

고속 블록 정합 움직임 예측을 위한 납작한 육각 패턴 기반 탐색 알고리즘

남현우*

A Flat Hexagon-based Search Algorithm for Fast Block Matching Motion Estimation

Hyeon-Woo Nam *

요 약

서로 다른 형태와 크기를 가지는 탐색패턴과 움직임 벡터의 분포는 고속 블록 정합 알고리즘에서 탐색 속도와 화질을 좌우하는 중요한 요소이다. 본 논문에서는 새로운 고속 블록 정합 알고리즘 개발을 위해 기존의 다이아몬드패턴과 육각패턴의 단점을 해결할 수 있는 납작한 육각패턴을 제안한다. 제안하는 알고리즘은 움직임이 큰 벡터에 대해서 납작한 육각패턴을 이용하여 고속으로 움직임 벡터를 찾게 하였다. 실험 결과를 통해, 제안된 납작한 육각패턴 탐색 알고리즘은 다이아몬드패턴 탐색 알고리즘이나 육각패턴 탐색 알고리즘에 비하여 움직임 벡터 추정의 속도에 있어서 약 0.4~21.3%의 성능 향상을 보였으며 화질 또한 PSNR 기준으로 약 0.009~0.531dB의 성능 향상을 보였다.

Abstract

In the fast 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 flat-hexagon search pattern that are solved disadvantages of the diamond pattern search algorithm(DS) and the hexagon-based search algorithm(HEXBS). Our proposed algorithm finds mainly the motion vectors that not close to the center of search window using the flat-hexagon search pattern. Through experiments, compared with the DS and HEXBS, the proposed flat-hexagon search algorithm(FHS) improves about 0.4 ~21.3 % in terms of average number of search point per motion vector estimation and improves about 0.009~0.531 dB in terms of PSNR(Peak Signal to Noise Ratio).

▶ Keyword : 고속 블록 정합 알고리즘(Fast Block Matching Algorithm), 탐색 패턴(Search Pattern), 움직임 추정(Motion Estimation)

• 제1저자 : 남현우

• 접수일 : 2007.2.14, 심사일 : 2007.3.6, 심사완료일 : 2007. 3.25.

* 동덕여자대학교 컴퓨터학과 전임강사

1. 서론

멀티미디어 컴퓨팅 환경에서 동영상 데이터의 빠른 전송 또는 효율적인 저장을 위해서는 동영상 내에 존재하는 시간적, 공간적 중복성을 동영상 분석 기법을 통해 제거하는 압축이 절실히 요구된다. 움직임 예측(ME: Motion Estimation)은 동영상을 이루고 있는 프레임간의 움직임 벡터(MV: Motion Vector)를 찾아 그 위치에 해당하는 이전 프레임의 블록과의 차를 부호화함으로써 시간적 중복성을 감소시킴으로써 압축 효율을 증가시킬 수 있는 매우 중요한 요소이다.

동영상 데이터로부터 움직임을 예측하기 위해 전통적으로 많이 사용하는 알고리즘은 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 재귀적 알고리즘은 정확도가 높고 종류가 다양하지만 영상의 전체 범위에 걸쳐 화소 단위의 복잡한 연산을 수행함으로써 처리 시간이 오래 걸리는 단점이 있다. 따라서 블록 정합 알고리즘이 화소 재귀적 알고리즘보다 널리 사용되고 있으며 대표적인 알고리즘이 전역 탐색(FS: full search) 기법이다 [1]. 이 알고리즘은 동영상의 각 프레임들을 일정한 모양과 크기를 가지는 사각형 블록으로 분할한 후 정합척도를 탐색영역 내의 모든 블록들에 적용하여 차이값이 가장 작은 블록을 움직임 벡터로 결정하게 된다. 전역 탐색은 처리과정이 간단하고 하드웨어적인 구현이 용이하며 정합오차가 가장 작은 움직임 벡터를 찾을 수 있지만 많은 계산을 필요로 하는 단점이 있다. 이러한 전역 탐색의 단점을 극복하기 위해 탐색속도를 향상시킬 수 있는 3단계 탐색(TSS: Three Step Search) [2], 4단계 탐색(FSS: Four Step Search) [3], 다이아몬드패턴 탐색(DS: Diamond Search) [4, 5, 6], 육각패턴 탐색(HEXBS: HEXagon-Based Search) [7, 8], 크로스-다이아몬드패턴 탐색(CDS: Cross-Diamond Search) [9], 크로스-육각패턴 탐색(CHS: Cross-Hexagonal Search) [10, 11] 등의 다양한 고속 블록정합 알고리즘(FBMA: fast block matching algorithm)이 개발되었다.

