論文
4-12-9

크로스 패턴과 납작한 육각 탐색패턴을 이용한 고속 블록 정합 알고리즘

A Fast Block Matching Algorithm by using the Cross Pattern and Flat-Hexagonal Search Pattern

남현우(Hyeon-Woo Nam)[1], 김종경(Jong-Gyung Kim)[2]

요 약

서로 다른 형태와 크기를 가지는 탐색패턴과 움직임 벡터의 분포는 블록 정합 알고리즘에서 탐색 속도와 화질을 좌우하는 중요한 요소이다. 본 논문에서는 크로스패턴과 납작한 육각패턴을 이용한 새로운 고속 블록 정합 알고리즘을 제안한다. 이 알고리즘은 초기에 크로스 패턴을 이용하여 탐색영역의 중심 주위에 분포 확률이 높은 움직임이 작은 벡터를 우선 찾은 다음에 움직임이 큰 벡터에 대해서는 납작한 육각패턴을 이용하여 고속으로 움직임 벡터를 찾게 하였다.

실험결과, 제안된 알고리즘은 육각패턴 탐색 알고리즘에 비하여 움직임 벡터 추정의 속도에 있어서 약 0.2~6.2% 의 성능 향상을 보였으며 화질 또한 PSNR 기준으로 약 0.02~0.31dB 의 향상을 보였다.

Abstract

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 cross pattern and the flat-hexagon search pattern. Our algorithm first finds the motion vectors that are close to the center of search window using the cross pattern, and then fastly finds the other motion vectors that are not close to the center of search window using the flat-hexagon search pattern.

Through experiments, compared with the hexagon-based search algorithm(HEXBS), the proposed cross pattern and flat-hexagonal pattern search algorithm(CFHPS) improves about 0.2~6.2% in terms of average number of search point per motion vector estimation and improves about 0.02~0.31dB in terms of PSNR(Peak Signal to Noise Ratio).

1) 정회원 : 강남대학교 지식정보공학부 강사
2) 정회원 : 인천전문대학 컴퓨터정보과 겸임교수

논문접수 : 2003. 12. 8.
심사완료 : 2003. 12. 15.

1. 서론

컴퓨터 그래픽스 분야나 영상 처리 분야에서는 여러 가지 파일 포맷을 가지는 정지 영상 및 동영상 데이터들을 주로 취급하게 되는데 이러한 영상 데이터를 사실적으로 표현하자면 컬러(Color)와 해상도(Resolution)가 고려되어져야한다. 하지만 트루컬러(True Color)와 고해상도의 영상 데이터들은 그 크기가 너무 커서 기억장소 내에 저장하거나 통신선로 상에서 전송을 하게 될 경우 많은 저장용량과 전송시간이 소요되어 데이터 처리 비용의 증가와 효율의 저하를 초래한다.

동영상의 빠른 전송 또는 효율적인 저장을 위해서는 동영상 내에 존재하는 시간적, 공간적 중복성을 동영상 분석 기법을 통해 제거하는 압축이 필요하다. 움직임 추정(ME: Motion Estimation)은 동영상 프레임간의 움직임 벡터(MV: Motion Vector)를 찾아 그 위치에 해당되는 이전 프레임의 블록과의 차를 부호화함으로써 시간적 중복성을 감소시켜 압축 효율을 증가시킬 수 있는 중요한 요소이다.

영상으로부터 움직임을 분석하기 위해 블록 정합 알고리즘이 널리 사용되고 있으며 대표적인 알고리즘은 전역 탐색(FS: full search) 기법이다. 이 알고리즘은 영상을 모양과 크기가 동일한 사각형 블록으로 분할한 후 정합척도를 탐색영역 내의 블록들에 적용하여 움직임 벡터를 찾는다. 전역 탐색은 과정이 간단하고 하드웨어 구현이 용이하며 정합오차가 가장 작은 움직임 벡터를 찾을 수 있지만 많은 계산을 필요로 하는 단점이 있다. 이러한 전역 탐색의 단점을 극복하기 위해 다양한 고속 블록정합 알고리즘(FBMA: fast block matching algorithm)이 개발되었다. 대표적인 고속 블록정합 알고리즘으로는 3단계 탐색(TSS: Three Step Search)[1], 4단계 탐색(FSS: Four Step Search)[2], 다이아몬드 탐색(DS: Diamond Search)[3], 육각형 기반 탐색(HEXBS: HEXagon-Based Search)[4][5], 크로스-다이아몬드 탐색(CDS: Cross-Diamond Search)[6], 작은 크로스-다이아몬드 탐색(SCDS: Small-Cross-Diamond Search)[7], 십자와 육각패턴을 이용한 탐색(CHS: Cross-Hexagonal Search)[8], 단위다이아몬드와 납작한 육각패턴을 이용한 탐색(UDFHS: Unit-Diamond and Flat-Hexagonal Search)[9] 등이 있다.

