

論文99-36T-9-6

고속 탐색 알고리즘을 이용한 블록정합 움직임 추정

(Block Matching Motion Estimation Using Fast Search Algorithm)

吳泰明 *

(Tae-Myung Oh)

요 약

본 논문에서는 SEA (successive elimination algorithm) 기반의 고속 블록정합 움직임 추정 알고리즘을 제시한다. 제안하는 기법은 블록 움직임 벡터가 탐색영역의 중앙에 편향되어 분포한다는 특성에 기반하여 기존의 SEA보다 탐색영역에서 탐색점의 수를 줄임으로써 성능을 향상시키는 기법이다. 또한 더 많은 연산량을 줄이기 위하여 제안된 기법은 블록정합에서 화소를 비교하는 연산량을 줄이는 RBMAD (reduced bits mean absolute difference) 정합기준과 블록정합에서 사용되는 화소의 수를 줄이는 부표본화 기법과 결합하였다. 제안된 기법은 완전탐색 (full-search, FS) 블록정합 알고리즘과 비슷한 성능을 유지하면서 기존의 고속 알고리즘보다 성능이 향상되었음을 실험을 통하여 확인하였다.

Abstract

In this paper, we present a fast block matching motion estimation algorithm based on successive elimination algorithm (SEA). Based on the characteristic of center-biased motion vector distribution in the search area, the proposed method improves the performance of the SEA with a reduced the number of the search positions in the search area. In addition, to reduce the computational load, this method is combined with both the reduced bits mean absolute difference (RBMAD) matching criterion which can be reduced the computation complexity of pixel comparison in the block matching and pixel decimation technique which reduce the number of pixels used in block matching. Simulation results show that the proposed method provides better performance than existing fast algorithms and similar to full-search block motion estimation algorithm.

I. 서 론

동영상 압축 부호화의 표준인 H.261이나 MPEG에서는 동영상의 시공간적인 정보의 중복성을 제거하여 고압축을 실현하는데 있어서 움직임 추정 및 보상은 가장 중요한 역할을 담당하고 있다. 뿐만 아니라 카메라

를 위한 영상 안정화 시스템, 비전 시스템 등과 같이 동영상을 처리하는 거의 모든 시스템들에서도 움직임 추정 기법은 핵심기술로 자리하고 있다.

움직임 추정에는 전체 화면을 몇 개의 블록으로 나누어 블록단위로 이동벡터를 찾아내는 블록정합 알고리즘이 가장 많이 알려져 있다. 특히 MAD를 이용한 완전탐색 블록정합 알고리즘은 정확도에서 가장 우수한 성능을 가지고 있다. 그러나 완전탐색 알고리즘은 방대한 양의 계산을 필요로 하기 때문에 처리시간이 길어지고, 하드웨어로 구현하였을 때 회로가 복잡해진다는 문제점을 안고 있다. 탐색영역을 줄임으로써 이와

* 正會員, 明知專門大學 電子科

(Myongji College, Dept. of Electronics Eng.)

接受日字 : 1999年8月25日, 수정완료일 : 1999年9月14日

같은 문제점을 해결하기 위하여 삼단탐색 (three-step search, 3SS), 로그스케일탐색 (logarithmic search) 등과 같은 많은 고속 알고리즘들이 발표되었고^[1,4-5], 탐색 과정에서 탐색영역내의 모든 화소값들을 비교할 수 있다는 특성을 갖고 있는 삼단탐색 기법이 가장 널리 알려져 있다. 블록의 움직임 벡터의 분포가 탐색원점에 편향되어 있다는 특성을 이용하여 삼단탐색의 성능을 향상시키는 고속 알고리즘도 발표되었다^[6-7]. 그러나 실시간 처리를 위해 요구되어지는 계산량과 VLSI로 구현할 경우 비용의 최소화 등을 고려한다면 여전히 많은 어려움이 있다.

본 논문에서는 고속 블록정합 움직임 추정을 위하여 RBMAD을 이용하여 움직임을 추정하고, SEA를 이용하여 탐색영역을 줄이면서 동시에 탐색위치에 따라 부표본화된 영상패턴을 이용함으로써 고속으로 블록정합 움직임 추정을 수행하는 알고리즘을 제안한다. 제안된 기법에서는 기존의 MAD을 이용한 블록정합 과정에서 영상의 각 화소에서 4개의 상위 비트들만을 사용하여 블록정합을 수행함으로써 하드웨어로 구현이 용이하고 처리속도를 향상시킬 수 있다. 또한 현재 프레임의 기준블록과 이전 프레임의 탐색영역 내에서 후보블록간의 차이에 대한 삼각 부등식의 조건을 이용하여 탐색점의 수를 줄이는 SEA를 사용한다. 그리고 본 논문에서는 블록내의 모든 화소를 활용하면서 탐색위치에 따라 부표본화된 영상패턴을 반복적으로 사용하여 블록정합 과정에서 보다 많은 계산량을 줄이는 기법을 이용한다.