고속 블록 정합 알고리즘은 주로 탐색영역 내에서 탐색점 후보의 개수를 감소시켜 전체 계산량의 감소를 유도하는 탐색패턴을 사용한다. 탐색패턴이란 블록 정합을 위해 각 탐색단계에서 정합척도를 검사하는 탐색점들을 의미하며, 이 탐색점들 중에서 최소 BDM(Block Distortion Measure) 값을

가지는 위치를 중심으로 다음 단계의 움직임 벡터의 탐색이 수행된다. 따라서 고속 블록 정합 알고리즘에서 사용되는 탐색패턴은 그 모양과 크기에 따라 탐색속도와 화질을 좌우하는 중요한 요소가 될 수 있다.

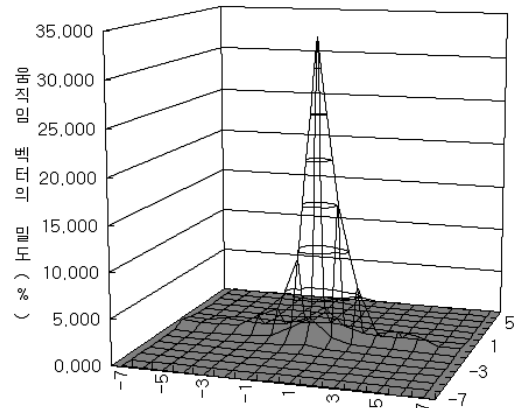


그림 1. 전체 실험동영상의 움직임 벡터 밀도
Fig. 1 Motion vector density of whole experimental videos

움직임 예측을 위한 탐색패턴의 설계를 위해 그림 1과 같이 실험에 적용한 12개의 동영상들에 존재하는 움직임 벡터에 대한 평균적 분포를 분석해 본 결과 대부분의 움직임 벡터가 탐색영역의 중심에 분포한다는 것을 볼 수 있다. 이런 분석 결과를 토대로 최근에는 고속 블록 정합 알고리즘들이 대부분의 움직임 벡터가 탐색영역의 중심 주위에 분포하는 것을 고려하여 다이아몬드패턴 [4, 5, 6]이나 육각패턴 [7, 8]과 같은 기존의 탐색패턴 적용 이전에 움직임 벡터를 찾는데 요구되는 계산량을 줄일 수 있도록 크로스패턴을 먼저 적용하여 움직임이 작은 벡터를 탐색하고 이때 찾지 못한 움직임이 큰 벡터에 대해서는 각각의 기본 탐색패턴을 적용하는 탐색패턴들이 제안되고 있다 [9, 10, 11].

고속 블록 정합 알고리즘 연구의 또다른 분류로는 같은 프레임 내의 인접한 이전 블록들의 움직임 벡터들의 정보를 이용하는 예측 탐색 알고리즘(PSA: Prediction Search Algorithm)들 [12, 13, 14] 과 다중해상도를 고려한 고속 블록 정합 알고리즘들이 연구되고 있다 [15, 16].

이런 고속 블록 정합 알고리즘들은 여러 단계의 패턴 적용을 위해 알고리즘의 복잡도가 커질 수 있으며 움직임이 큰 벡터에 대해서는 높은 효율을 나타내지 못하는 단점을 가지게 된다. 따라서 본 논문에서는 알고리즘의 복잡성을 높이지 않으면서도 동영상들의 움직임 벡터 분포를 고려한 탐색패턴을 적용한 고속 블록 정합 알고리즘을 제안한다.

본 논문의 구성은 2장에서 기존의 고속 블록 정합 알고리즘들에 대해 기술하고, 3장에서는 고속 블록 정합을 위해 제안하는 탐색패턴과 알고리즘에 대해 기술한다. 4장에서 기존의 알고리즘과 제안하는 알고리즘의 실험 결과를 비교하며, 5장에서는 결론적으로 제안하는 알고리즘에 대한 우수한 성능을 평가한다.

II. 기존의 고속 블록 정합 알고리즘

동영상 압축의 효율을 높이기 위한 움직임 예측 방법에 있어서 블록 단위의 움직임 예측이 널리 사용되는 이유는 동일한 블록 내의 화소들은 동일한 움직임을 갖는 것과 블록의 탐색을 수평, 수직방향으로만 한정한다는 두 가지 전제조건을 가지기 때문이다.

일반적으로 고속 블록 정합 알고리즘의 탐색패턴은 그 모양과 크기에 따라 탐색속도와 화질을 좌우하는 중요한 요소가 되므로 동영상들 내에 존재하는 움직임 벡터들의 분포를 분석하여 설계한다.

