

예측 움직임 벡터를 이용한 고속 적응 블럭 정합 알고리즘

정희원 신용달*, 김영춘*, 이건일**

Fast Adaptive Block Matching Algorithm using Predictive Motion Vector

Yong-Dal Shin*, Young-Choon Kim*, Kuhn Il Lee** *Regular Member*

요약

본 논문에서는 예측 움직임 벡터를 이용한 고속 적응 블럭 정합 알고리즘을 제안하였다. 제안 방법에서는 현재 블럭의 상단 블럭 및 좌측 블럭의 움직임 벡터 평균값으로 예측 움직임 벡터를 구하고, 이 두 블럭에 대한 움직임 정도에 따라 블럭을 움직임이 없는 블럭, 작은 블럭, 중간정도 블럭 혹은 큰 블럭으로 분류한 후, 각 블럭의 특성에 따라 적응적으로 움직임 벡터를 추정한다. 제안한 방법의 성능 평가를 위한 컴퓨터 시뮬레이션에서 제안 방법의 PSNR 성능은 8×8 블록크기에서 기존의 고속 움직임 추정 방법인 NTSS (new three step search) 방법보다 0.27dB에서 0.77dB 정도 증가하며, 계산량은 약 36% 에서 41% 정도 크게 감소하였다.

ABSTRACT

We present a fast adaptive block matching algorithm using predictive motion vector. In the presented method, the predictive motion vector is average of motion vector of upper and left blocks for current block. The block is classified into one of four motion categories: stationary, quasi-stationary, medium-motion or large-motion block according to motion of the left and upper blocks, and then each block estimates the motion vector adaptively. By the simulation, the PSNR of our algorithm increase 0.27dB~0.77dB more than that of NTSS (new three step search) method. The computational complexity of the presented method decreased about 36% ~ 41% more than that of NTSS method.

I. 서론

통신 매체의 발달로 고속으로 정보를 전달하는 것이 가능하게 되었으며, 특히 여러 가지 정보 중에서 동영상 정보를 전송하기 위한 표준안으로서 MPEG이 제안되었다. 동영상은 일반적으로 움직임 보상 부호화 (motion compensation coding)를 이용하여 높은 데이터 압축이 이루어진다. 이 방법에 의한 데이터의 압축은 이전 프레임에 이용하여 움직임 추정 및 보상을 수행하고, 이때 추정된 움직임 벡터 (motion vector)에 의해서 보상된 영상과 원영

상의 차이 신호를 부호화 한다.^[1] 이 방법은 크게 화소 순환 알고리즘 (pel recursive algorithm: PRA) 과 블럭 정합 알고리즘 (block matching algorithm: BMA)으로 나누어진다.^{[1][2]} 블럭 정합 알고리즘은 화소 순환 알고리즘에 비하여 수행시간이 적게 소요되고 하드웨어 구현이 용이하기 때문에 움직임 보상 부호화에 널리 사용되고있다.

블럭 정합 알고리즘은 입력영상을 임의의 작은 블럭으로 나눈 후, 한 블럭내의 모든 화소는 동일 방향의 움직임을 갖는다고 가정하여 움직임 벡터를 추정하는 것으로, 이전 프레임에서 설정된 탐색영역

* 영동대학교 전자공학부(ydshin@kachi.youngdong.ac.kr), ** 경북대학교 전자전기공학부
논문번호: 98348-0811, 접수일자: 1998년 8월 11일

에서 어떤 블럭이 현재프레임의 정해진 블럭으로 이동하였는가 찾는 것이다. 이때 블럭내의 모든 화소에 대하여 오차를 구하고 그 중 가장 작은 오차를 갖는 탐색점의 값을 움직임 벡터로 결정하는 방법을 전 탐색 블럭 정합 알고리즘 (full search block matching algorithm)라 한다. 이 방법은 탐색 영역내에서 최적인 움직임 벡터를 추정할 수 있으나, 정합하는 경우 탐색점이 너무 많으므로 많은 계산량을 필요로 하는 단점이 있다. 따라서 계산량을 줄이면서 거의 정확한 움직임 벡터를 찾을 수 있는 고속의 블럭 정합 알고리즘 즉, 2차원 로그 탐색 알고리즘, 3단계 탐색(three step search: TSS) 알고리즘, 교차 탐색 (cross search) 알고리즘, 새로운 3단계 탐색 (new three step search: NTSS) 알고리즘 등이 연구되었다.^{[1]-[6]} 이들 방법 중에서 계산량이 간단하면서도 거의 정확한 움직임 벡터를 추정하는 방법은 새로운 3단계 탐색 알고리즘^[6]이다.

