

고속 블록 정합을 위한 새로운 블록 기반 경사 하강 탐색 알고리즘 A New Block-based Gradient Descent Search Algorithm for a Fast Block Matching

곽성근(Sung-Keun Kwak)*

요 약

움직임 추정은 연속한 비디오 프레임간의 시간적 상관성을 이용하여 동영상 내에 존재하는 중복된 데이터를 제거하기 때문에 동영상 부호화에 있어서 중요한 역할을 한다. 그리고 서로 다른 형태와 크기를 가지는 탐색 패턴과 움직임 벡터의 분포는 블록 정합 기법에서 탐색 속도와 화질을 좌우하는 중요한 요소이다. 본 논문에서는 작은 크로스 탐색 패턴과 블록 기반 경사 하강 탐색 패턴을 이용한 새로운 고속 블록 정합 알고리즘을 제안한다. 이 방법은 작은 크로스 탐색 패턴을 이용하여 적은 탐색점으로 움직임이 적은 벡터를 우선 찾은 다음에 움직임이 큰 벡터에 대해서는 블록 기반 경사 하강 탐색 패턴을 이용하여 고속으로 움직임 벡터를 찾게 하였다.

실험결과, 제안된 알고리즘은 블록 기반 경사 하강 탐색 기법에 비하여 움직임 벡터 예측의 속도에 있어서 약 26~40% 이상의 높은 성능 향상을 보였다.

Abstract

Since motion estimation remove the redundant data to employ the temporal correlations between adjacent frames in a video sequence, it plays an important role in digital video coding. And in the block matching algorithm, search patterns of different shapes or sizes and the distribution of motion vectors have a large impact on both the searching speed and the image quality. In this paper, we propose a new fast block matching algorithm using the small-cross search pattern and the block-based gradient descent search pattern. Our algorithm first finds the motion vectors that are close to the center of search window using the small-cross search pattern, and then quickly finds the other motion vectors that are not close to the center of search window using the block-based gradient descent search pattern.

Through experiments, compared with the block-based gradient descent search algorithm(BBGDS), the proposed search algorithm improves as high as 26~40% in terms of average number of search point per motion vector estimation.

논문접수 : 2003.10.10.

심사완료 : 2003.10.17.

*정회원 시립인천전문대학 컴퓨터정보과교수

* 본 연구는 2003학년도 시립인천전문대학연구지원으로 수행되었음.

1. 서론

최근에 동영상 전송 수요가 확대되고 인터넷 및 이동 통신 등의 급격한 발전에 따라 동영상 분석 기법에 대한 관심이 고조되고 있다. 영상 부호화의 목적은 적은 양의 정보로 원영상을 충실히 표현하고 재생하는데 있으며, 동영상의 빠른 전송과 효율적인 저장을 위해서 동영상 내에 존재하는 시간적, 공간적 중복성을 동영상 분석 기법을 통해 제거함으로써 압축 효율을 증가 시키는 일이 효율적이다. 일반적으로 동영상 압축에 있어 동영상 프레임간의 움직임 벡터(motion vector)를 찾아 프레임간의 차를 부호화하는 움직임 추정(motion estimation)을 이용하여 압축 효율을 높이고 있다.

영상으로부터 움직임을 분석하기 위해 전통적으로 많이 사용하는 방법은 크게 화소 단위의 움직임 벡터를 추출하는 화소 재귀적 알고리즘(PRA: pel recursive algorithm)과 블록 단위의 움직임 벡터를 추출하는 블록 정합 알고리즘(BMA: block matching algorithm)이 있다. 화소 단위의 움직임 벡터를 추출하는 방법은 정확도가 높고 종류가 다양하지만 영상의 전체 범위에 걸쳐 화소 단위의 복잡한 연산을 수행함으로써 처리 시간이 오래 걸리는 단점이 있다.