다이아몬드패턴 탐색(DS) 알고리즘은 움직임 벡터가 탐색 영역의 중심에 치우쳐 존재한다는 가정 하에 영상의 움직임이 크고 작음에 무관하게 화질 측면에서나 속도 측면에서 가장 좋은 효율을 나타낼 수 있도록 그림 2(a)와 같이 중심점으로부터 반지름 2픽셀을 가지는 원 내부에 포함된 13개의 점 중에서 바깥 부분의 9개의 탐색점들로 구성된 큰 다이아몬드패턴(LDSP: large diamond search pattern)과 그림 2(b)와 같이 반지름 1픽셀을 가지는 작은 다이아몬드패턴(SDPS: small diamond search pattern)으로 구성된다. 탐색 과정에서 수평 또는 수직 방향으로 탐색이 반복 진행되면 5개의 점을 새로운 패턴에 추가하고 대각선 방향이면 3개의 점을 추가된다.

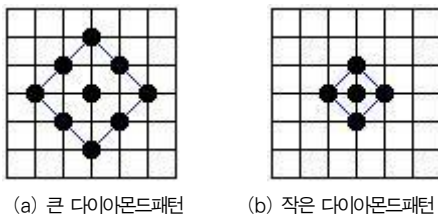


그림 2. 다이아몬드패턴 탐색 알고리즘의 탐색 패턴
Fig. 2 Search patterns used in the DS

다이아몬드패턴 탐색(DS) 알고리즘은 탐색 영역 내에 미리 정의된 탐색 블록의 중심점 (0,0)을 중심점으로 하는 9개의 탐색점을 가지는 큰 다이아몬드패턴을 구성하여 최소 BDM

점을 계산한다. 최소 BDM 점이 큰 다이아몬드패턴의 중심점이 될 때까지 이전 단계에서 구해진 최소 BDM 점을 중심으로 하는 큰 다이아몬드패턴을 구성하여 탐색을 반복한다. 큰 다이아몬드패턴의 중심점이 최소 BDM 점일 때, 탐색 패턴을 반지름 1픽셀로 이루어진 작은 다이아몬드패턴을 구성하고, 이 단계에서 구해지는 최소 BDM 점을 최종 움직임 벡터로 결정한다. 다이아몬드패턴을 이용한 탐색 경로는 그림 3과 같다[4, 5, 6].

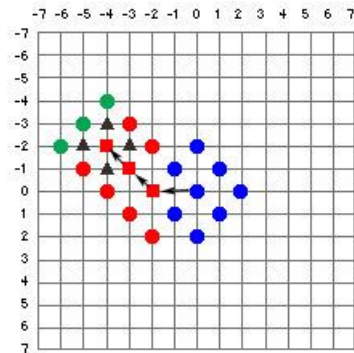


그림 3. 다이아몬드패턴 탐색 알고리즘의 탐색 경로
Fig. 3 Search path of the DS

육각패턴 탐색(HEXBS) 알고리즘은 그림 4(a)와 같이 중심점과 수평방향의 거리가 2인 두 점과 중심점으로부터 거리 $\sqrt{5}$ 인 4개의 점으로 이루어진 큰 육각패턴(LHEXBS: large hexagon-based search pattern)과 그림 4(b)와 같은 중심점으로부터 거리가 1인 4개의 탐색점을 갖는 작은 육각패턴(SHEXBS: small hexagon-based search pattern)을 이용함으로써 기존의 다이아몬드패턴 탐색보다 현저한 속도 증가를 나타내는 것으로 평가된 탐색 알고리즘이다[7, 8].

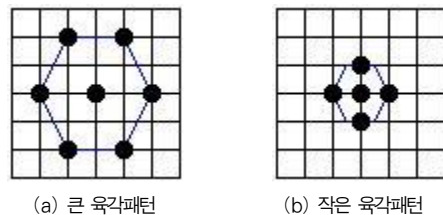


그림 4. 육각패턴 탐색 알고리즘의 탐색 패턴
Fig. 4 Search patterns used in the HEXBS

육각패턴 탐색(HEXBS) 알고리즘은 탐색 영역 내에 미리 정의된 탐색 블록의 중심점 (0,0)을 중심점으로 하는 7개의

탐색점을 가지는 큰 육각패턴을 구성하여 최소 BDM 점을 계산한다. 최소 BDM 점이 큰 육각패턴의 중심점이 될 때까지 이전 단계에서 구해진 최소 BDM 점을 중심으로 하는 큰 육각패턴을 구성하여 탐색을 반복한다. 큰 육각패턴의 중심점이 최소 BDM 점일 때, 중심점 이웃의 4개의 점을 포함하는 작은 육각패턴을 구성하여 최소 BDM 점을 구하고 이 점을 움직임 벡터로 결정한다. 육각패턴을 이용한 탐색 경로는 그림 5와 같다[7, 8].