고속 블럭 정합 알고리즘인 NTSS^[6] 방법에서는 움직임 벡터를 추정하는 과정에서, 제 1 단계에서는 블럭의 움직임 정도는 고려하지 않고 항상 고정적으로 탐색점을 17개 사용하여 움직임 벡터를 추정한다. 이와 같은 방법으로 움직임 벡터를 추정할 경우 움직임이 없는 블럭 및 적은 블럭에서는 계산량이 증가되는 단점이 있으며, 또한 물체의 움직임이 클 경우 움직임 벡터를 정확하게 추정하지 못하기 때문에 화질의 열화가 발생된다.

BMA에서 정확한 움직임 벡터를 추정하기 위해서는 블럭의 크기가 작거나 탐색 영역의 크기가 커야 한다. 그러나 이 경우 계산량이 많아지고 전송해야 할 움직임 벡터에 대한 정보의 양이 많아진다. 따라서, 계산량의 증가 없이 더욱 정확한 움직임 벡터를 추정하기 위하여 현재 블럭과 상관성이 많은 인접 블럭들의 움직임 벡터를 이용하여 현재 블럭의 움직임을 예측하는 예측 움직임 추정 기법이 연구되고 있다.^{[7]-[10]} 이 방법에는 움직임 벡터의 공간적인 상관성을 이용하는 방법과 시간적인 상관성을 이용하는 방법 및 시 공간적인 상관성을 이용하는 방법이 있다.^{[7]-[10]}

본 논문에서는 예측 움직임 벡터를 이용한 고속 적응 블럭 정합 알고리즘을 제안하였다. 제안 방법에서는 현재 블럭의 상단 블럭 및 좌측 블럭의 움직임 벡터 평균값으로 예측 움직임 벡터를 구하고, 이 두 블럭에 대한 움직임 정도에 따라 블럭을 움직임이 없는 블럭, 작은 블럭, 중간정도 블럭 혹은 큰 블럭으로 분류한 후, 각 블럭의 특성에 따라 적

응적으로 움직임 벡터를 추정한다.

제안한 방법의 성능 평가를 위한 컴퓨터 시뮬레이션에서 제안 방법의 PSNR 성능은 8×8 블럭크기에서 기존의 고속 움직임 추정 방법인 NTSS (new three step search) 방법보다 0.27dB에서 0.77dB 정도 증가하며, 계산량은 약 36% 에서 41% 정도 크게 감소하였다.

II. 이 론

1. 예측 움직임 추정 기법

예측 움직임 추정 기법은 움직임 벡터의 공간적인 상관성을 고려하여 현재 블럭의 예측 움직임 벡터를 구하는 것이다. 이 예측값으로 탐색 영역의 위치를 결정한 후, 이 영역에서 현재 블럭의 움직임 벡터를 추정한다. 예측 움직임 추정 기법은 그림 1에서와 같이 신호의 인과성 (causality) 및 움직임 벡터들의 공간적인 상관성을 고려하여 현재 블럭의 상단 블럭에 대한 움직임 벡터 $V(v-1, h)$ 및 좌측 블럭의 움직임 벡터 $V(v, h-1)$ 을 이용하여 현재 블럭의 예측 움직임 벡터 $P(v, h)$ 를

$$P(v, h) = \begin{pmatrix} P_v(v, h) \\ P_h(v, h) \end{pmatrix} = \begin{pmatrix} [V_v(v-1, h) + V_v(v, h-1)]/2 \\ [V_h(v-1, h) + V_h(v, h-1)]/2 \end{pmatrix} \quad (1)$$

와 같이 구하였다. 여기서 $P_v(v, h)$ 및 $P_h(v, h)$ 는 각각 수직 및 수평 방향의 움직임 예측값이며, $V_h(\cdot)$ 및 $V_v(\cdot)$ 는 각각 수평 및 수직 방향의 움직임 거리이다.

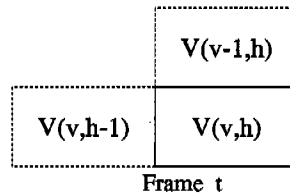


그림 1. 움직임 예측을 위한 인접블럭의 움직임 벡터
Fig. 1. Motion vector of neighboring blocks for motion prediction.