따라서 블록 단위의 움직임 벡터를 추출하는 블록 정합 알고리즘이 화소 재귀적 알고리즘보다 널리 사용되고 있으며, 대표적인 알고리즘은 전역 탐색 블록 정합 알고리즘(FS: full search)이다. 이 방법은 영상을 모양과 크기가 동일한 사각형 블록으로 분할한 후 정합 척도를 탐색 영역 내의 블록들에 적용하여 움직임 벡터를 찾는다. 전역 탐색 블록 정합 알고리즘은 과정이 간단하고 예측 효율과 예측의 정확도를 고려할 때 전체적으로 좋은 특성을 가지며 하드웨어 구현이 용이하고 또한 탐색 영역의 내부 전체를 탐색하면서 가능한 모든 블록들에 대한 정합을 수행하므로 정합 오차가 가장 작은 움직임 벡터를 찾을 수 있지만 많은

계산량이 필요한 단점이 있다.

이러한 전역 탐색법의 단점을 극복하기 위해 속도가 개선된 3단계 탐색(TSS: three step search)[1], 새로운 3단계 탐색(NTSS: new three step search)[2], 4단계 탐색(FSS: four step search)[3], 블록 기반 경사 하강 탐색(BBGDS: block-based gradient descent search)[4], 다이아몬드 탐색(DS: diamond search)[5], 크로스-다이아몬드 탐색(CDS: cross-diamond search)[6] 등의 다양한 고속 블록 정합 알고리즘(FBMA: fast block matching algorithm)이 개발되었다. 이들 고속 블록 정합 알고리즘은 주로 탐색 영역 내에서 탐색점 후보의 개수를 감소시켜 전체 계산량의 감소를 유도하는 탐색 패턴을 사용한다.

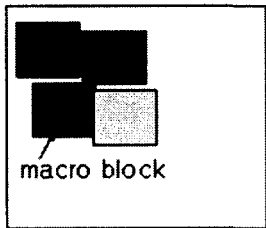
탐색 패턴이란 블록 정합을 위해 각 탐색 단계에서 정합 척도를 검사하는 탐색점들을 의미하며, 이 탐색점들 중에서 최소 BDM(minimum block distortion measure) 값을 가지는 위치를 중심으로 다음 단계의 움직임 탐색이 수행된다. 따라서 고속 블록 정합 움직임 탐색 방법에서 사용되는 탐색 패턴은 그 모양과 크기에 따라 탐색의 속도와 성능을 좌우하는 중요한 요소가 될 수 있다.

본 논문에서는 대부분의 움직임 벡터가 탐색 영역의 중심주위에 분포하므로 움직임 벡터를 찾는데 요구되는 계산량을 줄일 수 있도록 초기에 작은 크로스 탐색 패턴을 이용하여 적은 탐색점으로 움직임이 적은 벡터를 탐색하고, 이때 찾지 못한 움직임이 큰 벡터에 대해서는 블록 기반 경사 하강 탐색 패턴을 이용하여 고속으로 탐색하는 방법을 제안한다.

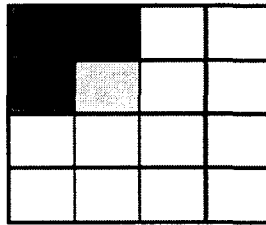
2. 기존의 움직임 추정 알고리즘

움직임 추정 기법은 영상의 각 단위로 블록 또는 화소 단위로 적용되며, 이 중에서 계산 복잡도 및 하드웨어 구현에 있어서 용이한 블록 단위의 움직임 추정이 널리 사용된다. 블록 단위의 움직임 추정은 동일한 블록 내의 화소

들은 동일한 움직임을 갖는다는 것과 블록들은 수평, 수직(상하좌우)으로만 움직인다는 두 가지 전체 조건을 가지므로 영상의 한 프레임을 [그림 1]과 같이 동일한 크기의 블록들로 나누고 이들의 각 블록들에 대하여 참조 프레임(reference frame)의 탐색 영역 내에서 정합 오차가 가장 작은 블록이 움직임을 갖게 된다.