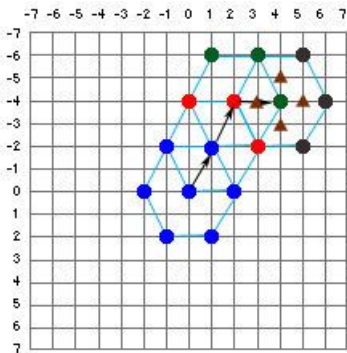


그림 5. 육각패턴 탐색 알고리즘의 탐색 경로
Fig. 5 Search path of the HEXBS

위에서 살펴본 두 가지의 탐색 알고리즘 중 다이아몬드패턴 탐색 알고리즘은 영상의 움직임이 크고 작음에 무관하게 화질 측면에서나 속도 측면에서 좋은 성능을 나타내는 방법이고, 육각 패턴 알고리즘은 영상의 움직임이 큰 경우에 좋은 성능을 나타내는 알고리즘으로 알려져 있다.

III. 제안하는 납작한 육각 패턴 기반 탐색 알고리즘

3.1 제안하는 납작한 육각 탐색 패턴의 설계

기존의 고속 블록 정합 알고리즘들은 여러 단계의 패턴 적용을 위해 알고리즘의 복잡도가 커지고 움직임이 큰 벡터에 대해서는 높은 효율을 나타내지 못하는 단점을 가지고 있다. 따라서 본 논문에서는 알고리즘의 복잡성을 높이지 않으면서도 동영상들의 움직임 벡터 분포를 고려한 탐색패턴을 적용한 고속 블록 정합 알고리즘을 제안한다.

그림 1의 움직임 벡터 분포도에서 살펴본 것과 같이 전체 실험 동영상의 탐색영역의 거리를 ±7로 두었을 때, 대부분의

움직임 벡터가 탐색영역의 중심 주위에 분포하는 것을 알 수 있었다. 이 중 탐색패턴의 설계에 영향을 미치는 영역은 탐색영역의 중심점으로부터 반경 2픽셀 내의 블록이다.

기존의 탐색패턴의 문제점과 새로운 탐색패턴의 설계를 위해 표 1과 같이 5x5 중심 블록 영역에 대해 움직임 벡터의 분포를 백분율로 나타내보았다.

표 1. 5x5 중심 블록 영역의 움직임 벡터의 밀도(%)
Table. 1 Motion vector density of 5x5 Center Block Area

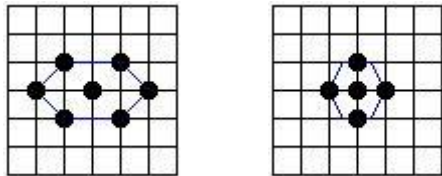
w	-2	-1	0	1	2
-2	0.155	0.235	0.496	0.160	0.107
-1	0.424	1.627	2.902	0.848	0.272
0	0.933	8.851	33.564	15.208	6.173
1	0.434	1.216	2.840	0.885	0.265
2	0.156	0.209	0.443	0.148	0.117

표 1의 5x5 중심 블록 전체의 움직임 벡터의 밀도는 전체 움직임 벡터의 80.668 % 를 차지하고 있고 y축 방향을 축소한 5x3 블록의 움직임 벡터의 밀도가 78.442 % 를 차지하고 있기 때문에 탐색패턴의 설계를 5x3 블록 영역을 기준으로 설계를 할 수 있다는 가능성을 발견했다. 또한 5x5 블록을 모두 이용하는 큰 다이아몬드패턴과 큰 육각패턴에 해당되는 움직임 벡터의 밀도가 48.185 %, 43.422 % 인데 비해 5x3 블록을 이용하면 되는 큰 납작한 육각패턴에 해당되는 움직임 벡터의 밀도는 47.246 % 였다.

탐색패턴의 포인트의 수가 큰 다이아몬드패턴에 비해 2개 적게 설계되었다는 것을 감안한다면 기존의 두 패턴보다 움직임 벡터의 밀도가 같거나 더 높음을 알 수 있다. 또한 기존의 HEXBS는 큰 육각패턴의 중심점이 최소 BDM인 경우에 작은 육각패턴을 구성하는데, 이 때, 탐색영역의 일부가 탐색대상에서 제외되어 부정확한 정합으로 인해 잡음이 많이 포함된 움직임 벡터를 추출할 확률이 높아진다.

본 논문에서는 이러한 기존의 탐색패턴의 문제점들을 해결하기 위해 그림 6(a)와 같이 5x3 블록 영역에 중심점과 중심점으로부터 수평방향의 거리가 2픽셀인 2점과 중심점으로부터 거리가 $\sqrt{2}$ 인 4개의 점으로 이루어진 큰 납작한 육각패턴(LFHSP: Large Flat-Hexagon Search Pattern)과 그림 6(b)와 같이 중심점으로부터 거리가 1인 4개의 탐색점을 갖는 작은 납작한 육각패턴(SFHSP: Small Flat-Hexagon Search Pattern)을 이용한 새로운 탐색패턴을 설계하고 납

작한 육각패턴을 적용한 새로운 고속 블록 정합알고리즘을 제안한다. 제안하는 탐색패턴은 특히 움직임이 큰 동영상의 움직임 예측에 효과적일 것으로 예측되며 빠른 탐색속도와 화질 향상에 기여하게 된다.