이 예측 기법에서 현재블럭의 움직임 예측은 공간적으로 인접한 상단 블럭 및 좌측 블럭과 상관성이 높기 때문에 공간적으로 인접하는 두개의 블럭을 사용하였다. 만약 공간적으로 인접한 두개 블럭의 움직임 벡터의 차이가 문턱값 THD보다 클 경우

즉,

$$|V_h(v-1, h) - V_h(v, h-1)| > THD \quad (2)$$

$$|V_v(v-1, h) - V_v(v, h-1)| > THD \quad (3)$$

일때 공간적으로 인접한 두 블럭간의 상관성이 거의 없기 때문에 현재 블럭의 움직임을 예측하기가 곤란하여 예측 움직임 벡터 $P(v, h)$ 를 (0,0)으로 한다.

탐색영역의 크기에 대하여 살펴보면 움직임 벡터를 예측 하지 않은 경우에서 BMA를 수행했을 때의 탐색영역 S 는

$$S = \{(k, l) \mid -M \leq k, l \leq M\} \quad (4)$$

이다. 여기서, M 은 최대 움직임 거리, (k, l) 은 탐색 영역의 탐색점이다. 이와 같이 설정된 탐색 영역에서 움직임 벡터를 구하여 전송하게 된다. 그러나 움직임 벡터의 공간적인 상관성을 이용한 예측 움직임 추정 기법은 예측 움직임 벡터 P 를 고려하여 탐색 영역이 설정된다. 즉 탐색 영역 S' 은

$$S' = \{(k, l) \mid -M + P_v \leq k \leq M + P_v, -M + P_h \leq l \leq M + P_h\} \quad (5)$$

로 설정된다. 여기서 P_v 및 P_h 는 수직 및 수평 방향의 움직임 예측값이다. 움직임 벡터를 예측하지 않은 방법에 의한 탐색 영역 S 및 움직임 벡터를 예측한 탐색 영역 S' 을 그림 2에 나타내었다. 만약 물체의 움직임이 빠를 경우 탐색 영역 S 에서는 움직임 벡터를 정확하게 추정하기 힘들지만 예측 움직임 벡터 P 만큼 이동된 탐색 영역 S' 을 탐색함으로써 움직임 추정을 더욱 정확하게 할 수 있다.

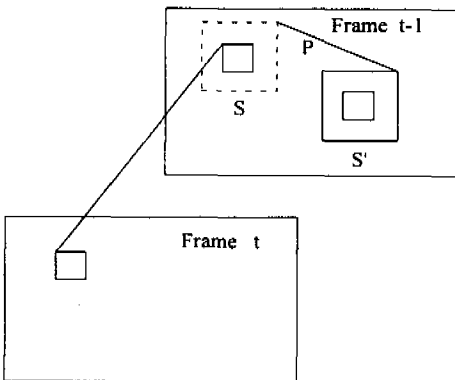


그림 2. 예측 움직임 벡터에 의한 탐색 영역
Fig. 2. Search area by predictive motion vector.

2. 제안한 예측 움직임 벡터를 이용한 고속 적응 블럭 정합 알고리즘

블럭 정합 알고리즘은 현재 프레임 t 의 (i, j) 에 있는 블럭을 이전 프레임 $t-1$ 에 있는 블럭 (i, j) 와 정합을 할 때 오차는 항상 탐색영역의 모든 탐색점에 대하여 평가를 해야만 움직임 추정이 정확하게 된다. 이와 같이 모든 탐색점에 대하여 움직임을 추정한다면 그 계산량은 매우 많아진다. 또한 고속으로 움직임을 추정한 뒤 위해서 탐색점 수를 너무 적게하여 블럭 정합 알고리즘을 수행하게 되면 국부최소에 빠져서 부정확한 움직임을 추정하게 되어 복원 화상에서 화질이 저하된다.

본 논문에서는 예측 움직임 벡터를 이용한 고속 적응 블럭 정합 알고리즘을 제안하였다. 제안 방법은 식 (1)에서와 같이 현재프레임에서 현재블럭의 상단 및 좌측 블럭의 움직임 벡터 평균값으로 예측 움직임을 구하고, 이 두 개 블럭에 대한 수평 및 수직 방향으로의 움직임 정도를 나타내는 척도인 등급 결정 계수 (classification coefficient : CC)의 값에 따라 움직임이 거의 없는 블럭, 작은 블럭, 중간정도인 블럭 혹은 큰 블럭으로 분류한 후, 각 블럭의 특성에 따라 적응적으로 움직임을 추정한다. 제안한 고속 적응 블럭 정합 알고리즘의 흐름도는 그림 3과 같다. 이 그림에서 움직임을 나타내는 척도인 등급 결정 계수는 공간적인 상관성을 고려하여, 그림 1의 $V_v(v-1, h)$ 및 $V_v(v, h-1)$ 각각에서 수평 및 수직 방향으로 움직임의 정도인 $V_v(v-1, h)$, $V_v(v, h-1)$, $V_h(v-1, h)$ 및 $V_h(v, h-1)$ 네 개 절대값중에서 최대값 즉,