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.

본 논문에서는 대부분의 움직임 벡터가 탐색영역의 중심 주위에 분포하므로 움직임 벡터를 찾는데 요구되는 계산량을 줄일 수 있도록 초기에 크로스패턴을 이용하여 적은 탐색점으로 움직임이 적은 벡터를 탐색하고 이때 찾지 못한 움직임이 큰 벡터에 대해서는 납작한 육각패턴을 이용하여 고속으로 탐색하는 알고리즘을 제안한다.

2. 기존의 고속 블록정합 알고리즘

시간적인 중복성에 의한 압축은 움직임 벡터를 찾고 그 위치에 있는 이전 블록과의 차이값을 압축/저장하는 방법이다. 그러므로 동영상 압축의 효율을 높이기 위한 척도는 공간적인 중복성보다는 시간적인 중복성을 어떻게 찾아내는지의 여부가 될 수 있다.

시간적 중복성을 찾기 위한 움직임 추정은 영상의 블록 또는 화소 단위로 적용되며, 계산 복잡도 및 하드웨어 구현에 있어서 용이한 블록 단위의 움직임 추정이 널리 사용된다. 블록 단위의 움직임 추정은 동일한 블록 내의 화소들은 동일한 움직임을 갖는다는 것과 블록의 탐색을 수평, 수직방향으로만 한정한다는 두 가지 전제조건을 가지므로 영상의 한 프레임을 동일한 크기의 블록들로 나누고 이들의 각 블록들에 대하여 참조 프레임(reference frame)의 탐색영역 내에서 정합오차가 가장 작은 블록이 움직임 벡터

로 결정된다. 이러한 영상간의 가장 유사한 블록을 찾는 작업을 움직임 추정이라고 한다.

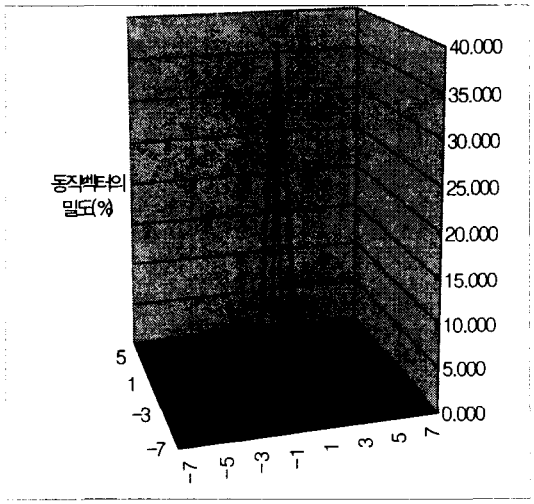
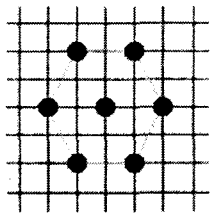


그림 1. 전체 실험 동영상의 움직임 벡터의 평균 분포도

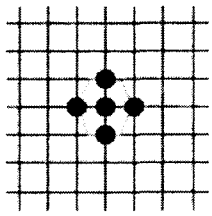
Fig 1. Average Distribution Chart of Motion Vector of all Experimental Sequence

움직임 추정을 위한 탐색 패턴의 설계를 위해 그림 1과 같이 실험에 적용한 18개의 동영상들에 존재하는 움직임 벡터에 대한 평균적인 분포도를 차트로 나타내었는데 대부분의 움직임 벡터가 탐색 영역의 중심에 분포한다는 것을 볼 수 있다.

