

論文97-34S-9-9

움직임 벡터 추정을 위한 고속 적응 블럭 정합 알고리즘

(Fast Adaptive Block Matching Algorithm for Motion Vector Estimation)

申容達*, 李承珍**, 金慶圭**, 鄭元植**, 金榮春**,
李奉洛***, 蔣鐘國*, 李健一**

(Yong-Dal Shin, Seung-Jin Lee, Kyoung-Kyoo Kim, Won-Sik Cheong,
Young-Choon Kim, Bong-Lak Lee, Jong-Kook Jang, and Kuhn-Il Lee)

요 약

본 논문에서는 움직임 벡터를 좀 더 정확하게 추정할 수 있도록 가변탐색영역 및 부표본화를 이용한 고속 적응 블럭 정합 알고리즘을 제안하였다. 제안한 방법에서는 정합의 척도인 평균 절대 오차를 기준으로 각 블럭을 움직임이 적은 블럭, 중간정도인 블럭 혹은 큰 블럭으로 분류한 후 블럭의 특성에 따라 부표본화 및 탐색영역을 가변하여 움직임 추정을 수행한다. 컴퓨터 모의실험 결과 계산량 측면에서 제안방법이 기존 3단계 탐색 알고리즘 및 새로운 3단계 탐색 알고리즘과 비교해서 영상에 따라 조금 감소 혹은 조금 증가하였으나, 움직임이 빠른 영상에서는 움직임 벡터를 좀 더 정확하게 추정하여 PSNR이 증가됨을 확인할 수 있었다.

Abstract

We present a fast adaptive block matching algorithm using variable search area and subsampling to estimate motion vector more exactly. In the presented method, the block is classified into one of three motion categories: zero motion vector block, medium-motion block or high-motion block according to mean absolute difference of the block. By the simulation, the computation amount of the presented method comparable to three step search algorithm and new three step search algorithm. In the fast image sequence, the PSNR of our algorithm increased more than TSS and NTSS, because our algorithm estimated motion vector more accurately.

1. 서론

* 正會員, 永同工科大學校 電子工學部
(School of Electronic Engineering, YoungDong Institute of Technology)

** 正會員, 慶北大學校 電子工學科
(Department of Electronic Engineering, Kyungpook National University)

*** 正會員, 安東工業專門大學 電子通信科
(Department of Electronics and Communication, Andong Technical College)

接受日字: 1996年11月12日, 수정완료일: 1997年8月6日

통신 매체의 발달로 고속으로 정보를 전달하는 것이 가능하게 되었으며, 특히 여러 가지 정보 중에서 동영상 정보를 전송하기 위한 표준안으로서 MPEG이 소개되고 있다. 동영상은 대체로 움직임 보상 부호화(motion compensation coding)를 이용하여 높은 데이터 압축이 이루어진다. 이 방법에 의한 데이터의 압축은 이전 프레임을 이용하여 움직임 추정 및 보상을 수행하고, 이때 추정된 움직임 벡터(motion vector)에 의해서 보상된 영상과 원 영상과의 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.^{[11][12]} 블럭 정합 알고리즘은 화소 순환 알고리즘에 비하여 수행시간이 적게 소요되고 하드웨어 구현이 용이하기 때문에 움직임 보상 부호화에 널리 사용되고있다.

블럭 정합 알고리즘에서는 한 블럭내의 모든 화소들이 같은 방향으로 평행이동한다는 것을 가정하여 움직임 추정을 하는 방법으로서, 일정한 탐색범위에서 모든 화소에 대한 블럭을 순차적으로 비교해보는 전 탐색 블럭 정합 알고리즘 (full search block matching algorithm: FSBMA)을 기준으로 하고 있으며, 이는 최적의 움직임 벡터를 구할 수 있지만 계산량이 많은 단점을 갖는다. 이와 같은 단점을 줄이기 위해 탐색점의 수를 줄임으로써 움직임 벡터를 보다 고속으로 추정할 수 있는 방법들이 연구되었다.^{[11]-[12]} 대표적인 고속 BMA방법으로는 2차원 로그 탐색 (two dimensional logarithm search: 2-D LOG), 3단계 탐색 (three step search: TSS) 알고리즘, 교차 탐색 (cross search) 알고리즘, 새로운 3단계 탐색 알고리즘 (new three step search: NTSS) 등이 있다.^[11]