(a) 참조프레임(기준영상)



(b) 대상프레임(현재영상)

[그림 1] 블록 정합 움직임 추정

움직임 예측은 물체의 움직임 변화를 추정하는 방법으로 연속적인 영상 신호에서 현재 프레임의 화소들이 이전 프레임에 비해 어느 정도 움직였는지를 나타내는 움직임 벡터를 추정하여 전체 영상의 전송 대신 움직임 벡터를 전송함으로써 전송 데이터량을 줄이는 방법이다. 이와 같이 움직임 예측은 주위 영상들의 움직임을 예측해서 부호화하는 신호를 줄임으로써 압축을 수행하는 것이다. 움직임 예측을 위해 가장 널리 사용하는 알고리즘은 블록 정합(block matching)이다. 이것은 연속된 두 영상 사이에서 각각의 화소를 16×16 단위로 비교하여 각 화소의 차이를 가지고 움직임을 예측하는 것이다. 현재 영상의 매크로블록과 기

준 영상의 매크로블록 사이의 식 (1)과 같이 절대 평균 오차(MAD: mean absolute difference)를 비교하여 MAD가 가장 작은 기준 영상의 위치를 검색하여 그 변위를 움직임 벡터로 결정한다.

$$MAD = \left(\frac{1}{M \times N} \right) \sum_{k=1}^M \sum_{l=1}^N | I_t(k, l) - I_{t-1}(k, l) | \quad (1)$$

여기서 N은 영상의 가로와 세로의 각각의 크기이며, $I_t(k, l)$ 은 원영상의 화면을 나타내고, $I_{t-1}(k+i, l+j)$ 은 움직임 추정 화면을 나타낸다.

실제 4개의 실험 비디오 시퀀스로부터 구한 평균 움직임 벡터의 분포를 살펴보면, <표 1>과 같이 대부분 중심 부근에 집중되어 있다. 특히 수평과 수직선상으로 많은 움직임이 있음을 알 수 있다. <표 1>에서 움직임 벡터가 $p=\pm 2$ 인 중앙 5×5 화소 내에서 찾아질 확률이 약 81.80%가 된다.

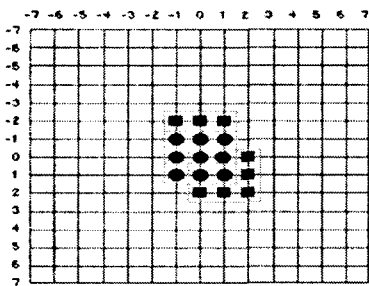
블록 기반 경사 하강 탐색(BBGDS)은 움직임 벡터의 대부분이 탐색 영역의 중심에 분포하는 특성을 고려하여, 중심점과 수평, 수직 방향의 거리가 1인 4점과 중심점으로부터 거리가 $\sqrt{2}$ 인 4개의 점으로 이루어진 3x3 화소를 가지는 정사각형 블록을 이용함으로써 다이아몬드 탐색 기법보다 현저한 속도 증가를 나타내는 것으로 평가된 탐색 기법이다[4].

블록 기반 경사 하강 탐색(BBGDS) 알고리

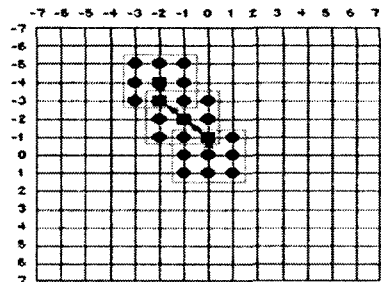
즘은 탐색 영역 내에 미리 정의된 탐색 블록의 중심점 (0,0)을 중심점으로 하는 9개의 탐색점을 가지는 정사각형 블록 패턴을 구성하여 최소 BDM 점을 계산한다. 최소 BDM 점이 정사각형 블록 패턴의 중심점이 될 때까지 이전 단계에서 구해진 최소 BDM 점을 중심으로 하는 정사각형 블록 패턴을 구성하여 탐색을 반복한다. 정사각형 블록 패턴을 이용한 탐색 경로는 [그림 2]와 같다.