육각패턴 탐색(HEXBS)은 그림 2의 (a)와 같이 중심점과 수평방향의 거리가 2인 두 점과 중심점으로부터 거리가 $\sqrt{5}$인 4개의 점으로 이루어진 큰 육각패턴(LHEXBSP: large hexagon-based search pattern)과 그림 2의 (b)와 같은 중심점으로부터 거리가 1인 4개의 탐색점을 갖는 작은 육각패턴(SHEXBSP: small hexagon-based search pattern)을 이용함으로써 기존의 다이아몬드 탐색보다 현저한 속도증가를 나타내는 것으로 평가된 탐색 알고리즘이다 [4][5].



(a) 큰 육각패턴(LHEXBSP)



(b) 작은 육각패턴(SHEXBSP)

그림 2. 육각패턴 탐색 알고리즘의 탐색패턴

Fig 2. Search Pattern of Hexagon-based Pattern Search Algorithm

육각패턴 탐색(HEXBS) 알고리즘은 탐색 영역 내에 미리 정의된 탐색 블록의 중심점 (0,0)을 중심점으로 하는 7개의 탐색점을 가지는 큰 육각 패턴을 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 3과 같다 [4][5].

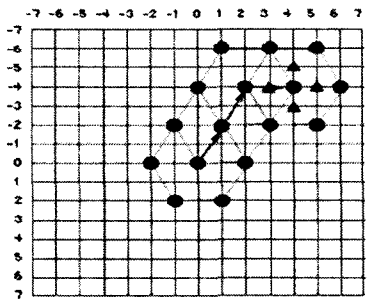
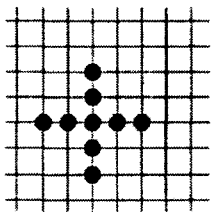


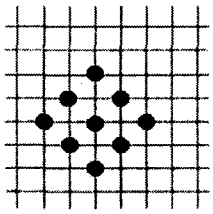
그림 3 육각패턴 탐색 알고리즘의 탐색경로

Fig 3. Search Path of Hexagon-based Pattern Search Algorithm

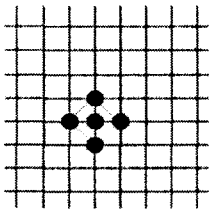
크로스-다이아몬드 탐색(CDS)은 그림 1에서 볼 수 있듯이 움직임 벡터의 대부분이 탐색영역의 중심에 분포하는 특성을 고려하여 그림 4의 (a)와 같이 중심점을 포함하는 초기 크로스패턴을 이용한 탐색을 수행한 후 중심점이 최소 BDM이 아닐 경우 비교적 탐색효율이 높은 것으로 평가된 그림 4의 (b)와 (c)의 다이아몬드 패턴을 이용한 탐색을 수행하는 알고리즘이다 [6].



(a) 크로스패턴



(b) 큰다이아몬드패턴(LDSP)



(c) 작은 다이아몬드패턴(SDSP)

그림 4 크로스패턴과 다이아몬드패턴을 이용한 크로스-다이아몬드패턴

Fig 4. Cross-Diamond Patterns using Cross Pattern and Diamond Pattern

크로스-다이아몬드 탐색(CDS) 알고리즘은 그림 4의 (a)와 같이 초기 크로스패턴의 9개의 탐색점을 계산하여 중심점이 최소 BDM이 되면 중심점을 움직임 벡터로 결정하고 탐색을 중단한다(첫 번째 종료조건). 그렇지 않으면 크로스패턴의 중심점으로부터 각각 $(-1,-1)$, $(1,-1)$, $(1,1)p(-1,1)p$에 위치하는 4개의 후보 탐색점 중 크로스패턴 탐색에서 구해진 최소 BDM과 가까운 두 개의 탐색점을 추가하여 작은 다이아몬드패턴 탐색을 수행한다. 이때 중심점이 최소 BDM이 되면 작은 다이아몬드패턴의 중심점을 움직임 벡터로 결정하고 탐색을 중단한다(두 번째 종료조건). 두 가지의 종료조건을 만족하지 못할 때에는 그림 4의 (b)와 같이 큰 다이아몬드패턴(LDSP: large diamond search pattern)을 적용하여 탐색패턴의 중심점이 최소 BDM이 될 때까지 큰 다이아몬드 형태의 주위 9개의 탐색점을 계산한다. 계산된 최소 BDM 점이 중심점일 때 그림 4의 (c)와 같이 작은 다이아몬드패턴(SDSP: small diamond search pattern)을 구성하고 최종적으로 BDM 계산을 통해 정합 블록의 움직임 벡터를 구한다. 크로스-다이아몬드패턴을 이용한 탐색 경로는 그림 5와 같다.[6]