^[6] 3단계 탐색 알고리즘은 움직임 벡터를 추정하는 과정에서 국부최소에 빠지는 단점이 있으며, 새로운 3단계 탐색 알고리즘^[6]은 움직임이 적은 영상인 경우에는 비교적 정확하게 움직임 벡터를 추정하지만 움직임이 빠른 영상인 경우에는 전역최소(global minimum)에 빠져서 부정확한 움직임 벡터를 추정하는 단점이 있다. 이 방법들은 고속으로 움직임 벡터를 추정하기 위해서 전 탐색 블럭 정합 알고리즘보다 탐색점의 수를 줄여 근사 움직임을 찾은 후 미세 움직임을 단계적으로 찾는 것으로 전 탐색 블럭 정합 알고리즘 보다 계산량은 많이 줄일 수 있으나, 움직임 벡터의 추정이 부정확하게 되므로 물체의 보다 정확한 움직임을 표현하기 위한 방법의 도입이 필요하다.

본 논문에서는 움직임 벡터를 좀 더 정확하게 추정하면서 계산량을 감소시킬 수 있는 고속 적용 블럭 정합 알고리즘을 제안하였다. 제안한 방법에서는 정합의 척도인 평균 절대 오차 (mean absolute difference: MAD)를 기준으로 각 블럭을 움직임이 작은 블럭, 중간정도인 블럭 혹은 큰 블럭으로 분류한 후 각 블럭에 따라 움직임 벡터 추정을 행한다. 동영상의 배경부분과 같이 움직임이 거의 없는 블럭에서는 영 움직임 벡터

(zero motion vector), 움직임의 정도가 작은블럭 및 큰 블럭에서는 각각 탐색영역의 크기를 -5~3 및 -7~7로 한 후 이들을 부표본화 (subsampling) 하였다.

제안한 방법의 성능을 평가하기 위해서 세 가지 동영상에 대하여 컴퓨터 시뮬레이션을 수행하였다. 이 결과로부터 8×8 블럭에서 제안한 방법은 움직임 추정 방법중 최적인 전 탐색 블럭 정합 알고리즘에 비해 계산량을 84.97% 내지 90.02% 줄일 수 있으며, PSNR 측면에서도 비교적 적은 값이 감소되었다. 또한 계산량 측면에서, 제안방법이 기존 3단계 탐색 알고리즘 및 새로운 3단계 탐색 알고리즘보다는 영상에 따라 조금 감소 혹은 증가하였다. PSNR측면에서 제안방법이 3단계 탐색 알고리즘보다 0.20 dB 내지 1.83 dB로 증가되었으며, 또한 새로운 3단계 탐색 알고리즘과의 비교에서는 움직임이 빠른 영상인 FOOTBALL 및 FLOWER GARDEN에서는 각각 0.21 dB 및 0.93 dB 증가 되었기 때문에 좀 더 정확하게 움직임 벡터를 추정할 수 있는 효과적인 방법임을 확인할 수 있었다.

II. 블럭 정합 알고리즘을 이용한 움직임 벡터의 추정

블럭 정합 알고리즘은 임의의 작은 블럭으로 나눈 후 한 블럭내의 모든 화소는 동일 방향의 움직임을 갖는다고 가정하여 움직임 벡터를 추정하는 것으로, 이전 프레임에서 설정된 탐색영역에서 어떤블럭이 현재 프레임의 정해진 블럭으로 이동하였는가 찾는 것이다. N×N 화소를 갖는 블럭에서 프레임간 화소의 최대 움직임의 거리를 P화소라 가정할 때 상하 좌우로 최대 P화소 만큼의 움직임이 발생할 수 있으므로 탐색 영역의 크기는 (2P+N)×(2P+N)이며, 이 탐색영역에서 정합할 수 있는 탐색점의 수는 (2P+1)² 이다. 이때 서로 정합될 수 있는 두 블럭사이에서 왜곡의 정도를 나타내는 척도로는 평균절대오차 (mean absolute difference: MAD), 평균자승오차 (mean squared difference: MSD)가 널리 사용되고 있다.^{[11][12]} MAD 및 MSD는 각각