-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.01%	0.02%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
0.01%	0.01%	0.00%	0.00%	0.00%	0.01%	0.01%	0.03%	0.01%	0.01%	0.01%	0.00%	0.00%	0.00%	0.00%
0.00%	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.05%	0.01%	0.01%	0.01%	0.01%	0.00%	0.00%	0.01%
0.01%	0.01%	0.01%	0.01%	0.01%	0.02%	0.02%	0.08%	0.01%	0.01%	0.01%	0.01%	0.01%	0.00%	0.00%
0.01%	0.02%	0.02%	0.02%	0.02%	0.02%	0.04%	0.11%	0.03%	0.03%	0.02%	0.01%	0.01%	0.01%	0.01%
0.02%	0.02%	0.02%	0.02%	0.04%	0.07%	0.16%	0.46%	0.09%	0.07%	0.04%	0.03%	0.02%	0.01%	0.01%
0.03%	0.03%	0.04%	0.05%	0.09%	0.23%	1.16%		0.88%	0.27%	0.15%	0.12%	0.09%	0.07%	0.03%
0.33%	0.21%	0.35%	0.28%	0.58%	1.11%				5.41%	1.01%	1.30%	1.06%	0.19%	0.11%
0.07%	0.06%	0.07%	0.08%	0.11%	0.20%	0.54%		0.55%	0.16%	0.09%	0.04%	0.03%	0.02%	0.02%
0.03%	0.04%	0.04%	0.05%	0.04%	0.06%	0.10%	0.38%	0.07%	0.05%	0.04%	0.02%	0.02%	0.02%	0.02%
0.00%	0.01%	0.01%	0.01%	0.02%	0.05%	0.12%	0.16%	0.03%	0.02%	0.03%	0.01%	0.01%	0.01%	0.01%
0.00%	0.00%	0.01%	0.01%	0.01%	0.02%	0.07%	0.14%	0.01%	0.01%	0.01%	0.02%	0.01%	0.01%	0.00%
0.00%	0.00%	0.00%	0.00%	0.01%	0.01%	0.02%	0.07%	0.01%	0.01%	0.00%	0.01%	0.00%	0.00%	0.00%
0.01%	0.01%	0.00%	0.01%	0.01%	0.01%	0.02%	0.05%	0.03%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

<표 1> 실험 영상의 평균 움직임 벡터 분포



(a) 추가점



(b) 탐색 경로

[그림 2] BBGDS 탐색 기법의 추가점 및 탐색 경로

3. 제안한 새로운 블록 기반 경사 하강 탐색 알고리즘

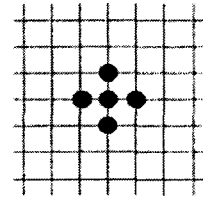
동영상에서 인접 프레임간의 시간 간격은 매우 짧기 때문에 단위 프레임의 시간당 움직임 크기 변화량은 일반적으로 적은 범위로 제한된다고 볼 수 있다. 즉, 연속하는 두 프레임간의 움직임에 많은 시간적 중복성을 가지고 있으므로 참조 프레임의 움직임 정보를 현재 프레임의 동일한 위치 매크로 블록의 탐색 시작점으로 사용함으로써 적은 탐색점들을 사용하여 움직임 벡터를 구할 수 있고, 양호한 보상 결과를 얻을 수 있다.

본 논문에서는 참조 프레임의 움직임 정보를 이용하여 해당 프레임에서 해당 블록의 움직임 예측을 위한 초기 탐색점을 설정하고, 움직임 벡터의 분포 특성에 맞는 적용적인 탐색 패턴을 사용한 새로운 움직임 탐색 기법을 제안한다.

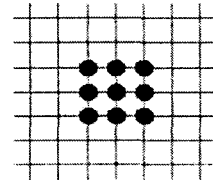
<표 1>과 같이 실험 동영상들에 대해 탐색 영역의 거리를 ± 7 로 두었을 때, 대부분의 움직임 벡터가 탐색 영역의 중심점으로부터 반경 2 픽셀 내에 거의 분포하기 때문에 기존의 블록 기반 경사 하강 패턴 기법은 움직임 벡터가 가운데 중심에 있을 경우에 초기 탐색점 수의 증가를 초래하게 되고 이는 전체적인 알고리즘의 속도를 저하 시킨다. 따라서 본 논문에서는 탐색 속도와 성능을 향상시키고자 중심점과 중심점으로부터 반경 1픽셀 내의 이웃점으로 구성된 [그림 3]의 (a)와 같은 작은 크로스 패턴을 초기 탐색 패턴으로 제안한다. 이 방법은 탐색 영역의 중심 주위에 분포하는 움직임 벡터의 탐색을 적은 탐색점으로 찾을 수 있어서 탐색 속도를 현저히 증가시킬 수 있다. 그리고 작은 크로스 패턴에서 탐색되지 못한 움직임 벡터에 대해서는 가장자리에 4개의 탐색점을 추가함으로써 기존의 블록 기반 경사 하강 기법과 같이 [그림 3]의 (b)와 같은 정사각형 블록 패턴이 적용된다..