$$CC = \max(|V_v(v-1, h)|, |V_v(v, h-1)|,$$

$$|V_h(v-1, h)|, |V_h(v, h-1)|) \quad (6)$$

을 CC로 정의한다.

$CC \leq th_1$ 이면 움직임이 거의 없는 블럭, $th_1 < CC \leq th_2$ 이면 움직임이 적은 블럭, $th_2 < CC \leq th_3$ 이면 중간정도인 블럭, 이 외는 움직임이 큰 블럭으로 분류한 후, 각 블럭에 대해서 적응적으로 블럭 정합 알고리즘을 사용하여 움직임을 추정한 뒤 여기서, th_1 , th_2 및 th_3 는 문턱값이다.

그림 3에서 제안 방법이 더욱 정확한 움직임을 추정한 뒤 위해서 현재블럭과 인접하는 좌측 및 상단 블럭의 움직임 벡터의 평균값으로 예측 움직임을 구한다. 이와 같이 움직임을 예측하면

식 (5)와 같이 예측 움직임 벡터 P 만큼 이동된 탐색 영역에서 움직임 벡터를 추정하기에 더욱 정확한 움직임 벡터를 구할 수 있다. 또한 제 1, 2, 3, 및 4단계는 다음과 같다.

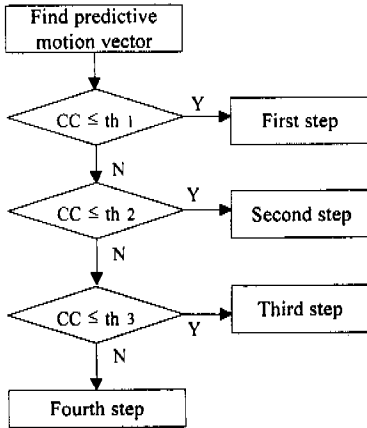


그림 3. 제안한 고속 적응 블럭 정합 알고리즘 블럭선도
Fig. 3. The block diagram of the proposed fast adaptive BMA.

1. 제 1단계: $CC \leq th_1$ 인 경우이며, 이는 움직임이 거의 없는 블럭으로 그림 4(a)와 같이 검은색 원점인 예측 움직임 벡터 $P(v,h)$ 를 중심으로 인근의 네 모점에 블럭 정합을 수행하여 오차가 최소인 화소를 움직임벡터로 결정한다.

2. 제 2단계: $th_1 < CC \leq th_2$ 경우이며, 이는 움직임이 적은 블럭으로 그림 4(b)와 같이 검은색 원점인 예측 움직임 벡터 $P(v,h)$ 를 중심으로 인근의 8개의 탐색점(검은 네모점)을 이용하여 블럭정합을 하여 가장 작은 오차를 갖는 탐색점 (흰 삼각형)을 찾는다. 그리고 이 탐색점을 중심으로 검은 삼각형인 인근의 8개에 대하여 한번 더 블럭정합을 수행하여 최소 오차를 갖는 탐색점을 움직임 벡터로 결정한다.

3. 제 3단계: $th_2 < CC \leq th_3$ 인 경우이며, 이는 움직임이 중간정도의 블럭으로 그림 4(c)와 같이 검은색 원점인 예측 움직임 벡터 $P(v,h)$ 를 중심으로 검은색 네모점 8개의 탐색점에 대하여 블럭정합을 수행한다. 이 중에서 최소 오차를 갖는 탐색점을 기준으로 하여, 이 탐색점 주위 검은점 삼각형인 인근의 8개에 대하여 또다시 블럭정합을 수행하여, 최소 오차를 갖는 탐색점을 움직임 벡터로 결정한다.

4. 제 4단계: $CC > th_3$ 인 경우이며, 이는 움직임 큰 영역으로서 탐색점의 수를 많이하여 움직임 벡

터를 결정하여야 한다. 따라서 예측 움직임 벡터 $P(v,h)$ 에서 NTSS방법에서와 같이 수행하여 움직임 벡터를 결정한다.