$$MAD(k, l) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |X_i(i, j) - X_{i-1}(i+k, j+l)| \quad (1)$$

$$MSD(k, l) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [X_t(i, j) - X_{t-1}(i+k, j+l)]^2 \quad (2)$$

이다. 여기서 $X_t(i, j)$ 는 현재 프레임, $X_{t-1}(i, j)$ 는 이전 프레임, (k, l) 은 탐색 영역에서의 탐색점이다. 각 탐색 점에 대해 MAD 혹은 MSD가 최소가 되는 (k, l) 을 움직임 벡터로 정의한다. MAD는 MSD에 비하여 계산량이 적고 하드웨어 구현이 용이하여 널리 이용하고 있지만, MSD보다 정확한 움직임 벡터를 탐색하지 못하는 단점을 지니고 있다. 이렇게 정합될 수 있는 모든 화소에 대하여 오차를 구하고 그 중 가장 작은 오차를 갖는 탐색점 (i, j) 의 값을 움직임 벡터로 결정하는 방법을 전 탐색 블록 정합 알고리즘이라 한다. 이 방법은 탐색영역내에서 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 블록 정합 알고리즘 즉, 2차원 로그 탐색 알고리즘, 3단계 탐색 알고리즘, 교차 탐색 알고리즘, 새로운 3단계 탐색 알고리즘 등이 연구되었다.^{[11]-[16]} 이 방법 중에서 간단하면서도 거의 정확한 움직임 벡터를 추정할 수 있는 3단계 탐색 알고리즘^[3]이 많이 사용되고 있다. 이는 움직임의 최대 변위가 -6~6인 경우에 대해 세 번의 탐색을 행함으로써 움직임 벡터를 추정할 수 있는 방법이다.

III. 제안한 고속 적응 블록 정합 알고리즘

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

본 논문에서는 기존 전 탐색 블록 정합 알고리즘에서 요구되는 많은 계산량 및 3단계 탐색 알고리즘 및 새로운 3단계 탐색 알고리즘의 문제점인 움직임 벡터

의 부정확한 추정을 개선할 수 있는 방법이다. 제안방법에서는 영상에서 블록의 움직임 정도를 판단하여 적당한 탐색영역을 할당한 후 부표본화를 행하여 계산량 및 부정확한 움직임 벡터를 개선할 수 있는 고속 적응 블록 정합 알고리즘을 제안한다.

제안한 고속 적응 블록 정합 알고리즘의 흐름도는 그림 1과 같다. 이 그림에서 움직임 정도의 척도는 $MAD(0,0)$ 를 이용한다. $MAD(0,0) < th_1$ 이면 움직임이 거의 없는 블록, $th_1 \leq MAD(0,0) < th_2$ 이면 움직임이 중간정도인 블록, 그밖에서는 움직임이 큰 블록으로 분류한 후 각 블록에 대해서 움직임 벡터를 추정한다. 여기서, th_1 및 th_2 는 문턱값을 나타낸다.

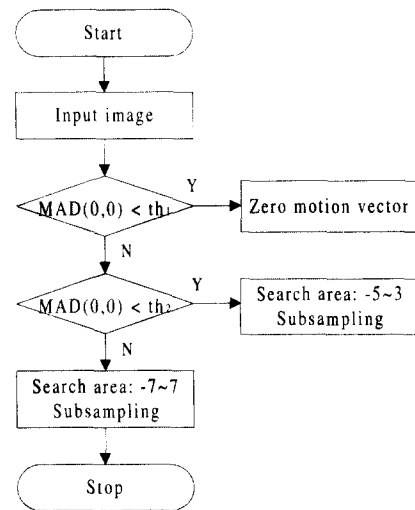


그림 1. 제안한 고속 적응 블록 정합 알고리즘 흐름도
Fig. 1. The flow chart of the proposed fast adaptive BMA.