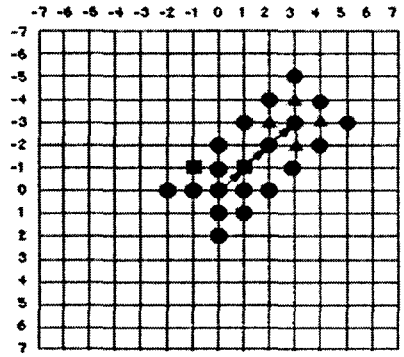


그림 5 크로스-다이아몬드패턴의 탐색경로

Fig 5. Search Path of Cross-Diamond Pattern

3. 제안한 크로스 패턴과 납작한 육각 탐색패턴을 이용한 고속 블록정합 알고리즘

동영상에서 연속하는 두 프레임간의 움직임에는 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 보상결과를 얻을 수 있다.

본 논문에서는 참조 프레임의 움직임 정보를

이용하여 해당 프레임에서 해당 블록의 움직임 추정을 위한 초기 탐색점을 설정하고, 움직임 벡터의 분포특성에 맞는 적응적인 탐색패턴을 사용한 새로운 고속 블록정합 알고리즘을 제안한다.

그림 1의 움직임 벡터 분포도에서 살펴본 것과 같이 실험 동영상들에 대해 탐색영역의 거리를 ±7로 두었을 때, 대부분의 움직임 벡터가 탐색영역의 중심 주위에 분포하는 것을 알 수 있었고, 표 1과 같이 움직임 벡터의 분포를 백분율로 나타내 보았을 때 탐색영역의 중심점으로부터 반경 2픽셀 내에 많은 분포를 가지고 있다는 것을 알 수 있다. 그러므로 기존의 CDS 기법과 같은 크로스패턴을 사용한다면 탐색영역의 중심 주위에 분포하는 움직임 벡터를 빨리 탐색할 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 탐색 알고리즘이 가진 문제점들을 해결하기 위해 중심점과 중심점으로부터 수평, 수직 방향의 반경 2픽셀 내의 이웃점으로 구성된 그림 6의 (a)와 같은 크로스패턴과 그림 6의 (b)와 (c)의 납작한 육각패턴을 결합한 새로운 알고리즘을 제안한다. 제안하는 크로스패턴의 적용에 의하여 탐색영역의 중심주위에 분포하는 움직임 벡터를 빨리 탐색할 수 있으며, 크로스패턴에서 탐색되지 못한 움직임 벡터에 대해서는 납작한 육각패턴이 적용되어 빠른 탐색과 화질 향상에 기여하게 된다.

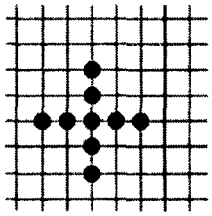
제안하는 크로스패턴과 납작한 육각패턴을 이용한 탐색 알고리즘은 실험대상 동영상의 움직임 벡터들이 표 1과 같이 탐색영역의 중심점과 수평, 수직방향의 이웃점에 분포할 확률이 평균 약 71.3%라는 사실을 이용하여 초기 탐색점들