본 논문에서는 제안된 기법의 성능을 평가하기 위하여 동영상으로부터 블록 움직임 벡터를 검출하는 실험을 수행하였다. 먼저 II장에서는 기존의 블록정합 움직임 추정을 소개하고, SEA와 부표본화된 영상패턴을 이용한 고속 탐색기법은 III장에서 소개한다. IV장에서는 동영상에서 제안된 기법을 이용한 실험결과를 제시하고, 기존방식과의 비교를 통하여 성능을 평가한다. V장에서는 결론을 맺는다.

II. 블록정합 움직임 추정

기존의 대표적인 움직임 추정 기법인 BMA (block matching algorithm)는 일정 크기의 블록 내의 화소들이 회전 혹은 확대·축소되지 않고 동일하게 이동만

하였다는 가정 하에서 블록 당 하나의 움직임 벡터를 찾는 것이다. 본 장에서는 기존의 완전탐색 블록정합 알고리즘을 소개하고, 실시간 처리를 위해 요구되어지는 계산량과 VLSI로 구현될 경우 비용의 최소화 등을 고려한 고속 블록정합 움직임 추정기법에 대해서 살펴보고자 한다.

1. 완전탐색 블록정합 알고리즘

블록 움직임 벡터는 현재 프레임의 기준블록과 이전 프레임에서 상관계수가 가장 큰 후보블록 사이의 상대적인 위치로 정의된다. 그리고 MAD를 이용한 완전탐색 블록정합 알고리즘은 정확도에서 가장 우수한 성능을 가지고 있다. 그림 1은 완전탐색 블록정합 과정을 보여주고 있다.

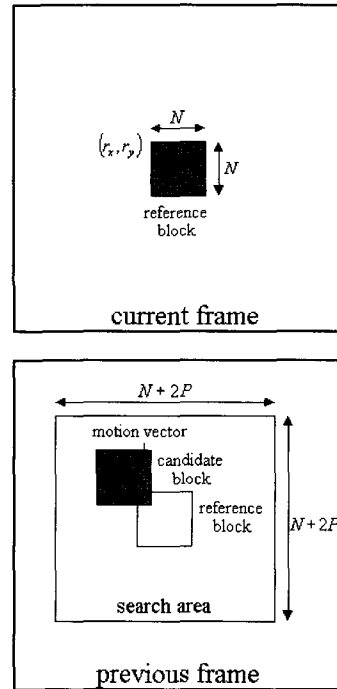


그림 1. BMA를 이용한 움직임 추정
Fig. 1. Motion estimation using BMA.

그림 1에서 현재 프레임 f' 내의 (r_x, r_y) 에 위치한 $N \times N$ 크기의 기준블록의 움직임을 이전 프레임 f'^{-1} 에서 $\pm P$ 화소만큼의 범위에서 추정한다고 가정하자. 그러면 이전 프레임 내의 움직임 추정 영역의 크기는 $(N + 2P) \times (N + 2P)$ 가 되고, 여기에서 움직임 벡터의 후보가 될 수 있는 총 $(2P + 1)^2$ 개의 위치에서 상

관계수를 모두 계산한 후 최대 상관도를 보여주는 위치를 움직임 벡터로 결정한다. 이때 현재 프레임에서의 기준블록 X 와 이전 프레임에서의 후보블록 Y 간의 차이는 다음과 같이 MAD 혹은 l_1 metric으로 불리는 정합기준이 주로 사용된다.

$$\|X - Y(m, n)\|_1 = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N |f'(r_x + i, r_y + j) - f^{t-1}(r_x + i + m, r_y + j + n)| \quad (1)$$

여기서 (r_x, r_y) 은 프레임 상에서의 기준블록의 위치이고, (m, n) 은 탐색영역 내에서의 위치를 가리킨다. 따라서 최종적으로 블록의 움직임 벡터는 다음과 같이 기준블록과 후보블록의 평균 절대값 차이가 가장 작은 값을 갖는 탐색영역의 위치 (m, n) 으로 결정된다.