먼저, $MAD(0,0) < th_1$ 인 블록은 움직임이 작은 블록으로서 동영상에서 배경 (background) 및 움직임이 적은 전경 (foreground)으로 구성된다. 배경 영역은 동영상 중에서 대부분 차지하고 있으며, 이 영역에서는 프레임이 변화되어도 움직임이 거의 발생되지 않는다. 또한 전경부분 중에서 프레임이 변해도 움직임의 변화가 적게 나타나는 영역들이 많이 포함되어 있다. 이와 같이 움직임의 정도가 거의 없는 영역에서는 움직임 벡터를 추정하고자 하는 t 프레임에 있는 블록 (i, j) 이 이전 프레임에서 정해진 탐색영역에 있는 탐색점 전부에 대하여 평균절대오차를 계산하여 움직임 벡터를 추정할 필요가 없다. 그러므로 이 영역에서는 이

전 프레임 $t-1$ 의 (i,j) 에 있는 블럭을 움직임 벡터로 추정하여 이를 움직임이 없는 영 움직임 벡터 (수평방향으로의 움직임 0, 수직방향으로의 움직임 0)로 정하게 되어 움직임 벡터를 추출하는데 사용되는 계산량이 필요없다. 그러나 전 탐색 블럭 정합 알고리즘에서는 블럭크기가 8×8 (탐색영역 $-7 \sim 7$) 및 16×16 (탐색영역 $-7 \sim 7$) 경우에서 225개 블럭의 계산량을 필요로 하므로 제안방법에서 계산량이 매우 감소된다. 또한 성능면에서도 배경 혹은 움직임이 거의 없는 블럭에서만 적용이 되므로 전 탐색 블럭 정합 알고리즘과 비교했을 때 오차가 거의 나지 않는다.

다음으로, $MAD(0,0) > th_2$ 인 블럭에서는 움직임이 큰 블럭이므로 좀 더 정확한 움직임 벡터를 추정하기 위해서 2 단계로 움직임 벡터를 추정한다. 1 단계로는 그림 2에서와 같이 탐색영역의 크기를 $-7 \sim 7$ 로 하고 탐색점을 부표분화한다.

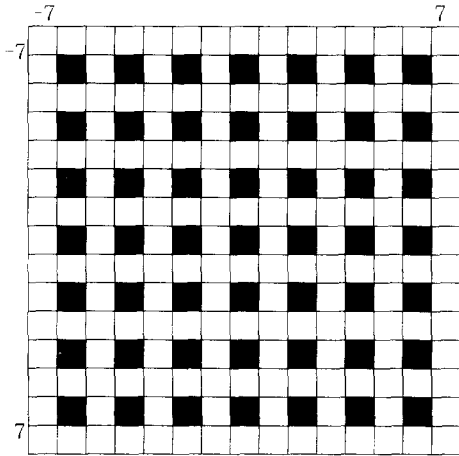


그림 2. 탐색 영역에서 부표분화 형태
Fig. 2. Pattern of subsampling in search area.

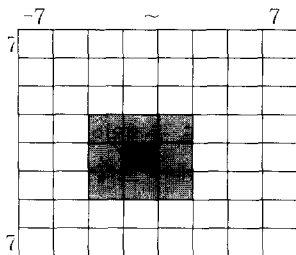


그림 3. 인근 8개 화소를 탐색하기 위한 형태
Fig. 3. Pattern of 8 pixel neighbourhood for search.