(a) 큰 납작한 육각패턴 (b) 작은 납작한 육각패턴

그림 6. 납작한 육각패턴 탐색 알고리즘의 탐색 패턴
Fig. 6 Search patterns used in the FHS

```

For each MacroBlock in the Current Frame
    Set a Center Point of LFHSP with MB(0,0)
    Make a LFHSP and Calculate a Min(Minimum) SAD point
    IF Min SAD is not equal to Center Point of LFHSP
        Set a Center Point of LFHSP with Min SAD(x,y)
        Make a LFHSP and Calculate a Min SAD point
    While Min SAD is equal to Center Point of LFHSP
        Set a Center Point of LFHSP with Min SAD(x,y)
        Make a LFHSP and Calculate a Min SAD point
    End While
    Set a Center Point of SFHSP with Min SAD(x,y)
    Make a SFHSP and Calculate a Min SAD point
    Decide MV with a Min SAD(x,y), Stop;
End For
    
```

그림 7. 납작한 육각패턴 탐색 알고리즘의 의사코드
Fig. 7 Pseudo Code of the FHS

3.2 제안하는 납작한 육각 패턴 기반 탐색 알고리즘의 탐색 과정

제안하는 납작한 육각패턴을 이용하여 움직임 벡터를 탐색하는 과정은 다음과 같은 3단계의 탐색 단계로 나타낼 수 있다.

1단계: 탐색 영역의 중심점과 주위 6개의 탐색 후보점을 포함하는 큰 납작한 육각패턴을 구성하고 최소 BDM을 계산한다. 계산된 최소 BDM 점이 탐색 패턴의 중심점에 위치하면 최종 3단계로 진행하고, 아니면 2단계로 진행한다.

2단계: 이전 단계에서 발견된 최소 BDM 점을 새로운 큰 납작한 육각패턴의 중심점으로 지정하여 탐색 후보 점을 추가하고 최소 BDM을 계산한다. 이 때, 계산된 최소 BDM 점이 탐색패턴의 중심점에 위치하면 3단계로 진행하고, 아니면 2단계를 반복 수행한다.

3단계: 탐색패턴을 이전 단계에서 계산된 최소 BDM 점을 중심으로 수직, 수평 방향의 1픽셀 거리의 4개의 이웃점을 포함하는 작은 납작한 육각패턴으로 변경하고, 이 단계에서 구한 최소 BDM 점이 최소 정합 오차를 가지는 움직임 벡터의 최종 해로 결정한다.

제안하는 납작한 육각패턴 탐색 알고리즘을 의사코드로 나타내면 아래 그림 7과 같다.

IV. 실험 결과

4.1 실험 환경

제안한 납작한 육각패턴 기반 탐색 알고리즘의 성능을 평가하기 위하여 다음 표 2와 같이 서로 다른 해상도를 가지는 CIF(해상도 352X288) 동영상, SIF(해상도 352X240) 동영상, 그리고 QCIF(해상도 176X144) 동영상을 각각 4개씩 선택하여 총 12개의 실험 동영상을 각각 80프레임씩을 기준으로 실험하였다. 실험 결과를 바탕으로 납작한 육각패턴 기반 탐색 알고리즘을 최종적으로 제안하고자 한다.

표 2. 실험 동영상
Table. 2 Experimental videos

동영상 형식	실험에 사용된 동영상 (괄호 안은 전체 프레임 수)	
CIF	Akiyo(300) Coastguard(300)	Foreman(400) Stefan(100)
SIF	Susie(120) Mobile(140)	Flower Garden(150) Popp(150)
QCIF	Akiyo(300) Coastguard(300)	Table Tennis(300) Stefan(300)

움직임 추정의 성능 비교를 위해 기존의 탐색 알고리즘 중에서 FS, DS, HEXBS, CDS, CHS 그리고 제안하는 납작한 육각 패턴 기반 탐색 알고리즘을 사용하였다.

움직임 추정에 사용된 매크로 블록의 크기는 16x16 픽셀

이며, 탐색영역의 변위는 ±7을 적용하여 Pentium IV 3.0GHz와 512MB 메모리가 장착된 컴퓨터상에서 실험을 수행하였다.