$$V = \arg \min \{ \|X - Y(m, n)\|_1, -P \leq m, n \leq P \} \quad (2)$$

2. 고속 블록정합 알고리즘

블록정합 알고리즘에서 완전탐색 알고리즘은 정확도에서 가장 우수한 성능을 가지고 있지만, 방대한 양의 계산을 필요로 하기 때문에 처리시간이 길어지고, 하드웨어로 구현하였을 때 회로가 복잡해진다는 단점을 지닌다. 이를 해결하기 위해 탐색점의 수를 줄이는 삼단탐색, 로그스케일탐색 등과 같은 고속 알고리즘들이 발표되었고^[4-7], 정합기준의 최소값으로 찾아가는 수렴속도에 따라 탐색영역의 크기를 가변적으로 조절함으로써 삼단탐색의 성능을 향상시키는 고속 알고리즘들도 발표되었다^[8-9]. 또한 이웃한 블록들간의 높은 상관도를 이용하여 예측된 움직임 벡터를 이용하는 방법들도 널리 알려져 있다^[3,8]. 지금까지 이러한 고속 알고리즘들은 MAD를 이용한 블록정합 알고리즘에 많이 이용되어 왔지만, 실제 비디오 영상에서 움직임 추정의 정확성은 완전탐색 기법과 다소 차이가 있으며, 실시간 처리를 위해 요구되어지는 계산량과 VLSI로 구현할 경우 비용의 최소화 등을 고려한다면 여전히 어려움이 있다. 본 논문에서는 블록정합에서 화소를 비교하는 연산량을 줄이기 위한 RBMAD 정합기준^[2]과 완전탐색의 성능을 유지하면서 탐색영역에서 연속적으로 탐색점의 수를 줄이는 SEA^[1], 그리고 블록정합에서 사용되는 화소를 줄이기 위한 부표본화 기법^[3]을 결합한 고속 알고

리즘을 제안한다. 먼저 각각의 기법들의 방법과 특성을 소개하기로 한다.

1) RBMAD을 이용한 블록정합 알고리즘

기존의 블록정합 움직임 추정기법은 블록정합에 필요한 많은 산술연산들을 처리해야 하기 때문에 블록정합 과정에서 화소의 크기를 줄임으로써 처리시간을 줄이기 위한 다양한 기법들이 제시되었다^[2,10-12]. 특히 그레이 영상으로부터 에지를 추출하여 이를 이진화한 후, 이로부터 영상의 움직임을 추정하는 고속 블록정합 기법들이 제시된바 있으나, 에지 추출을 위한 별도의 계산이 필요하며 에지 추출의 정확도에 따라 움직임의 성능이 크게 달라지는 문제점이 발생한다. 따라서 본 논문에서는 기존의 MAD에서 사용되는 화소의 비트수를 줄임으로써 하드웨어의 복잡도를 줄이고 동시에 처리속도를 높이는 RBMAD를 이용하여 블록정합 움직임 추정을 수행한다. 먼저 그레이 영상으로부터 RBMAD를 이용하여 블록정합 움직임을 추정하는 방식을 제시한다.

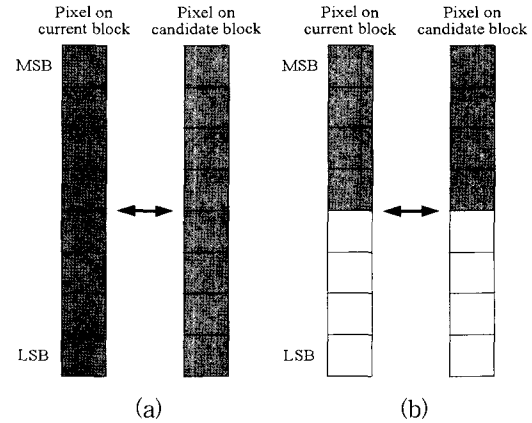


그림 2. 블록정합에서 화소비교

(a) MAD (b) RBMAD

Fig. 2. Pixel comparison in block matching.

(a) MAD (b) RBMAD

2^{K+1} 의 그레이 레벨 해상도와 $L \times L$ 공간 해상도를 갖는 동영상을 f 로 하고, 시각 t 에서의 현재 프레임과 그 이전 프레임을 각각 f^t 와 f^{t-1} 로 나타내기로 하면, 현재 프레임내의 (i, j) 번째 화소는 다음과 같이 $K+1$ 비트로 표현된다.

$$f^t(i, j) = a_K 2^K + a_{K-1} 2^{K-1} + \dots + a_1 2^1 + a_0 \quad (3)$$

이 때 각 화소를 이루는 $K + 1$ 비트의 데이터들 중에서 최상위 비트로부터 k 개의 비트들만을 모아서 만들어진 영상 f_k' ($0 < k \leq K + 1$)을 생성할 수 있다. 그림 2에서는 f_4' 와 $f_4'^{-1}$ 인 영상을 생성하여 블록정합의 예를 보였다. 그림 2에서 보인 RBMAD를 이용한 블록정합 움직임 추정기법이 갖는 가장 큰 장점은 RBMAD를 이용하여 하드웨어로 구현하였을 경우 기존 방식보다 단순해진다는 것이다. 만약 k 가 전체 비트수의 절반값을 갖는다면 하드웨어 크기도 거의 절반 정도의 크기로 줄어든다^[2]. 그리고 일반적으로 블록정합 과정에서 화소간의 비교는 비트 시리얼 연산(bit-serial operation)으로 수행된다^[2]. 따라서 RBMAD를 이용하였을 경우 기존 방식보다 감소된 비트 수만큼 처리속도를 향상시킬 수 있다.