이 그림에서 탐색점으로는 검은점이 사용되고, 흰 점

은 사용되지 않는다. 이와 같이 부표분화 된 각 탐색점에 대하여 평균절대오차를 계산하여 가장 작은 오차를 갖는 탐색점 혹은 $MAD < th_1$ 를 갖는 탐색점을 찾는다. 이렇게 하여 움직임 벡터로 결정할 경우에는 부정확한 움직임 벡터이므로 많은 오차가 발생된다. 따라서 더욱 정확한 움직임 벡터를 추정하기 위해 2 단계에서는 그림 3에서와 같이 1 단계에서 결정된 탐색점(검은점)을 중심으로 인접하는 8개의 화소에 대해서 각각 평균절대오차를 계산하여 가장 정합이 좋은 블럭 혹은 $MAD < th_1$ 를 갖는 탐색점을 찾는다. 이렇게 1 및 2 단계를 수행한 후 정합이 좋은 탐색점을 구하여 이를 움직임 벡터로 정한다.

마지막으로, 움직임이 중간정도인 블럭 즉, $th_1 \leq MAD(0,0) < th_2$ 에서는 위의 움직임이 큰 블럭에서 움직임 벡터를 추정하는 방법과 같으면서, 단지 차이 나는 것은 1단계에서 탐색영역을 $-5 \sim 3$ 로 줄여서 부표분화하여 탐색을 시작한다.

IV. 실험 결과 및 고찰

본 논문에서 제안한 고속 적응 블럭 정합 알고리즘의 성능을 평가하기 위하여 컴퓨터 모의 실험을 수행하였다. 본 실험에서는 720×480 크기의 TABLE TENNIS, FOOTBALL 및 FLOWER GARDEN 각 30, 60 및 60 프레임을 사용하였다. FOOTBALL은 카메라 및 물체의 움직임 빠른 영상, FLOWER GARDEN은 정지된 물체에 카메라가 아주 빠르게 움직이는 영상, TABLE TENNIS은 정지된 카메라를 이용하여 많은 배경을 가지면서 비교적 움직임이 적은 영상이다. 제안방법에서는 블럭 크기를 8×8 및 16×16 , 움직임의 정도에 따라 탐색범위를 $-7 \sim 7$ 및 $-5 \sim 3$ 로 하여 부표분화를 하였으며, 또한 영 움직임 벡터를 적용하였다. 기존 전 탐색 블럭 정합 알고리즘, 3 단계 탐색 알고리즘 및 새로운 3 단계 탐색 알고리즘에서는 블럭 크기를 8×8 및 16×16 , 탐색범위 $-7 \sim 7$ 로, 정합의 척도는 평균절대오차를 사용하여 모의실험을 하였다. 본 모의실험에서는 기존의 전 탐색 블럭 정합 알고리즘, 3 단계 탐색 알고리즘, 새로운 3 단계 탐색 알고리즘^[6] 및 제안한 방법에 대한 결과를

$$PSNR = 10 \log \frac{255^2}{\sigma_e^2} \text{ [dB]}$$

을 이용하여 평가하였다. 여기서, σ_e^2 은 원 영상과 움

직접 보상된 영상의 평균 자승 오차이다. 또한 움직임이 거의 없는 블록으로 판단하는 문턱값 th_1 및 움직임이 중간과 큰 블록으로 구분하는 문턱값 th_2 를 결정하기 위해서 th_1 을 4, 5 및 6, th_2 를 17, 19 및 21에 대해서 각각 모의실험을 수행하여 PSNR 및 계산량을 비교하여, 이 중에서 적당한 $th_1=5$ 및 $th_2=19$ 로 하였다. 이들 각 방법에 대하여 프레임 변화에 따른 PSNR 결과는 그림 4 및 5에서와 같다. 그리고 각 영상들에 대한 평균 PSNR 결과는 표 1과 같다.

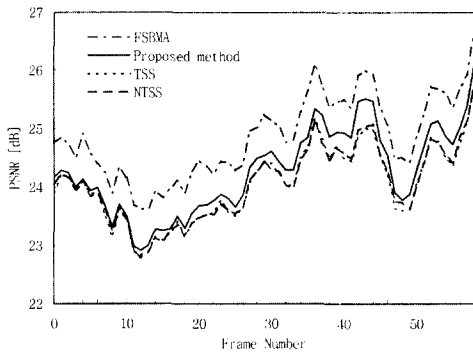


그림 4. FOOTBALL 영상의 블록크기 8×8 (탐색범위: -7~7)에 대한 PSNR
 Fig. 4. PSNR of block size 8×8 (search area: -7~7) for FOOTBALL image .