따라서 제안하는 새로운 블록 기반 경사 하

강 패턴 기법은 실험 대상 동영상의 움직임 벡터들이 <표 1>과 같이 탐색 영역의 중심점과 이웃하는 4점에 분포할 확률이 평균 약 74.76%라는 점에 착안된 것으로 초기 탐색점 수가 작아 빠르게 탐색을 수행할 수 있다.



(a) 초기 탐색 패턴



(b) 정사각 블록 패턴

[그림 3] 제안한 알고리즘에서 사용하는 탐색 패턴

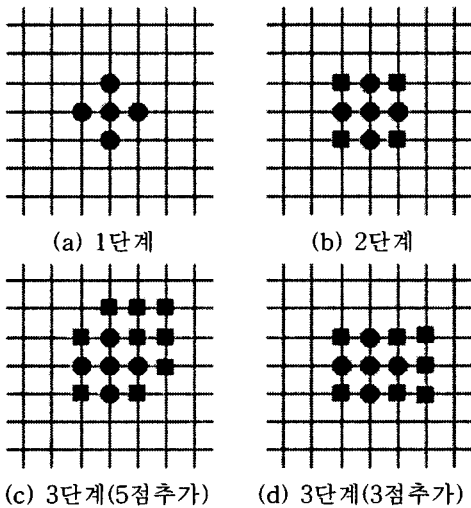
제안된 알고리즘은 비디오의 특성상 움직임을 가지는 블록이 프레임의 중앙에 있을 확률이 높다는 가정하에 프레임의 중심점을 포함하는 (0,0), (0,-1), (0,1), (-1,0), (1,0)의 5개의 탐색점을 블록 정합의 시작으로 한다. 최소 BDM이 (0,0)이면 탐색을 중단하고, 아니면 이 탐색점에 (-1,-1), (1,-1), (-1,1), (1,1)의 4개의 탐색점을 추가하여 최소 BDM 점을 중심으로 3x3 화소를 가지는 정사각형 블록을 구성하여 다음 단계의 블록 정합을 수행한다.

1단계: 초기 작은-크로스(SCSP: small cross-shaped pattern)의 5개의 탐색점에서 최소 BDM을 계산하여 중심점이 최소 BDM이면

탐색을 중단하고(MV=(0,0)), 아니면 2단계로 간다.

2단계: 이전 단계의 중심점에서 $(\pm 1, \pm 1)$ 에 위치하는 4개의 탐색점을 추가하여 최소 BDM을 구한 후, 3단계로 간다.

3단계: 탐색 패턴 내에 존재하는 9개의 탐색 블록 중심점에서 최소 BDM이 발견되면 탐색을 중단하고 이를 움직임 벡터의 최종 해로 한다. 이 경우가 아니라면, 다른 탐색점들 중에서 최소 BDM인 탐색점을 중심 탐색점으로 결정하고 [그림 4]의 (c), (d)와 같이 탐색 후보점을 추가한 후에 3단계를 반복한다.



[그림 4] 새로운 BBGDS 탐색 기법의 단계별 과정

4. 실험 결과

제안된 기법의 성능을 평가하기 위하여 4개의 CIF 실험 영상 Akiyo, News, Hall_monitor, Mother and Daughter의 실험 영상에 대해 각각 80프레임씩을 대상으로 실험하였고, 비교 탐색 기법으로는

FS, DS, CDS, BBGDS, 그리고 제안한 NBBGDS를 사용하였다. 움직임 예측에 사용된 매크로 블록의 크기는 16×16 픽셀이며, 탐색 영역의 변위는 ± 7 을 적용하여 Pentium IV 1.6GHz와 256MB 메모리가 장착된 컴퓨터상에서 실험을 수행하였다.