표 1은 블록기반 움직임 추정에서 블록정합에 사용되는 비트 수에 따른 RBMAD와 기존의 MAD를 실험 영상에 적용하여 각각 블록정합 움직임 추정을 수행하고, 추정된 블록 움직임 벡터를 이용하여 보상된 영상간의 평균 PSNR 값을 보여주고 있다. 표 1에서 상위 4개의 비트를 이용한 RBMAD 방식이 기존의 MAD 방식과 비교하여 MSE와 PSNR에서 각각 2.0과 0.05 dB 이하의 오차를 보이고 있다. 따라서 본 논문에서는 기존 MAD와 움직임 추정 성능이 거의 비슷한 상위 4개의 비트를 이용한 RBMAD ($k = 4$)를 이용한다.

표 1. 실험 영상에 대한 정합 기준들의 성능 비교

Table 1. Performance comparison of the matching criteria for test video sequences.

Matching criteria	Table Tennis		Football	
	average MSE	average PSNR (dB)	average MSE	average PSNR (dB)
MAD	90.38	28.73	394.21	22.26
RBMAD (k=7)	90.40	28.73	394.23	22.26
RBMAD (k=6)	90.45	28.73	394.30	22.26
RBMAD (k=5)	90.53	28.72	394.59	22.26
RBMAD (k=4)	91.12	28.70	395.70	22.25
RBMAD (k=3)	94.05	28.55	397.80	22.23
RBMAD (k=2)	103.72	28.12	419.96	21.99
RBMAD (k=1)	145.50	26.69	476.49	21.44
MiniMax	111.03	27.85	511.88	21.13

2) SEA를 이용한 고속 움직임 추정

SEA는 완전탐색 블록정합 알고리즘으로 추정된 움직임 벡터와 동일한 움직임 벡터를 찾으면서, 동시에 블록정합을 수행해야 할 탐색점의 수를 줄이는 고속 탐색기법이다. SEA 탐색기법은 블록 움직임 추정을 위하여 현재 프레임에서의 기준블록 X_k 와 이전 프레임에서의 후보블록 $Y_k(m, n)$ 간의 블록정합 과정에서 다음과 같이 삼각 부등식을 이용한다.

$$| \|X_k\|_1 - \|Y_k(m, n)\|_1 | \leq \|X_k - Y_k(m, n)\|_1 \quad (4)$$

완전탐색은 탐색영역에서 모든 (m, n) 위치에서 현재 프레임의 기준블록과 이전 프레임 후보블록간의 정합이 이루어지고, 그 중에서 평균 절대 차이값이 최소인 위치가 블록의 움직임 벡터가 된다. 이때 현재 탐색점 (m', n') 에서 블록정합이 수행되고, 그 결과가 이전 탐색점에서 얻어진 값보다 작다면 (m', n') 은 후보 벡터가 된다. 이 과정을 탐색영역의 모든 위치가 비교될 때까지 수행한 후 얻어진 후보벡터는 최종적으로 블록의 움직임 벡터가 된다. 그러나 후보벡터 (m', n') 과 그 이후에 비교될 탐색점 (m, n) 에서 두 블록간의 차이가 식 (4)를 만족하지 않는다면, 즉 $|\|X_k\|_1 - \|Y_k(m, n)\|_1| > \|X_k - Y_k(m', n')\|_1$ 하면 (m, n) 위치에서는 블록정합을 수행하지 않고 다음 탐색위치로 넘어간다. 그리고 $\|X_k\|_1$ 와 $\|Y_k(m, n)\|_1$ 는 탐색영역 전체에 대하여 한번만 계산되면 되므로 매우 적은 계산량으로 블록 움직임 벡터를 찾을 수 있다.

3) 화소간측 방법을 이용한 고속 움직임 추정

블록정합을 수행하는 과정에서 부표본화된 영상패턴을 이용함으로써 연산량을 줄일 수 있다^[3]. 이 기법은 블록정합에서 특정한 부표본화된 영상패턴만을 모든 탐색영역에 적용하는 것이 아니라 탐색위치에 따라 부표본화된 영상패턴을 규칙적으로 번갈아 사용함으로써 연산량을 줄이고 있다. 그림 3은 그레이 영상에서 4:1로 부표본화된 영상패턴을 이용하여 블록의 움직임을 추정하는 과정을 보여주고 있다.

그림 3(a)는 4개의 영상패턴으로 분류한 하나의 블록을 보여주고 있고, 그림 3(b)는 탐색위치에 따라 부표본화된 영상패턴을 선택적으로 적용하는 것을 보여주고 있다. 즉, 그림 3(b)의 "1" 탐색위치에서는 그림 3(a)에서 "a"로 구성된 영상패턴을 이용하고, "2"의 탐색위

치에서는 "b"로 구성된 영상패턴을 이용한다. 마찬가지로 탐색위치 "3", "4"에서도 각각 "c", "d"로 구성된 영상패턴을 이용하여 블록의 움직임을 추정하고 난 후, 4개의 움직임 블록 움직임 벡터에서 블록간의 차이값이 가장 작은 값을 갖는 위치가 최종적인 블록의 움직임 벡터로 결정된다.