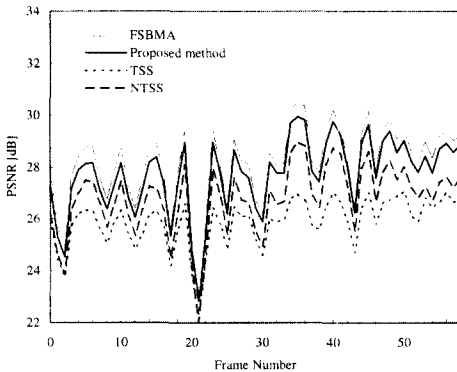


그림 5. FLOWER GARDEN 영상의 블록크기 8×8 (탐색범위: -7~7)에 대한 PSNR
 Fig. 5. PSNR of block size 8×8 (search area: -7~7) for FLOWER GARDEN image.

이 표에서 블록크기 8×8인 경우, 기존 전 탐색 블록 정합 알고리즘보다 제안한 방법에서는 0.35 dB 내지 0.59 dB 감소되지만, 고속의 블록 정합 알고리즘인 3 단계 알고리즘보다는 0.20 dB 내지 1.83 dB 증가하

였다. 그리고 블록크기 16×16인 경우, 기존의 전 탐색 블록 정합 알고리즘보다 제안한 방법에서 0.30 dB 내지 0.37 dB 감소하였다.

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

Block size	Image	FSBMA	Propose method	TSS	NTSS
8×8	FOOTBALL	24.86	24.27	24.04	24.06
	FLOWER GARDEN	28.14	27.72	25.89	26.79
	TABLE TENNIS	26.37	26.02	25.82	26.00
16×16	FOOTBALL	23.30	23.00	22.93	22.83
	FLOWER GARDEN	27.26	26.89	26.08	26.78
	TABLE TENNIS	25.83	25.52	25.34	25.53

블록크기에 대한 결과에서, 먼저 전 탐색 블록 정합 알고리즘과 제안한 방법을 비교해 보면 블록크기가 8×8 보다 16×16에서 제안한 방법이 전 탐색 블록 정합 알고리즘에 근사되는 것을 알 수 있다. 이는 블록크기가 크기 때문에 각 영상에서의 물체의 움직임이 이 탐색영역을 거의 벗어나지 않기 때문에 부표본화를 하더라도 국부최소에 빠지는 경우가 거의 없게되어 물체의 움직임을 거의 정확하게 따라가기 때문이다. 또한, 8×8 블록크기에서 고속의 블록 정합 알고리즘인 기존의 3단계 탐색 알고리즘과 제안방법을 PSNR 측면에서 비교하면 제안한 방법이 TABLE TENNIS, FOOTBALL 및 FLOWER GARDEN 각 영상에 대하여 0.20 dB, 0.23 dB 및 1.83 dB 증가하였다. 이 결과에서 제안방법의 PSNR이 증가한 이유는 먼저, 움직임이 중간인 블록의 경우, 3단계 탐색 알고리즘에서는 1단계로 탐색점의 개수가 9개 이지만 제안방법에서는 16개를 탐색점으로 하기 때문에 국부 최소에 빠질 가능성이 줄어들었으며, 또한 움직임이 큰 블록인 경우 제안방법에서 1단계의 탐색점이 49개를 선택하기 때문에 3단계 탐색 알고리즘보다 국부 최소에 빠질 가능성이 거의 없어서 움직임 벡터를 좀 더 정확하게 추정할 수 있기 때문이다. 그리고 고속의 블록 정합 알고리즘인 새로운 3단계 탐색 알고리즘⁽⁶⁾과 제안한 방법을 PSNR 측면에서 비교하면 8×8 블록크기에서 제안한 방법이 TABLE TENNIS, FOOTBALL 및 FLOWER GARDEN 각 영상에