성능 비교 평가 함수로는 영상 화질의 품질을 평가하기 위해 평균 제곱 오차(MSE: mean squared error)와 PSNR(peak signal-to-noise ratio), 평균 절대값 오차(MAD: mean absolute difference)를 이용하였으며, 정합 오차 측정 함수로는 절대값 오차의 합(SAD: sum of absolute difference)을 이용하였다. 또한 제안하는 기법의 성능 향상을 측정하기 위해 블록당 탐색점의 수를 기존 방법들과 비교하였다.

탐색 기법의 성능을 비교평가하기 위해 사용한 함수인 MSE와 PSNR은 각각 식 (2)~(3)과 같다.

$$MSE = \left(\frac{1}{M \times N} \right) \sum_{k=1}^M \sum_{l=1}^N [I_t(k, l) - I_{t-1}(k, l)]^2 \quad (2)$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (3)$$

블록의 정합 오차를 측정하기 위한 함수 SAD는 식(4)과 같다.

$$SAD = \sum_{k=1}^M \sum_{l=1}^N |I_t(k, l) - I_{t-1}(k+i, l+j)| \quad (4)$$

실험 영상에 대한 실험 결과는 <표 2>, 원영상과 움직임 예측에 의해 생성된 영상의 각 화소 사이의 평균 오차를 각각 나타내며, PSNR은 각 실험 영상에 대한 PSNR[dB]의 평균을 나타내었다. Search Point 항목은 각 실험 영상에서 각 블록의 움직임 벡터 예측시 사용된 평균 탐색점의 수를 나타내었고 이를 이용하여 계산된 속도 향상의 비율을 SpeedUp Ratio 항목으로 나타내었다.

[그림 5], 그리고 [그림 6]에 각각 나타내었다. <표 2>에서는 실험 영상별로 전체 프레임에 대한 기존의 탐색 기법과 제안하는 탐색 기법의 실험 결과를 나타내었으며 [그림 5]와 [그림 6]에서는 실험 영상 Hall_monitor의 프레임별 블록당 평균 탐색점 수와 프레임별 픽셀당 MSE 값을 각각 그래프로 나타내었다.

<표 2>의 MSE 항목과 MAD 항목은

<표 2> 각 실험 영상에 대한 실험 결과

(a) 성능 비교 평가 함수의 결과값

Method	Akiyo			News			Hall_Monitor			Mother & Daughter		
	MSE	MAD	PSNR	MSE	MAD	PSNR	MSE	MAD	PSNR	MSE	MAD	PSNR
FS	4.029	0.603	42.780	12.615	1.042	38.225	25.747	2.541	34.410	8.379	1.496	40.204
DS	4.053	0.604	42.760	12.866	1.045	38.156	26.398	2.551	34.326	8.636	1.510	40.126
CDS	4.076	0.604	42.737	12.965	1.046	38.129	26.894	2.557	34.260	9.148	1.521	40.031
BBGDS	4.025	0.603	42.780	12.893	1.045	38.147	26.912	2.558	34.262	8.630	1.509	40.134
Proposed	4.025	0.603	42.780	12.885	1.045	38.149	26.896	2.557	34.264	8.626	1.508	40.137