a	b	a	b	a	b	a	b
c	d	c	d	c	d	c	d
a	b	a	b	a	b	a	b
c	d	c	d	c	d	c	d
a	b	a	b	a	b	a	b
c	d	c	d	c	d	c	d
a	b	a	b	a	b	a	b
c	d	c	d	c	d	c	d

(a)

3	2	3	
4	1	4	1
3	2	3	2
	1	4	1

(b)

그림 3. 부표본화를 이용한 블록 움직임추정
(a) 4 : 1 부표본화된 영상패턴
(b) 탐색위치에서 부표본화된 영상패턴의 선택적 적용

Fig. 3. Block motion estimation using the pixel decimation.
(a) patterns of pixels with a 4 to 1 subsampling ratio.
(b) alternating schedule of the pixel subsampling patterns over the search area.

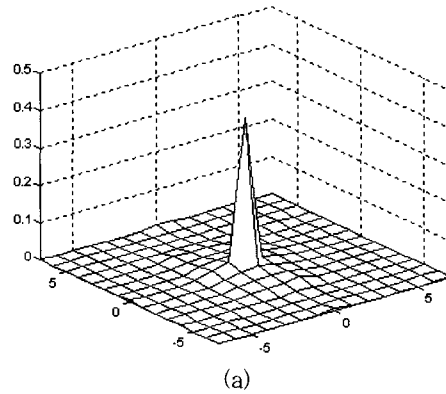
III. 고속 탐색기법을 이용한 블록정합 알고리즘

지금까지 SEA를 이용한 움직임 추정기법에서는 탐색영역에서 래스터 스캔 (raster scan) 순서로 탐색하거나 일정한 간격의 탐색점에서 블록정합을 수행하여 블

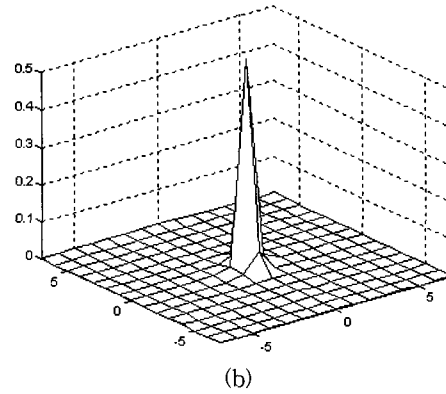
록 움직임 벡터를 찾는다^[1]. 그러나 일반적인 영상에서 블록의 움직임은 이웃한 블록의 움직임과 비슷하고 시간적으로 느리게 변한다^[38]. 따라서 블록의 움직임 벡터가 탐색영역에서 균일하게 분포한다고 가정하여 일정한 간격의 탐색점에서 혹은 래스터 스캔 방식으로 탐색하는 기법은 비효율적이다. 본 논문에서는 블록의 움직임 벡터가 탐색원점에 편향되어 있다는 특성을 이용하여 탐색패턴을 구성하고, 이를 이용하여 탐색점의 수를 더 줄임으로써 기존의 SEA의 성능을 향상시키는 고속 탐색기법을 제시한다.

1. 블록 움직임 벡터의 분포

그림 4는 "Table Tennis"와 "Football" 비디오 영상에 완전탐색 블록정합 알고리즘을 적용하여 얻어진 움직임 벡터의 분포이다. "Football" 영상의 경우 다양한



(a)



(b)

그림 4. 블록 움직임 벡터의 분포
(a) Table Tennis
(b) Football

Fig. 4. The distribution of block motion vector
(a) Table Tennis
(b) Football

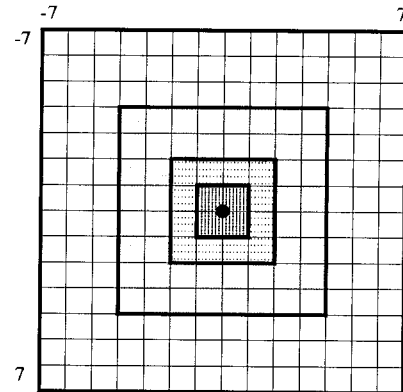
움직임이 혼합된 영상으로 대부분의 움직임 벡터가 탐색영역 중심의 3×3과 5×5 영역 내에서 각각 78%, 86% 이상 존재한다. “Table Tennis” 영상의 경우에는 비교적 빠른 움직임과 카메라 줌이 포함된 영상이다. 그리고 블록 움직임 벡터가 탐색영역의 중심의 3×3과 5×5 영역 내에서 각각 81%, 92% 이상 존재하고 있으므로, 마찬가지로 블록의 움직임 벡터는 탐색원점에 편향되어 있다는 것을 확인 할 수 있다.