성능 비교 평가 함수로는 영상 화질의 품질을 평가하기 위해 식 (1)의 평균 절대값 오차(MAD: mean absolute difference)[4, 8, 11]와 식 (2)의 평균 제곱 오차(MSE: mean squared error)를 이용한 PSNR(peak signal-to-noise ratio)를 이용하였다[1-16].

$$MAD(i, j) = \left(\frac{1}{N^2}\right) \sum_{k=1}^N \sum_{l=1}^N |I_i(k, l) - I_{i-1}(k+i, l+j)| \dots (1)$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}$$

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

또한 정합 오차 측정 함수로는 식 (3)의 절대값 오차의 합(SAD: sum of absolute difference)을 이용하였다[1, 4, 11].

$$SAD(i, j) = \sum_{k=1}^N \sum_{l=1}^N |I_i(k, l) - I_{i-1}(k+i, l+j)| \dots (3)$$

4.2 실험 결과

제안한 납작한 육각패턴 탐색 알고리즘을 12개의 전체 동영상에 실험을 진행하였다. 성능 평가 결과 중 해상도가 서로 다른 실험 동영상 특히 제안하는 탐색패턴의 설계 목표에 부합되는 움직임이 큰 동영상 2개씩(MAD 가 4 이상인 동영상)에 대해 그 실험 결과를 표 3과 표 4에 나타내었다. 표 3

에서는 실험 동영상에서의 각 블록에 대한 평균 탐색 회수를 PN(search Point Number) 과 FS 의 탐색 회수를 기준으로 하여 각 움직임 예측 기법의 상대적인 속도 향상을 의미하는 SR(Speedup Ratio) 로 비교 결과를 나타내었다. 또한 표 4에서는 기존의 움직임 예측 기법과 제안하는 알고리즘의 각 매크로 블록에 대한 MAD 와 평균 PSNR 값을 비교한 결과를 나타내었다.

실험 결과를 볼 때, 제안한 알고리즘은 전역 탐색의 속도 문제를 개선한 고속 블록 정합 알고리즘이므로 전역 탐색과의 비교는 불필요한 것으로 예측된다. 실험 결과에 대한 성능 비교는 제안한 알고리즘의 탐색패턴 설계의 근간이 된 문제점을 가진 다이아몬드패턴 탐색 알고리즘(DS)과 육각패턴 탐색 알고리즘(HEXBS)을 기본적인 대상으로 정했고, 탐색영역의 중심 부분에 확률적으로 높은 분포를 가지는 움직임 벡터를 가지는 동영상의 빠른 탐색을 위해 개발된 크로스-다이아몬드 패턴 탐색 알고리즘(CDS)과 크로스-육각패턴 탐색 알고리즘(CHS)의 실험 결과도 성능 비교에 적용하였다.

표 3에서 나타낸 바와 같이 제안한 알고리즘과 비교 대상인 두 알고리즘과의 탐색 속도 측면의 실험 결과를 보면, 제안하는 알고리즘의 탐색패턴 모델인 DS 에 대해서는 평균 탐색점의 수가 약 18.1 ~27.0 % 정도 감소하여 약 15.4~21.3 % 정도의 탐색속도의 향상을 나타내었고, 탐색속도가 빠른 HEXBS 에 대해서는 평균 탐색점의 수가 약 0.3~3.9 % 정도 증가되어 약 0.2~4.1 % 정도의 탐색속도가 저하되었다. 또 다른 비교대상인 CDS 와 CHS 에 대해서도 평균 탐색점의 수가 약 0.3~11.8 % 정도 감소하여 약 0.4~10.5 % 정도의 탐색속도의 향상을 나타내었다.

표 3. 실험 결과(탐색 횟수 및 속도 향상율)
Table. 3 Experimental Result(search Point Number and Speedup Ratio)

Method	CIF				SIF				QCIF			
	Coastguard		Stefan		Flower Garden		Mobile		Coastguard		Stefan	
	PN	SR	PN	SR	PN	SR	PN	SR	PN	SR	PN	SR
FS	204.280	1.000	204.280	1.000	202.050	1.000	202.050	1.000	184.560	1.000	184.560	1.000
DS	16.940	12.062	16.770	12.180	16.810	12.018	12.640	15.989	12.760	14.465	14.270	12.932
HEXBS	13.070	15.635	12.980	15.737	13.150	15.365	10.450	19.344	10.440	17.684	11.360	16.241
CDS	14.970	13.646	14.440	14.143	14.960	13.504	10.130	19.944	10.500	17.579	12.810	14.411
CHS	15.000	13.620	14.620	13.972	14.570	13.867	10.130	19.953	10.480	17.619	12.710	14.515
FHS	13.600	15.025	13.200	15.471	13.400	15.083	10.700	18.890	10.470	17.644	11.460	16.105

표 4. 실험 결과(MAD 및 PSNR)
Table. 4 Experimental Result(MAD and PSNR)