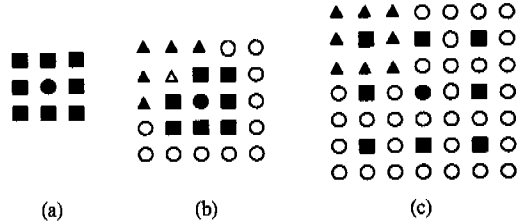


그림 4. 제안 방법의 (a)제 1단계, (b)제 2단계 및 (c)제 3 단계
Fig. 4. (a) The first, (b) second, and (c) third step of the proposed method.

계산량 측면에서는 기존의 NTSS 방법의 제 1단계에서는 17개의 탐색점, 움직임이 클 경우는 33개 탐색점이 필요하다. 반면에 제안 방법에서는 제 1단계에서 9개 탐색점, 제 2단계에서 첫번째 탐색점 9개와 두 번째의 탐색점 중 (-1,-1), (-1,1), (1,-1), (1,1)에서는 5개, (-1,0), (1,0), (0, -1), (0,1)에서는 3개의 탐색점이 필요하고, 제 3단계에서는 첫 번째 9개, 두 번째 탐색에서는 8개의 탐색점을 필요로 한다. 제 4단계에서는 NTSS와 계산량이 같다.

III. 실험 결과 및 고찰

본 논문에서 제안한 방법에 대하여 성능을 평가하기 위하여 컴퓨터 모의 실험을 수행하였다. 본 실험에서는 720×480 크기의 FOOTBALL, FLOWER GARDEN 및 SUSIE 각 60 프레임을 사용하였다. FOOTBALL은 카메라 및 물체의 움직임이 빠른 영상, FLOWER GARDEN은 정지된 물체에 카메라가 빠르게 움직이는 영상, SUSIE는 물체가 천천히 움직이는 영상 및 빠르게 움직이는 영상이 혼합되어 있다. 본 실험에서는 블럭 크기를 8×8 및 16×16, 탐색범위 -7~7로, 정합의 척도는 평균절대오차 (mean absolute difference)를 사용하여 계산량 및 PSNR 에 대하여 평가하였다.

제안 방법에서는 부록 1에서와 같이 CC 값이 작으면 움직임의 정도가 작고, CC값이 크면 움직임의 정도가 크게되는 특성을 기초로하여 적응적으로 블럭정합 알고리즘을 행하였다. 제안 방법에서의 문턱값은 PSNR 및 계산량을 고려하여 실험적으로 $th_1=0$, $th_2=2$ 및 $th_3=4$ 로 결정하였다.

제안 방법과 기존의 NTSS 방법에 대하여 프레임

변화에 따른 PSNR 결과는 그림 5에서와 같다. 그리고 각 영상들에 대한 평균 PSNR 및 계산량의 결과는 각각 표 1 및 2와 같다. 이 때 계산량은 움직임 벡터를 추정하기 위해서 요구되는 탐색점의 수를 나타낸 것이다. 이를 식으로 표현하면 $9N_1+16N_2+16N_3+K$ 이다. 여기서, N_1 , N_2 및 N_3 는 각각 1, 2 및 3단계로 선택되어지는 블럭 수를 나타내고, 또한 9, 16 및 16은 1, 2 및 3단계 각각의 알고리즘에서의 탐색점 수, K는 제 4 단계에서의 움직임 추정 방법인 NTSS 수행시의 탐색점 수를 나타낸다. 대부분 영상에서는 움직임이 큰 블럭의 수가 적고, 상대적으로 움직임이 작은 블럭의 수가 많이 존재하기에 계산량이 줄어든다.

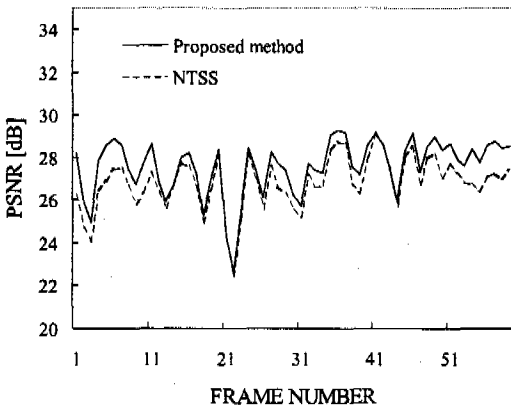


그림 5. FLOWER GARDEN 영상의 블럭크기 8×8 (탐색 범위: -7~7)에 대한 PSNR.
Fig. 5. PSNR of block size 8×8 (search area: -7~7) for FLOWER GARDEN image.