대하여 0.02 dB, 0.21 dB 및 0.93 dB 증가하였다. 이 결과에서 TABLE TENNIS 영상과 같이 움직임이 거의 없는 경우에는 기존의 새로운 3단계 탐색 알고리즘과 성능이 거의 비슷하다. 그러나, FOOTBALL 및 FLOWER GARDEN과 같은 움직임이 빠른 영상인 경우에는 제안한 방법이 좀 더 정확하게 움직임 벡터를 추출하여 PSNR이 증가되었다. 이는 기존 새로운 3단계 탐색 알고리즘에서는 움직임이 많은 영상에서 전역최소(global minimum)에 빠지는 경우가 발생하여 부정확한 움직임 벡터를 추출하였기 때문에 오차가 증가되었다. 또한 움직임이 빠른 FOOTBALL 영상의 블럭크기 16×16 에서 3단계 탐색 알고리즘이 새로운 3단계 탐색 알고리즘보다 PSNR이 증가되었다. 이 결과로부터 움직임이 빠른 영상인 경우에는 새로운 3단계 탐색 알고리즘이 부정확한 움직임 벡터를 추정하는 단점이 있다.

기존의 전 탐색 블럭 정합 알고리즘, 3단계 탐색 알고리즘, 새로운 3단계 탐색 알고리즘 및 제안한 방법에 대한 계산량을 비교한 결과는 표 2와 같다.

표 2. 각 방법에 대한 % 계산량
Table 2. The percent computation amount for each method.

Block size	Algorithms	FOOTBALL	FLOWER GARDEN	TABLE TENNIS
8×8	FSBMA	100.0	100.0	100.0
	Proposed method	9.98	15.03	10.18
	TSS	11.11	11.11	11.11
	NTSS	11.29	13.05	9.46
16×16	FSBMA	100.0	100.0	100.0
	Proposed method	11.17	16.06	10.81
	TSS	11.11	11.11	11.11
	NTSS	11.26	12.96	9.28

이 표에서, 각 방법에 대한 계산량은 평균절대오차의 % 계산량 (덧셈, 뺄셈, 절대값)이다. NTSS 및 제안 방법에서의 비교연산자의 계산량은 상대적으로 너무 적은 양이기 때문에 무시하였다. 블럭 크기 8×8 인 경우 기존 전 탐색 블럭 정합 알고리즘과 제안한 방법을 비교해 보면 FLOWER GARDEN, FOOTBALL 및 TABLE TENNIS의 영상에서 각각 84.97%, 90.02% 및 89.82% 감소되었으며, 블럭 크기 16×16 인 경우 83.94%, 88.83% 및 89.19% 감소되었다. 이와 같이 계산량이 감소되는 이유는 대부분의 영상에서는 배경

부분 및 움직임이 적은 블럭이 많이 존재하여 탐색 영역의 크기를 줄이고 부표본화하여 움직임 벡터를 추정하기 때문이다. 그리고 제안한 방법과 3단계 탐색 알고리즘의 계산량을 비교하면 영상에 따라 조금 감소 혹은 증가하였다.

이상의 결과로부터 제안한 방법이 움직임 추정 방법 중 최적인 전 탐색 블럭 정합 알고리즘보다 계산량을 많이 줄일 수 있으며, PSNR 측면에서도 비교적 작게 감소되었다. 기존 고속 방법인 3단계 탐색 알고리즘 및 새로운 3단계 탐색 알고리즘보다 계산량 측면에서는 영상에 따라 조금 감소 혹은 증가하였으며, 또한 움직임이 빠른 FOOTBALL 및 FLOWER GARDEN 영상에서는 기존 고속 방법보다 PSNR이 증가되었기 때문에 좀 더 정확하게 움직임 벡터를 추정할 수 있는 효과적인 방법임을 확인할 수 있었다.