Method	CIF				SIF				QCIF			
	Coastguard		Stefan		Flower Garden		Mobile		Coastguard		Stefan	
	MAD	PSNR	MAD	PSNR	MAD	PSNR	MAD	PSNR	MAD	PSNR	MAD	PSNR
FS	5.333	29.642	6.715	25.693	8.520	23.994	9.546	22.943	4.180	31.016	6.794	25.716
DS	5.529	29.409	7.717	24.655	8.768	23.806	9.604	22.888	4.266	30.926	6.896	25.622
HEXBS	5.689	29.216	7.929	24.501	9.509	23.320	9.761	22.698	4.399	30.801	7.018	25.531
CDS	5.720	29.304	8.076	24.409	8.822	23.729	9.553	22.933	4.277	30.917	6.924	25.599
CHS	5.567	29.378	7.883	24.526	8.833	23.703	9.555	22.930	4.304	30.894	6.969	25.566
FHS	5.554	29.387	7.706	24.686	8.700	23.851	9.597	22.897	4.292	30.906	6.896	25.622

제안하는 알고리즘과 성능비교 대상인 여러 알고리즘과의 탐색 속도 결과 비교에 의해 제안하는 알고리즘이 평균 탐색점수의 감소로 인한 탐색 속도의 향상을 가져옴을 알 수 있었다.

표 4로 나타낸 바와 같이 제안한 알고리즘과 비교 대상인 두 알고리즘과의 화질 향상 측면의 실험결과를 보면, 제안하는 알고리즘의 탐색패턴 모델인 DS 에 대해서는 평균 MAD 값이 약 0.1~0.8 % 정도 감소하여 약 0.009~0.045 dB 정도의 화질 향상을 나타내었지만 Coastguard 동영상의 경우에는 CIF 형식이나 QCIF 형식 모두 미미한 화질 저하를 나타내었다. 또한 탐색속도가 빠른 HEXBS 에 대해서는 평균 MAD 값이 약 1.7~9.3 % 정도 감소되어 약 0.091~0.531 dB 정도의 화질이 향상되었다. 또다른 비교대상인 CDS 와 CHS 에 대해서도 평균 MAD 값이 약 0.2~4.8 % 정도 감소하여 약 0.009~0.277 dB 정도의 화질 향상을 나타내었다.

제안하는 알고리즘과 성능비교 대상인 여러 알고리즘과의 화질 향상에 대한 결과 비교에 의해 제안하는 알고리즘이 평균 MAD 값의 감소로 인해 화질 향상이 됨을 볼 수 있었다.

제안하는 탐색 패턴과 알고리즘이 움직임이 큰 동영상에 다루기 위해 설계된 것이기 때문에 실험에 사용된 동영상에 대한 실험 결과 중 움직임이 적은 경우는 표에서 나타나지 않았다. 이들 동영상의 경우에도 DS 에 대해서는 탐색 속도의 높은 향상 효과를 나타내었고, HEXBS 에 대해서는 화질의 높은 향상 효과를 나타내었다.

V. 결론

본 논문에서는 움직임이 큰 동영상 프레임의 시간적 상관성에 따라 움직임 벡터의 분포의 중심지향적인 특성을 고려하였고, 알고리즘의 복잡도를 높이지 않으면서도 기존의 고속 블록 정합 알고리즘보다 효율적인 탐색을 위한 새로운 탐색패턴을 설계하였고 이것을 적용한 새로운 고속 블록 정합 알고리즘을 제안하였다.

제안하는 납작한 육각패턴 탐색 알고리즘과 탐색패턴 설계 기준이 된 기존의 두 고속 블록 정합 알고리즘, 실험 비교에 포함시킨 다른 고속 블록 정합 알고리즘들과의 전체적인 실험 결과를 통해, 움직임이 큰 동영상에 적용된 제안하는 알고리즘의 성능에 대해 평가를 하자면 탐색 속도의 측면에서는 제안하는 알고리즘이 평균 탐색점수의 감소로 인해 비교 대상 알고리즘들에 비해 약 0.4~21.3 % 정도의 탐색 속도의 향상을 나타내었고, 복원된 동영상의 화질 측면에서도 제안하는 알고리즘이 비교 대상 알고리즘들에 비해 약 0.009~0.531 dB 정도의 성능 향상을 나타내었음을 볼 수 있었다. 또한 탐색패턴의 설계 기준이 아니었던 움직임이 작은 동영상에 대해서도 일정한 성능 향상을 나타내었다.

본 논문에서는 제안하는 알고리즘과 기존의 알고리즘들과의 실험 결과를 통해 동영상의 구성 형식과 촬영 형태에 따라 탐색패턴의 설계가 다르게 적용되어야 한다는 결론을 도출하게 되었으며, 제안한 알고리즘의 탐색 패턴이 설계 기준이었던 움직임이 큰 동영상에서만 아니라 움직임이 작은 동영상에서도 일정한 성능 향상을 나타내었다는 것을 통해 제안하는 알고리즘이 다른 기존의 탐색 알고리즘들과 비교할 때에 우수한 성능을 제공할 수 있음을 알 수 있었다.

