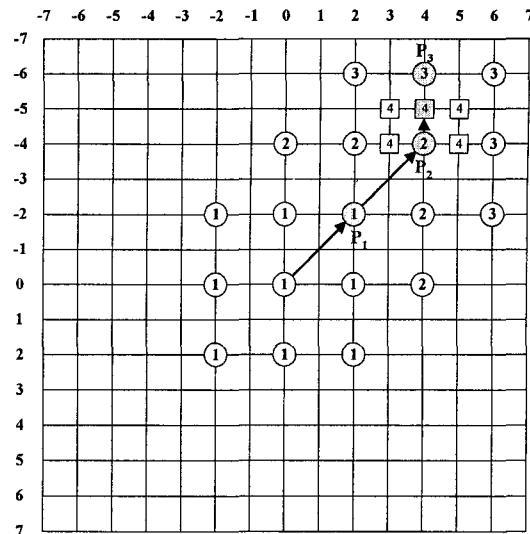


그림 5. 움직임 벡터가 (4, -5)인 경우 제안한 움직임 추정의 예.

$$(MAE(P_1) - MAE(P_3) > TH_H)$$

Fig. 5. An example of the proposed motion estimation for the motion vector(4, -5).

$$(MAE(P_1) - MAE(P_3) > TH_H)$$

량의 척도로는 움직임 추정의 대부분을 차지하는 블록 당 평균 정합 횟수를 사용하였다.

표 1. 제안한 방법과 기존 방법의 PSNR[dB] 비교(블록 크기 : 8×8)

Table 1. The PSNR[dB] comparison of the proposed and the conventional methods.
(Block size: 8×8).

Sequences	FSBMA	TSS ^[3]	FSS ^[6]	Proposed
FOOTBALL	26.08	22.20	23.26	23.00
TABLE TENNIS	31.02	28.94	29.67	29.12
SALESMAN	35.97	35.58	35.68	35.46

움직임 벡터의 통계적 특성을 구하기 위해서, FOOTBALL, TABLE TENNIS, FLOWER GARDEN, MOBILE 및 SALESMAN 영상, 각 100 프레임씩에 대하여 실험하였다. 본 논문에서 제안한 가정을 검증하기 위해, 인접 참조 탐색점들의 크기에 따른 확률값을 조사한 결과, 모든 영상에서 비슷한 분포를 확인할 수 있었다. 제안한 방법의 마지막 탐색 단계에서의 문턱값 TH_L 및 TH_H 는 실험적으로 각각 4, 7로 구하였으며,

표 2. 제안한 방법과 기존 방법의 계산량 비교
(블록 크기 : 8×8)

Table 2. The comparison of the computational complexity of the proposed and the conventional methods. (Block size: 8×8).

Sequences	FSBMA	TSS ^[3]	FSS ^[6]	Proposed
FOOTBALL	225	25	18.84	16.80
TABLE TENNIS	225	25	18.53	15.52
SALEMAN	255	25	17.27	14.70

표 3. 제안한 방법과 기존 방법의 PSNR[dB]비교 (블록 크기 : 16×16)

Table 3. The PSNR[dB] comparison of the proposed and the conventional methods. (Block size: 16×16).

Sequences	FSBMA	TSS ^[3]	FSS ^[6]	Proposed
FOOTBALL	22.32	21.87	21.68	21.46
TABLE TENNIS	29.53	28.12	28.53	28.00
SALEMAN	34.89	34.72	34.74	34.58

표 4. 제안한 방법과 기존 방법의 계산량 비교
(블록 크기 : 16×16)

Table 4. The comparison of the computational complexity of the proposed and the conventional methods. (Block size: 16×16).