V. 결론

본 논문에서는 움직임 벡터를 좀 더 정확하게 추정하면서 계산량을 감소시킬 수 있는 고속 적응 블럭 정합 알고리즘을 제안하였다. 제안한 방법에서는 정합의 척도인 평균 절대 오차 (mean absolute difference: MAD)를 기준으로 각 블럭을 움직임이 작은 블럭, 중간정도인 블럭 혹은 큰 블럭으로 분류한 후 각 블럭에 따라 움직임 벡터 추정을 행한다. 동영상의 배경부분과 같이 움직임이 거의 없는 블럭에서는 영 움직임 벡터 (zero motion vector), 움직임의 정도가 작은 블럭 및 큰 블럭에서는 각각 탐색영역의 크기를 $-5 \sim 3$ 및 $-7 \sim 7$ 로 한 후 이들을 부표본화 (subsampling) 하였다.

제안한 방법의 성능을 평가하기 위해서 세 가지 동영상에 대하여 컴퓨터 시뮬레이션을 수행하였다. 이 결과로부터 8×8 블럭에서 제안한 방법은 움직임 추정 방법중 최적인 전 탐색 블럭 정합 알고리즘에 비해 계산량을 84.97% 내지 90.02% 줄일 수 있으며, PSNR 측면에서도 비교적 적은 값이 감소되었다. 또한 계산량 측면에서, 제안방법이 기존 3단계 탐색 알고리즘 및 새로운 3단계 탐색 알고리즘보다는 영상에 따라 조금 감소 혹은 증가하였다. PSNR 측면에서 제안방법이 3단계 탐색 알고리즘보다 0.20 dB 내지 1.83 dB로 증가되었으며, 또한 새로운 3단계 탐색 알고리즘과의 비교에서는 움직임이 빠른 영상인 FOOTBALL 및

FLOWER GARDEN에서는 각각 0.21 dB 및 0.93 dB 증가 되었기 때문에 좀 더 정확하게 움직임 벡터를 추정할 수 있는 효과적인 방법임을 확인할 수 있었다.

참 고 문 헌

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

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

[3] 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.

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

[5] 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, Apr 1993.

[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.

저 자 소 개

申 容 達(正會員)

1984년 2월 경북대학교 전자공학과 졸업 (공학사), 1986년 2월 경북대학교 대학원 전자공학과 졸업 (공학석사), 1994년 8월 경북대학교 대학원 전자공학과 졸업 (공학박사). 1991년 3월~1996년 2월 경주전문대학 전자과 전임강사, 조교수. 1996년 3월 ~ 현재 영동공과대학교 전자공학부 조교수. 주관심분야는 영상신호처리, 영상압축 및 코딩.



李 承 珍(正會員)

1994년 2월 경북대학교 전자공학과 졸업 (공학사), 1996년 2월 경북대학교 대학원 전자공학과 졸업 (공학석사), 1996년 3월 ~ 현재 경북대학교 대학원 전자공학과 박사과정 재학중. 주관심분야는 영상신호처리, 영상압축 및 코딩.



金 慶 圭(正會員)

1993년 2월 경북대학교 전자공학과 졸업 (공학사), 1995년 2월 경북대학교 대학원 전자공학과 졸업 (공학석사), 1995년 3월 ~ 현재 경북대학교 대학원 전자공학과 박사과정 재학중. 주관심분야는 영상신호처리, 영상압축 및 코딩.

鄭 元 植(正會員)

第 33 卷 B編 第 6號 參照

현재 경북대학교 대학원 전자공학과 박사과정 재학중

金 榮 春(正會員)

第 33 卷 B編 第 6號 參照

李 奉 洛(正會員)

第 30卷 B編 第 12號 參照

현재 안동공업전문대학 전자통신과 조교수

蔣 鐘 國(正會員)

第 33 卷 B編 第 4號 參照

현재 영동공과대학교 전자공학부 전임강사

李 健 一(正會員)

第 33卷 B編 第 4號 參照

현재 경북대학교 전자·전기공학부 교수