이 그림에서 제안 방법이 기존의 NTSS 방법보다 PSNR이 증가함을 알 수 있다. 또한 표 1에서와 같이 8×8 블럭크기에 대하여 평균 PSNR이 기존의 NTSS 방법보다 제안 방법이 0.27dB에서 0.77dB 증가함을 알 수 있다. 이 결과로부터 예측 움직임 벡터를 이용하여 인접한 블럭의 움직임의 정도에 따라 적응적으로 블럭 정합 알고리즘을 수행하여 현재블럭의 움직임 벡터를 결정하면 현재블럭의 움직임 벡터를 더욱 정확하게 추정하여 PSNR 증가함을 알 수 있다. 또한 표 2에서의 계산량은 탐색점의 수 및 퍼센트 (%) 정도를 각각 나타내었다. 퍼센트 정도는 NTSS의 계산량을 100.00 % 기준으로 해서 제안 방법에서의 계산량을 퍼센트 (%)로 표시한 것이다. 이 표에서 보는데와 같이 기존의 NTSS 방법에 비하여 제안방법의 계산량이 36.26% 에서 41.60% 감소되었다. 이와 같이 계산량이 감소되는

이유는 동영상에서는 블럭마다 움직임의 정도가 다르다는 특성을 이용하여 예측 움직임 벡터 및 인근 블럭의 움직임의 정도에 따라 적응적으로 움직임 벡터를 추정하기 때문이다.

표 1. 각 방법에 대한 평균 PSNR [dB]
Table 1. The average PSNR [dB] for each method.

Block size	Image	Proposed method	NTSS
		PSNR	PSNR
8×8	FLOWER GARDEN	27.56	26.79
	SUSIE	36.07	35.40
	FOOTBALL	24.33	24.06
	FLOWER GARDEN	26.94	26.78
16×16	SUSIE	35.37	34.85
	FOOTBALL	22.88	22.83

표 2. 각 방법에 대한 계산량
Table 2. The computational complexity for each method.

Block size	Images	Proposed method		NTSS	
		No. of search points	Percent	No. of search points	Percent
8×8	FLOWER GARDEN	5165394	58.40	8844208	100.00
	SUSIE	4697143	58.94	7968984	100.00
	FOOTBALL	4804224	62.79	7653888	100.00
16×16	FLOWER GARDEN	1224584	59.08	2072780	100.00
	SUSIE	1074315	58.27	1843660	100.00
	FOOTBALL	1147867	63.74	1800940	100.00

이상의 결과로부터 제안 방법이 블럭 정합 알고리즘 방법중 성능이 우수한 NTSS 방법과 비교해서 PSNR은 증가하면서도 계산량을 많이 줄여 고속으로 움직임 벡터를 추정할 수 있는 효과적인 방법임을 확인할 수 있었다.

IV. 결론

본 논문에서는 예측 움직임 벡터를 이용한 고속 적용 블럭 정합 알고리즘을 제안하였다. 제안 방법에서는 현재 블럭의 상단 블럭 및 좌측 블럭의 움직임 벡터 평균값으로 예측 움직임 벡터를 구하고,

이 두 블럭에 대한 움직임 정도에 따라 블럭을 움직임이 없는 블럭, 작은 블럭, 중간정도 블럭 혹은 큰 블럭으로 분류한 후, 각 블럭의 특성에 따라 적응적으로 움직임 벡터를 추정한다.

제안한 방법의 성능 평가를 위한 컴퓨터 시뮬레이션에서 제안 방법의 PSNR 성능은 8×8 블럭크기에서 기존의 고속 움직임 추정 방법인 NTSS (new three step search) 방법보다 0.27dB에서 0.77dB 정도 증가하며, 계산량은 약 36%에서 41% 정도 크게 감소하였다.

참고 문헌

[1] H. G. Musmann, P. Pirch, and H. J. Grallert, "Advances in picture coding," *Proc. of IEEE*, vol. 73, no. 4, pp. 523-548, April 1985.

[2] A. K. Jain, "Image data compression: A review," *Proc. of IEEE*, vol. 69, no. 3, pp. 349-389, Mar. 1981.

[3] A. M. Tekalp, *Digital video processing*, Prentice Hall, 1995.

[4] 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./Dec. 1981.

[5] M. Ghanbari, "The cross search algorithm for motion estimation," *IEEE Trans. on Commun.*, vol. COM-38, no. 7, pp. 950-953, July 1990.