블록정합에 소요되는 연산량을 줄임으로써 SEA의 성능을 향상시키기 위해서 PDE (partial distortion elimination), 임계값, 균일한 탐색패턴 등을 이용한 다양한 기법들이 제안되었다^[1]. 특히 블록정합에서 MAD 값이 변화가 큰 경우에는 식 (4)의 조건이 성립되는 탐색점의 수가 많이 발생할 수 있고, 이 경우에 MAD 초기값을 적당한 임계값으로 설정하여 MAD의 범위를 제한함으로써 블록정합이 요구되는 탐색점의 수를 줄일 수 있다^[13]. 따라서 MAD 최소값에 가까운 임계값을 설정한다면 더 많은 탐색점의 수를 줄일 수 있지만, 블록정합을 수행하고 얻어진 MAD 값의 범위가 영상 특성에 따라 달라질 수 있기 때문에 고정된 임계값을 설정하기가 쉽지 않다. 특히 영상에서 장면전환이 발생하는 경우에는 MAD 값이 커질 수 있고, 이런 경우에는 미리 정한 임계값을 초과하는 경우도 발생할 수 있기 때문에 블록 움직임 추정 오차가 클 가능성이 높다.

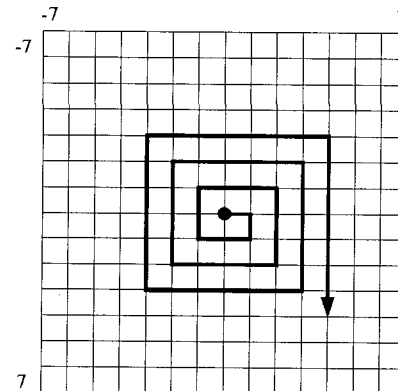
2. 탐색패턴을 이용한 고속 탐색

제안하는 기법은 탐색영역에서 블록 움직임 벡터의 분포 특성을 고려한 탐색패턴을 이용하여 탐색점의 수를 줄임으로써 기존의 SEA 성능을 향상시키는 기법이다. 그림 5는 움직임 벡터가 탐색원점에 편향되어 있다는 특성을 이용하여 구성된 탐색패턴이다. 그림 5(a)의 탐색패턴은 먼저 탐색원점을 중심으로 3×3, 5×5, 7×7, 그리고 그 외의 탐색영역내 탐색점에서 식 (4)의 조건이 성립되는지 여부를 확인한다. 그림 5(b)는 탐색원점에 근접한 탐색점부터 식 (4)의 조건을 확인한다. 그리고 조건에 만족하는 탐색점 위치에서 블록정합을 수행하여 블록 움직임 벡터를 찾는다. 이 기법들은 블록 움직임 벡터가 탐색원점에 편향되어 있다고 가정하여 MAD 최소값에 가까운 초기값을 얻을 수 있기 때문에 기존의 SEA의 정확성을 그대로 유지하면서 블록정합이 요구되는 탐색점의 수를 더욱 줄일 수 있다.

또한 탐색패턴을 이용하여 보다 많은 탐색점을 줄일



(a)



(b)

그림 5. 탐색패턴을 이용한 탐색기법

- (a) zonal
- (b) spiral 탐색패턴

Fig. 5. Search algorithms using the searching pattern.

- (a) zonal.
- (b) spiral searching pattern.

수 있지만, 블록정합이 요구되는 탐색점은 여전히 존재한다. 따라서 블록정합에서 필요한 연산량을 줄이기 위하여 블록내의 화소를 4:1로 부표본화하고 블록정합이 요구되는 탐색점에서 이를 정합하여 움직임을 추정하는 기법을 제시한다. 이때 SEA의 정확성은 다소 떨어지지만 블록내의 모든 화소를 이용하는 경우보다 연산량은 1/4로 줄어들기 때문에 보다 고속으로 움직임 추정을 수행할 수 있다. 또한 제안된 기법은 정합기준으로 MAD와 비슷한 성능을 유지하는 RBMAD를 이용하여 비교되는 화소의 연산량을 줄이는 기법과 결합하여 더 많은 연산량을 줄일 수 있다.

IV. 실험결과 및 검토

본 논문에서 제안한 고속 움직임 추정기법의 성능을 평가하기 위하여 컴퓨터 시뮬레이션을 수행하였다. 본 실험에서는 그림 6과 같이 비디오 영상 352×240 의 크기의 Football과 Table Tennis에서 각각 80 프레임을 사용하였다. 움직임 추정에 사용된 블록의 크기는 16×16 이며, 탐색범위는 $-7 \sim +7$ 로 설정하였다.