본 논문에서는 움직임 벡터의 추정을 위해 탐색영역에서의 고정된 탐색 시작점을 이용하였다. 동영상의 현재 프레임에 존재하는 인접 매크로 블록이나 이전 프레임의 움직임 벡터를 고려하여 가변적인 탐색 시작점을 설정할 수 있는 적응성을 고려한다면 본 논문에서 제안한 탐색패턴과 탐색 알고리즘을 더욱 효과적인 고속 블록 정합 알고리즘을 위한 탐색패턴과 탐색 알고리즘으로 발전시켜나갈 수 있을 것이다.

참고문헌

- [1] Information Technology-Generic Coding of Moving Pictures and Associated Audio Information: Video, ISO/IEC 13818-2(MPEG-2 Video), 2000.
- [2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated Interframe Coding for Video Conferencing", In Proc. of the International Telecommunications Conference, New Orleans, LA, pp. G5.3.1-G5.3.5, Nov. 1981.
- [3] L. M. Po, and W. C. Ma, "A Novel Four-Step Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Circuits & System for Video Technology, Vol. 6, No. 3, pp. 313-317, June 1996.
- [4] Shan Zhu, and 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.
- [5] T. Sappasitwong, S. Aramvith, S. Jitapunkul, A. Tamtrakarn, P. Kitti-punyangam, and H. Kortrakulkij, "Adaptive asymmetric diamond search algorithm for block-based motion estimation", In Proc. of the International Conference on Digital Signal Processing, pp. 563-566, July 2002.
- [6] 홍성용, "구속조건을 적용한 다이아몬드 탐색 알고리즘에 의한 고속블록정합움직임 추정", 한국컴퓨터정보학회 논문지, Vol. 8, No. 4, pp. 13-20, Dec. 2003.
- [7] Ce Zhu, Xiao Lin, Lap-Pui Chau, Keng-Pang Lim, Hock-Ann Ang, and Choo-Yin Ong, "A novel hexagon-based search algorithm for fast block motion estimation", In Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 1593-1596, May 2001.
- [8] Ce Zhu, Xiao Lin, and Lap-Pui Chau, "Hexagon-Based Search Pattern for Fast Block Motion Estimation", IEEE Transactions on Circuits & System for Video Technology, Vol. 12, No. 5, pp. 349-355, May 2002.
- [9] C. H. Cheung, and L. M. Po, "A Novel Cross-Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Circuits & System for Video Technology, Vol. 12, No. 12, pp. 1168-1177, Dec. 2002.
- [10] 박인영, 남현우, 위영철, 김하진, "십자와 육각패턴을 이용한 고속 블록 정합 동작 예측기법", 한국정보처리학회 논문지 B, Vol. 10, No. 7, pp. 811-814, Dec. 2003.
- [11] 남현우, "개선된 크로스-육각 패턴을 이용한 고속 블록 정합 움직임 추정 알고리즘", 한국컴퓨터정보학회 논문지, Vol. 11, No. 4, pp. 77-85, Sep. 2006.
- [12] Oscar T.-C. Chen, "Motion Estimation Using a One-Dimensional Gradient Descent Search", IEEE Transactions on Circuits & System for Video Technology, Vol. 10, No. 4, pp. 608-616, June 2000.
- [13] A. M. Tourapis, O. C. Au, and M. L. Liou, "Highly efficient predictive zonal algorithms for fast block-matching motion estimation", IEEE Transactions on Circuits & System for Video Technology, Vol. 12, No. 10, pp. 934-947, Oct. 2002.
- [14] F. Moschetti, M. Kunt, and E. Debes, "A Statistical Block-Matching Motion Estimation", IEEE Transactions on Circuits & System for

- Video Technology, Vol. 13, No. 4, pp. 417-431, Apr. 2003.
- [15] Tae-Gyoung Ahn, Yong Ho Moon, and Jae Ho Kim, "Fast full-search motion estimation based on multilevel successive elimination algorithm", IEEE Transactions on Circuits and Systems for Video Technology, VOL. 14, NO. 11, pp. 1265-1269, Nov. 2004.
- [16] G. Calvagno, F. Fantozzi, R. Rinaldo, and A. Viareggio, "Model-based global and local motion estimation for videoconference sequences", IEEE Transactions on Circuits and Systems for Video Technology, VOL. 14, NO. 9, pp. 1156-1161, Sep. 2004.

저 자 소 개



남 현 우

1993년 아주대학교 공과대학

전자계산학과 (공학사)

1996년 아주대학교 대학원

컴퓨터공학과(공학석사)

2004년 아주대학교 대학원

컴퓨터공학과(공학 박사)

2005년~현재: 동덕여자대학교 컴

퓨터학과 전임강사

관심분야 : 컴퓨터그래픽스, 영상처

리, 멀티미디어 및 게임