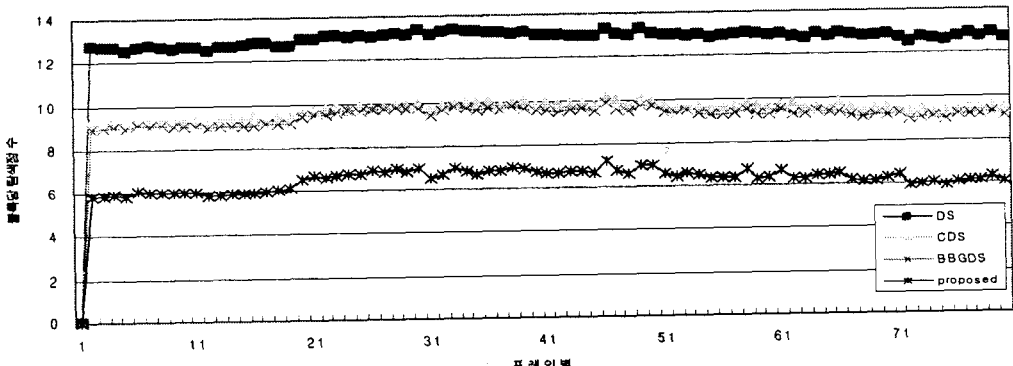
(b) 블록당 탐색점 수와 속도 향상 비율

Method	Akiyo		News		Hall_Monitor		Mother & Daughter	
	Search Point	SpeedUp Ratio	Search Point	SpeedUp Ratio	Search Point	SpeedUp Ratio	Search Point	SpeedUp Ratio
FS	204.28	1.000	204.28	1.000	204.28	1.000	204.28	1.000
DS	12.29	16.621	12.52	16.316	12.96	15.764	13.48	15.160
CDS	8.73	23.402	8.06	22.546	9.66	21.154	10.34	19.754
BBGDS	8.56	23.869	8.84	23.102	9.38	21.785	10.09	20.237
Proposed	5.11	39.989	5.58	36.587	6.44	31.739	7.50	27.237

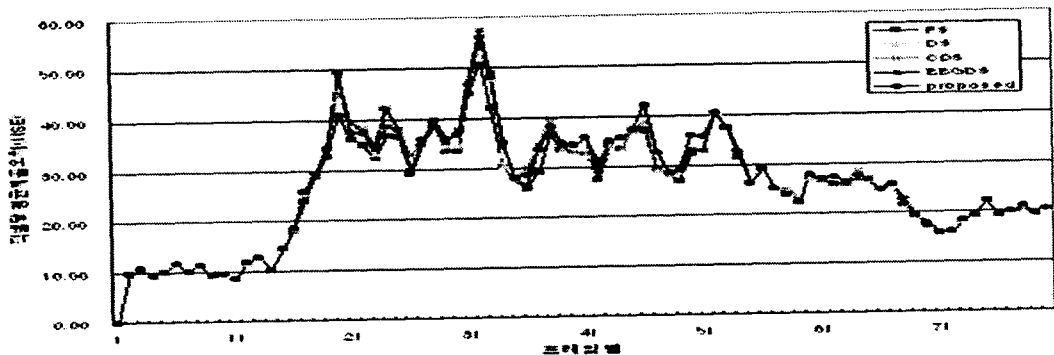
<표 2>의 실험 결과를 볼 때, 모든 실험 영상에 대해서 제안하는 탐색 기법이 기존의 탐색 기법들에 비해 탐색점의 수가 현저하게 감소함으로써 탐색 속도의 향상을 가져오는 것을 알 수 있다. 제안된 탐색 기법은 기존의 블록 기반 경사 하강 탐색 기법에 비해 탐색점의 수가 약 26~40% 정도 감소되었고 탐색 속도 측면에서는 약 135~167% 정도의 성능 향상을 나타내었다. 또한 크로스-다이아몬드 탐색 기법에 대해서는 탐색점의 수가 약 27~42% 정도 감소하였고 탐색 속도 측면에서는 약 138~171% 정도

의 성능 향상을 나타내었다.

영상의 화질에 대한 성능 비교 평가 함수로 사용된 MSE와 MAD의 실험 결과는 제안된 탐색 기법보다 27~40배 이상 많은 탐색점을 사용하여 움직임 벡터를 예측하는 전역 탐색 기법과도 많은 차이가 나지 않는 것을 볼 수 있다. 제안된 탐색 기법은 비교 대상이 된 블록 기반 경사 하강 탐색 기법이나 크로스-다이아몬드 탐색 기법에 비해 근사한 움직임 예측 성능을 가지면서, 탐색 속도 면에서는 우수한 성능 향상을 보였다.