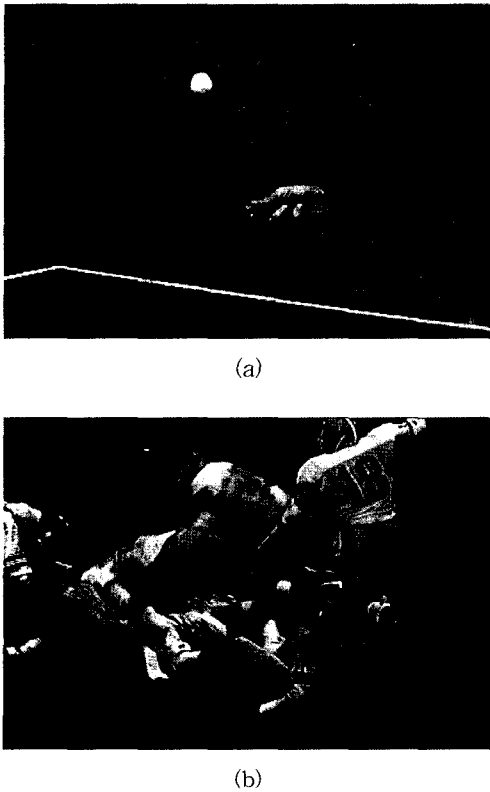


그림 6. 실험 비디오 영상

- (a) 탁구
(b) 미식축구

Fig. 6. Test video sequences.

- (a) table tennis
(b) football

그리고 제안된 기법의 블록정합 움직임 추정의 성능을 평가하기 위하여 $N \times M$ 크기의 원영상 I_0 과 추정된 영상 I_1 과의 MSE (mean square error)와 PSNR (peak-to-peak signal-to-noise ratio)을 이용한다. 두 영상간의 MSE와 PSNR은 각각 다음과 같다.

$$MSE(I_1, I_0) = \frac{1}{N \times M} \sum_{i=1}^N \sum_{j=1}^M (I_1(i, j) - I_0(i, j))^2 \quad (5)$$

$$PSNR(I_1, I_0) = 10 \log \frac{255^2}{MSE(I_1, I_0)} \quad (6)$$

먼저 정합기준들의 성능을 평가하기 위하여 MAD를 이용한 완전탐색, MAD를 이용한 3SS, MiniMax를 이용한 완전탐색과 제안하는 기법에 대해서 실험영상에 적용하여 움직임 추정에 대한 성능을 비교하였다.

표 2는 추정된 블록 움직임 벡터를 이용하여 보상된 영상간의 평균 MSE와 PSNR 값을 보여주고 있다. 그리고 그림 7은 움직임 추정 성능을 살펴보기 위해서 실험영상에서 프레임별로 비교한 결과이다.

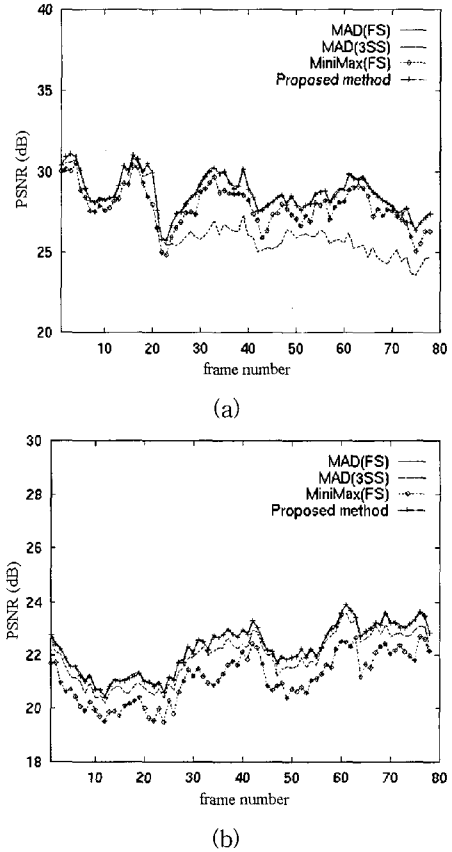


그림 7. 실험영상에 대한 MAD(FS), MAD(3SS), MiniMax(FS) 및 제안하는 방식에 대한 성능 비교

- (a) 탁구 (b) 미식축구

Fig. 7. Performance comparison of MAD(FS), MAD(3SS), MiniMax(FS), and proposed method.

- (a) Table Tennis. (b) Football video sequence.

표 2. 실험 영상에 대한 움직임 추정 알고리즘들의 평균 MSE와 PSNR

Table 2. Average MSE and PSNR of motion estimation algorithms for test video sequences.

Motion estimation algorithms	Table Tennis		Football	
	average MSE	average PSNR (dB)	average MSE	average PSNR (dB)
MAD (FS)	90.38	28.73	394.21	22.26
MAD (3SS)	153.21	26.65	428.03	21.91
MiniMax (FS)	111.03	27.85	511.88	21.13
Proposed method	91.68	28.67	397.43	22.23

표 2에서 제안하는 방식이 기존의 MAD를 이용한 완전탐색 방식과 비슷한 성능을 유지하면서 MAD를 이용한 삼단탐색보다 1.17 dB 좋은 성능을 유지하고 MiniMax를 이용한 완전탐색 알고리즘보다 0.96 dB 좋은 움직임 추정 성능을 유지하고 있음을 확인하였다.