Sequences	FSBMA	TSS ^[3]	FSS ^[6]	Proposed
FOOTBALL	225	25	18.11	16.49
TABLE TENNIS	225	25	17.70	15.16
SALEMAN	255	25	16.11	14.81

이 값들로 제안한 고속 움직임 추정 방법을 모의 실험하여, 기존 방법들과의 성능 비교를 표 1에 나타내었다. 실험 영상으로는 352×240 크기의 SIF 영상 FOOTBALL 및 TABLE TENNIS을 각각 100 프레임씩 사용하였다. 제안한 방법이 TSS 방법보다는 PSNR 측면에서 우수한 결과를 보였으며, FSS 방법에 비해서는 PSNR이 약간 떨어짐을 볼 수 있다. 또한 제안한 방법과 기존의 방법들과의 계산량, 즉 블록당 평균 정합 횟

수를 표 2에 나타내었다. 마지막 탐색 단계에서, 움직임 벡터의 통계적 특성을 이용하여 인접 참조 탐색점들의 크기에 따라 3 가지의 다른 탐색 패턴으로 탐색함으로써, 제안한 방법의 블록당 평균 탐색점 수가 기존의 고속 움직임 추정 방법들보다 2~10점 정도 줄어듬을 확인할 수 있다. 또한 16×16 블록 크기에 대한 제안한 방법과 기존 방법들과의 성능 비교를 표 3 및 표 4에 나타내었다. 표 3에서 보는 바와 같이, 블록 크기가 16×16 으로 커짐에 따라서 모든 움직임 추정 방법들에 대해서 화질이 전체적으로 나빠진다. 이는 블록의 크기가 커질 경우, 움직임 추정 성능이 떨어지기 때문이다. 그러나 표 4에서 보는 바와 같이, 평균 탐색점수는 8×8 블록 크기일 때와 마찬가지로 기존의 고속 움직임 추정 방법들보다 2~10 점 정도 떨어짐을 확인할 수 있다.

V. 결 론

본 논문에서는, 탐색 영역 내에서의 움직임 벡터의 통계적 특성을 조사하여, 이를 이용한 새로운 고속 움직임 추정 방법을 제안하였다. 탐색 영역 내에서 인접 참조 탐색점들의 정합 오차의 크기에 따른 움직임 벡터의 분포 특성을 조사하였다. 그 결과, 이전 단계에서의 인접 참조 탐색점들의 정합 오차의 차가 클 경우, 마지막 단계에서 최소 오차 탐색점이 정합 오차가 작은 인접 참조 탐색점 근방에서 일어날 확률값이 거의 “1”이었으며, 또한 이전 단계에서 인접 참조 탐색점들의 정합 오차의 차가 아주 작을 경우에는, 그 방향에 상관 없이 정합 오차의 차가 아주 작게 나타났다. 그러므로 이와 같은 움직임 벡터의 통계적 특성으로부터 문턱값을 정하여, 움직임 추정시 세 가지 탐색 패턴으로 분류하여 탐색함으로써, 모든 탐색점들에 대해서 세부 탐색하지 않고, 탐색점 수를 제한함으로써, PSNR의 큰 저하 없이, 계산량을 크게 줄일 수 있었다.

참 고 문 헌

- [1] A. N. Netravali and B. G. Haskell, *Digital Pictures Representation and Compression*, Plenum Press, New York and London, 1989.
- [2] J. R. Jain, A. K. Jain, “Displacement measurement and its application in

- interframe image coding," *IEEE Trans. on Comm.*, vol. COM-29, No. 12, Dec. 1981.
- [3] T. Koga, K. Inuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," Proc. Nat. Telecommun. Conf., pp. G5.3. 1-G5.3.5, Nov./Dec. 1981.
- [4] R. Strinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Comm.*, vol. COM-33, pp. 1011-1014, Sept. 1985.
- [5] M. Ganbari, "The cross search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. COM-38, no. 7, pp. 950-953, July 1990.
- [6] L.-M. Po and W.-C. Ma, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation," *IEEE Trans. on Circuits and Systems for Video Technol.*, vol. 6, no. 3, pp. 313-317, June 1996.

저자 소개

朴 大 圭(正会員)

1972년 2월 21일생. 1992년 3월~1998년 2월 경북대학교 전자공학과 학사. 1998년 3월~현재 경북대학교전자공학과 석사과정. 주관심 분야는 HDTV, 영상 신호 처리 및 압축

崔 正 鉉(正会員) 第35卷 S編 第5號 參照

李 京 桓(正会員) 第35卷 S編 第5號 參照

李 法 基(正会員) 第35卷 S編 第5號 參照

金 德 奎(正会員) 第35卷 S編 第5號 參照