[6] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuit and Systems for Video Technology*, vol. 4, no. 4, pp. 438-442, Aug. 1994.

[7] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuit and Systems for Video Technology*, vol. 6, pp. 148-157, April 1993.

[8] S. Zafar, Y. Q. Zhang, and J. S. Baras, "Predictive block-matching motion estimation for TV coding-Part I: Inter-block prediction," *IEEE Trans. on Broadcasting*, vol.

37, no. 3, pp. 97-101, Sept. 1991.

[9] Y. Q. Zhang and S. Zafar, "Predictive block-matching motion estimation for TV coding-Part II: Inter-frame prediction," *IEEE Trans. on Broadcasting*, vol. 37, no. 3, pp. 102-105, Sept. 1991.

[10] C. H. Hsieh, P. C. Lu, J. S. Shyn, and E. H. Lu, "Motion estimation algorithm using interblock correlation," *IEE Electronic Letters*, vol. 26, no. 5, pp. 276-277, March 1990.

부록

이 부록은 FOOTBALL, FLOWER GARDEN (FL. GA.) 및 SUSIE 영상에 대하여 8×8 블럭 크기, 탐색영역 -7에서 7에 대하여 예측 움직임 벡터를 이용해서 전역탐색 알고리즘으로 움직임 벡터를 추정하여 얻은 결과를 CC에 따른 움직임 벡터 분포를 퍼센트로 나타내었다. 여기서 분포 (Dis: Distribution)는 영상에서 CC에 해당하는 범주에서의 분포의 정도, 누적값 (Acc : Accumulation)은 CC가 증가하면서 누적된 % 정도를 의미한다. 또한 움직임 벡터의 크기인 0에서 -7~7에서의 분포의 정도는 누적된 % 정도이다. 예를 들어 CC가 1 인 경우 FOOTBALL 영상에서 28.1 % (47.01 %)에서, 이 28.1%는 CC가 1 때 28.1 %, 팔호안의 47.01 %는 CC가 1보다 작거나 같을때의 분포정도이다. 즉 CC가 0 일때와 CC 가 1 일때의 합이다. 또한 예측움직임 벡터를 중심으로 움직임 벡터가 (0,0)인 경우는 (61.91%), -1에서 1인 움직임 벡터는 96.79 % (이것은 움직임 벡터가 (0,0)인 61.91 %가 더하여진 것이다), -2 에서 2 인 움직임 벡터는 97.82 % (이것은 움직임 벡터가 (-1~1)인 96.79 %가 더하여진 것이다)이다. 이와 같이 계속하여 -7에서 7 인 움직임 벡터까지는 100.00 %이다.

부록 1. 예측 움직임 벡터를 이용한 움직임 벡터의 분포 (블록 크기: 8×8, 탐색영역: -7 ~ 7)
 Appendix 1. The distribution of the motion vector using predictive motion vector
 (block size:8×8, search area: -7 ~ 7).