[그림 5] 실험 영상 Hall_monitor의 프레임별 블록당 평균 탐색점 수



[그림 6] 실험 영상 Hall_monitor의 프레임별 픽셀당 MSE

[그림 5]에서는 실험에 사용된 Hall_monitor 영상에 대한 프레임별 블록당 평균 탐색점 수에 대한 그래프에서 나타나듯이 기존의 탐색 기법, 특히 직접적인 비교 대상인 크로스-다이아몬드 탐색 기법과 블록 기반 경사 하강 탐색 기법에 비해 움직임 예측에 필요한 탐색점의 수가 현저하게 감소된 것을 볼 수 있다. 제안하는 탐색 기법은 시간적 상관성이 적은 영상에서도 크로스-다이아몬드 탐색 기법에 대해서는 약 33%, 블록 기반 경사 하강 탐색 기법에 대해서는 약 31% 정도의 탐색점 수의 감소를 나타내었다. 제안하는 방법이 기존의 탐색 기법에 비해 움직임 예측에 필요한 탐색점 수를 현저히 감소시킴으로써 속도면에서 높은 성능 향상을 보였다.

5. 결론

본 논문에서는 동영상 프레임의 시간적 상관성에 따라 움직임 벡터의 분포를 바탕으로 하여 적용적으로 탐색 원점과 탐색 패턴을 바꾼 새로운 탐색 기법을 제안하였다. 초기에 작은 크로스 패턴을 이용한 빠른 탐색과 수평 방향으로 많이 분포하는 움직임 벡터의 빠른 탐색을 결합함으로써 실험을 통하여 알 수 있듯이 제안된 기법은 움직임 예측에 필요한 탐색점의 수를 약 27~75% 이상 감소시킴으로써 속도면에서 높은 향상을 보였고, 화질면에서도 가장 우수한 성능을 가지는 전역 탐색 알고리즘에 근접하는 MAD 값을 얻을 수 있었다. 블록 기반 경사 하강 탐색 알고리즘과 비교하였을 경우 움직임 보상 예측된 화질에 있어서 약 0.02~0.03dB 정도의 성능을 향상시켰으며 다른 기존의 탐색 기법들과 비교할 때에도 우수하거나 근사한 성능을 보였다.

움직임 벡터의 예측 과정에서 발생할 수 있는 예외적인 요소를 모두 고려하여 가장 적절한 현재 프레임의 매크로 블록 탐색 원점을 예측하고 본 논문에서 제안한 탐색 기법을 사

용하여 움직임 예측을 한다면 보다 빠르게 움직임 벡터를 찾을 수 있을 것이며, 우수한 보상 결과를 얻을 수 있을 것으로 기대된다. 또한 일정한 시간 간격을 두고 연속적으로 추출되는 움직임 벡터의 크기를 일정하게 조절할 수 있는 시간적인 동기화에 대한 연구도 보강될 필요가 있다.

참고문헌

- [1] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated Interframe Coding for Video Conferencing", in Proc. National Telecommunications Conf., New Orleans, LA, pp.G5.3.1-G5.3.5, Nov. 1981.
- [2] R. Li, Bing Zeng, "A New Three-Step Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Circuits & System for Video Tech., Vol. 4, No. 4, pp.438-442, Aug. 1994.
- [3] L. M. Po, W.C. Ma, "A Novel Four-Step Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Circuits & System for Video Tech., Vol. 6, No. 3, pp.313-317, June 1996.
- [4] L.K. Kuo, E. Feig, "A Block-Based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding", IEEE Transactions on Circuits & System for Video Tech., Vol. 6, No. 4, pp.419-422, Aug. 1996.
- [5] Shan Zhu, Kai-Kuang Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Image Processing, Vol. 9, No. 2, pp.287-290, Feb. 2000.
- [6] C. H. Cheung, L. M. Po, "A Novel Cross-Diamond Search Algorithm for Fast

Block-Matching Motion Estimation", IEEE Transactions on Circuits & System for Video Tech., Vol. 12, No. 12, pp.1168-1177, Dec. 2002.

곽 성 근



1973.2 건국대학교 졸업(공학사)

1980.2 연세대학교 대학원 졸업(공학석사)

1998.8 아주대학교 대학원 컴퓨터공학과(박사과정수료)

2003.~현재 시립인천전문대학

컴퓨터정보과 교수

관심분야: 동영상압축, 컴퓨터그래픽스, 멀티미디어, 애니메이션