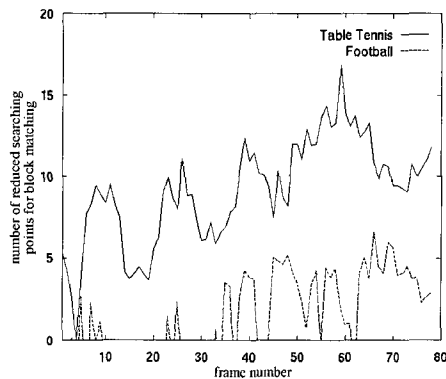


그림 8. zonal 탐색패턴을 이용한 탐색점 수의 감소
Fig. 8. Reduction of searching points using the zonal searching pattern.

그림 8은 제안하는 방식과 기존의 SEA 기법에 대한 블록당 평균 탐색점의 수를 비교한 결과이다. 제안하는 방식이 기존의 SEA 기법보다 각 블록당 평균적으로 약 5개의 탐색점을 더 줄이는 것을 확인하였다. 그리고 spiral 탐색패턴을 이용한 경우에도 zonal 탐색패턴을 이용한 경우와 비슷한 성능을 보여주고 있다. 따라서 움직임 벡터의 분포가 탐색원점에 편향되어 있다는 특성을 이용하여 제안하는 기법은 부가적인 정보없이 기존의 SEA의 성능을 향상시킨다.

V. 결론

본 논문에서는 SEA 기반의 고속 블록정합 움직임 추정 알고리즘을 제시하였다. 제안하는 기법은 블록 움직임 벡터가 탐색영역의 중앙에 편향되어 분포한다는 특성에 기반하여 기존의 SEA보다 탐색영역에서 탐색점의 수를 줄임으로써 성능을 향상시키는 기법이다. 제안된 기법은 균일한 탐색패턴과 래스터 스캔 방식으로 탐색하는 기법보다 블록당 약 5개 정도의 탐색점을 더 줄일 수 있었다. 또한 더 많은 연산량을 줄이기 위하여 제안된 기법은 블록정합에서 화소를 비교하는 연산량을 줄이는 RBMAD (reduced bits mean absolute difference) 정합기준과 블록정합이 요구되는 탐색점에서 블록내의 부표본화된 화소를 사용하였다. 제안된 기법은 완전탐색 블록정합 알고리즘과 비슷한 성능을 유지하면서 기존의 고속 알고리즘보다 성능이 향상되었음을 실험을 통하여 확인하였다.

참고 문헌

- [1] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. on Image Processing*, vol. 4, no. 1, Jan. 1995.
- [2] Y. Baek, H.-S. Oh, and H.-K. Lee, "An efficient block-matching criterion for motion estimation and its VLSI implementation," *IEEE Trans. on Consumer Electronics*, vol. 42, no. 4, Nov. 1996.
- [3] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 3, no. 2, April 1993.
- [4] T. Koga, K. Inuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," *Proc. NTC81*, pp. C9.6.1-9.6.5, New Orleans, LA, Nov. 1981.
- [5] J. R. Jain and A. K. Jain, "Displacement measurement and its application in inter-frame image coding," *IEEE Trans. on Communications*, vol. COM-29, pp. 1799-

- 1808, Dec. 1981.
- [6] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438-442, Aug. 1994.
- [7] 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 Technology*, vol. 6, no. 3, pp. 313-317, June 1996.
- [8] J. Feng, K.-T. Lo, H. Mehrpour, and A. E. Karbowiak, "Adaptive block matching motion estimation algorithm using bit-plane matching," International Conference on Image Processing, pp. 496-499, Oct. 1995.
- [9] L. W. Lee, J. F. Wang, J. Y. Lee, and J. D. Shie, "Dynamic search-window adjustment and interlaced search for block-matching algorithm," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 3, pp. 85-87, Feb. 1993.
- [10] B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low-complexity block-based motion estimation via one-bit transforms," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, no. 5, pp. 702-706, Oct. 1996.
- [11] K. Sauer and B. Schwartz, "Efficient block motion estimation using integral projections," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, no. 5, pp. 513-518, Oct. 1996.
- [12] M.-J. Chen, L.-G. Chen, T.-D. Chiueh, and Y.-P. Lee, "A new block-matching criterion for motion estimation and its implementation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 5, no. 3, pp. 231-236, June 1995.
- [13] H.-S. Wang and R. M. Mersereau, "Fast algorithms for the estimation of motion vectors," *IEEE Trans. on Image Processing*, vol. 8, no. 3, pp. 435-438, March 1999.

 저자 소개



吳泰明(正會員)

1961년 12월 25일생. 1986년 2월 광운대학교 전자통신공학과 졸업. 1986년~1998년 (주)삼성전관 연구소 책임 연구원. 1997년 고려대학교 대학원 전자공학과 졸업. 1998년~고려대학교 전자공학과 박사과정.

현재 명지전문대학 전자과 전임강사. 주간심분야는 영상압축 및 디지털 신호처리.