CC	images	Dis. [%] (Acc.)	0	-1~1	-2~2	-3~3	-4~4	-5~5	-6~6	-7~7
0	FOOTBALL	18.91(18.91)	88.85	98.78	99.00	99.10	99.24	99.37	99.62	100 %
	FL. GA.	0.18(0.18)	28.21	43.93.	59.64	70.54	77.32	83.39	90.36	100 %
	SUSIE	3.17(3.17)	66.10	89.98	94.2	95.78	97.15	98.18	99.13	100 %
1	FOOTBALL	28.1(47.01)	61.91	96.79	97.82	98.23	98.55	98.88	99.29	100 %
	FL. GA.	0.96(1.13)	17.78	53.07	67.01	76.94	83.37	89.31	95.18	100 %
	SUSIE	18.57(21.74)	44.72	83.71	90.95	93.92	95.83	97.36	98.67	100 %
2	FOOTBALL	8.43(55.44)	37.2	79.63	88.98	91.86	93.71	95.56	97.23	100 %
	FL. GA.	28.94(30.07)	82.65	92.3.	95.01	96.35	97.43	98.21	99.17	100 %
	SUSIE	22.29(44.03)	36.23	71.25	83.62	88.86	92.33	95.08	97.56	100 %
3	FOOTBALL	2.43(57.87)	13.31	47.27	70.84	79.05	84.4	88.57	92.96	100 %
	FL. GA.	15.34(45.41)	63.63	86.77	91.10	94.12	95.82	97.2	98.45	100 %
	SUSIE	9.61(53.64)	18.84	44.98	63.41	74.4	82.39	88.9	94.46	100 %
4	FOOTBALL	4.33(62.2)	10.9	39.59	64.19	76.5	83.82	88.13	92.91	100 %
	FL. GA.	12.44(57.85)	63.5	78.4	85.74	89.44	92.85	95.24	97.59	100 %
	SUSIE	9.17(62.82)	11.9	35.67	54.95	68.46	78.21	86.04	92.91	100 %
5	FOOTBALL	2.29(64.49)	5.71	22.64	40.2	59.54	69.42	77.64	85.41	100 %
	FL. GA.	11.86(69.71)	65.02	79.58	84.79	88.64	91.62	94.81	97.4	100 %
	SUSIE	5.63(68.45)	2.93	15.45	31.93	50.21	64.23	77.09	88.39	100 %
6	FOOTBALL	4.64(69.13)	6.35	24.99	39.71	57.97	69.06	77.45	86.33	100 %
	FL. GA.	9.73(79.44)	51.95	68.12	74.91	81.68	86.41	90.66	95.48	100 %
	SUSIE	6.14(74.59)	3.51	16.15	30.8	47.9	62.54	75.44	87.75	100 %
7	FOOTBALL	4.1(73.23)	2.32	9.78	18.24	28.96	44.42	54.86	67.85	100 %
	FL. GA.	5.62(85.06)	37.64	51.21	58.97	67.48	74.23	80.39	86.84	100 %
	SUSIE	4.83(79.42)	1.72	9.33	19.69	33.37	48.9	62.91	76.57	100 %
8	FOOTBALL	4.32(77.55)	4.89	19.7	32.06	42.49	59.69	70.52	80.75	100 %
	FL. GA.	3.27(88.33)	22.53	39.34	50.53	62.68	74.19	82.92	90.82	100 %
	SUSIE	4.19(83.61)	1.96	11.03	22.62	36.88	53.37	68.13	82.91	100 %
9	FOOTBALL	2.49(80.05)	3.34	12.29	21.99	31.28	43.08	58.79	7.17	100 %
	FL. GA.	4.09(92.42)	28.04	47.86	59.75	69.43	79.56	86.51	92.83	100 %
	SUSIE	2.67(86.29)	1.51	8.99	18.73	30.91	46.44	62.04	79.28	100 %
10	FOOTBALL	4.08(84.13)	5.73	19.66	31.4	41.24	51.26	64.74	76.88	100 %
	FL. GA.	3.58(96.00)	22.66	44.64	56.45	65.95	75.97	84.03	91.46	100 %
	SUSIE	3.01(89.3)	2.69	12.03	22.32	33.2	46.28	60.82	77.56	100 %
11	FOOTBALL	2.37(86.5)	2.78	12.49	21.69	30.55	40.8	51.94	68.89	100 %
	FL. GA.	1.45(97.45)	7.95	24.91	36.83	48.79	60.77	74.48	85.98	100 %
	SUSIE	1.9(91.19)	1.77	9.76	19.03	29.1	41.15	56.04	73.72	100 %
12	FOOTBALL	3.14(89.64)	3.06	14.68	25.21	34.56	43.85	53.35	70.1	100 %
	FL. GA.	0.80(98.25)	2.40	12.10	22.55	33.79	46.99	62.55	78.43	100 %
	SUSIE	2.13(93.32)	2.24	11.18	21.72	32.75	44.25	56.81	73.19	100 %
13	FOOTBALL	2.21(91.85)	2.6	10.87	20.4	29.81	39.33	50.06	64.54	100 %
	FL. GA.	0.59(98.84)	1.33	8.63	19.71	33.35	46.24	59.08	74.05	100 %
	SUSIE	1.44(94.76)	2.05	10.96	20.67	30.64	41.05	52.47	68.57	100 %
14	FOOTBALL	2.74(94.59)	1.82	9.63	18.19	26.4	36.77	46.98	62.39	100 %
	FL. GA.	0.43(99.27)	1.96	7.71	18.69	32.73	45.82	57.45	71.93	100 %
	SUSIE	1.90(96.66)	1.96	10.27	20.34	29.6	40.46	51.35	68.21	100 %
15	FOOTBALL	5.41(100.00)	0.65	4.4	9.0	14.49	21.6	29.86	44.23	100 %
	FL. GA.	0.73(100.00)	0.64	5.66	16.89	28.72	41.88	53.06	67.51	100 %
	SUSIE	3.34(100.00)	0.83	4.72	9.57	15.28	21.96	29.22	44.7	100 %