							-1	0	1	2	3	4	5	6	7
	0.080	0.029	0.023	0.035	0.050	0.037	0.043	0.139	0.047	0.036	0.041	0.030	0.029	0.035	0.073
	0.034	0.031	0.020	0.037	0.033	0.035	0.033	0.202	0.038	0.027	0.033	0.038	0.025	0.031	0.032
	0.047	0.024	0.026	0.035	0.044	0.040	0.058	0.124	0.043	0.037	0.050	0.030	0.026	0.024	0.043
	0.049	0.029	0.033	0.039	0.053	0.053	0.080	0.197	0.060	0.048	0.054	0.052	0.036	0.034	0.048
	0.064	0.040	0.038	0.052	0.068	0.091	0.126	0.265	0.107	0.080	0.085	0.060	0.049	0.041	0.064
	0.082	0.047	0.054	0.074	0.133	0.186	0.262		0.292	0.155	0.145	0.087	0.051	0.043	0.088
	0.114	0.052	0.069	0.121	0.180	0.394	1.210		0.971	0.381	0.258	0.153	0.099	0.101	0.158
	0.556	0.274	0.453	0.534	1.744						1.025	0.916	0.814	0.285	0.625
	0.158	0.127	0.316	0.272	0.303	0.377	1.021		1.056	0.459	0.201	0.126	0.064	0.077	0.139
	0.107	0.061	0.060	0.092	0.161	0.167	0.273		0.244	0.170	0.149	0.077	0.055	0.051	0.104
	0.060	0.037	0.035	0.061	0.096	0.100	0.149	0.274	0.115	0.083	0.096	0.058	0.040	0.038	0.079
	0.041	0.027	0.031	0.043	0.060	0.059	0.089	0.224	0.062	0.048	0.058	0.054	0.029	0.033	0.054
	0.047	0.022	0.029	0.041	0.057	0.038	0.063	0.117	0.052	0.041	0.060	0.036	0.027	0.024	0.061
	0.036	0.027	0.026	0.042	0.037	0.041	0.144	0.238	0.050	0.038	0.039	0.046	0.020	0.033	0.049
	0.101	0.034	0.036	0.043	0.052	0.043	0.067	0.307	0.063	0.058	0.063	0.038	0.032	0.032	0.128

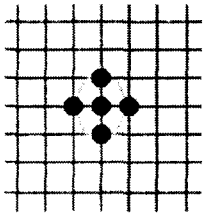
표 1. 움직임 벡터의 평균 분포표(%)

Table 1. Average Distribution Table of Motion Vector

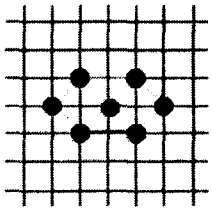
을 배치하였고, 탐색영역의 중심에서 수평방향으로 존재하는 움직임 벡터를 빨리 탐색하여 점진적으로 움직임 추정의 확률을 높일 수 있도록 추가 탐색점들을 적절히 배치하였다.



(a) 크로스패턴



(b) 큰 납작한 육각패턴



(c) 작은 납작한 육각패턴

그림 6. 크로스패턴과 수평 방향(X축)의 움직임 벡터를 고려한 납작한 육각패턴

Fig 6. Cross Pattern and Flat-Hexagonal Pattern for motion vector of horizontal direction(X-axis)

본 논문에서 제안한 크로스패턴과 납작한 육각 탐색 패턴을 이용하여 움직임 벡터를 탐색하는 절차를 나타내면 다음과 같은 5단계의 탐색 구조를 나타낸다.

1단계: 탐색 영역의 중심점을 포함하는 크로스 패턴의 9개의 탐색 후보점으로 최소 BDM을 계산하여 탐색패턴의 중심점이 최소 BDM이면 탐색을 중단하고 중심점을 동작벡터((MV)=(0,0))로 결정한다. 아니면 2단계로 진행한다.

2단계: 탐색 영역의 중심점을 중심으로 하는 큰 다이아몬드 탐색패턴의 (±1,±1)에 위치하는 네 개의 탐색 후보점 중 이전 단계에서 구해진 최소 BDM 점에 인접한 두 개의 탐색 후보점을 추가하여 작은 다이아몬드 패턴 형태의 최소 BDM을 계산한다. 만약 최소 BDM이 크로스 패턴의 중간 날개에 위치하면 탐색을 중단하고 동작벡터로 결정한다. 아니면 3단계로 진행한다.

3단계: 이전 단계에서 계산된 최소 BDM 점을 탐색패턴의 중심점으로 설정하고 육각형의 꼭지점에 해당하는 주위 6개의 탐색 후보점을 포함하는 큰 납작한 육각 탐색패턴(LFHSP: Large Flat-Hexagon Search Pattern)을 구성하고 최소 BDM을 계산한다. 계산된 최소 BDM 점이 탐색패턴의 중심점에 위치하면 최종 5단계로 진행하고, 아니면 4단계로 진행한다.

4단계: 이전 단계에서 발견된 최소 BDM 점을 새로운 큰 납작한 육각 탐색패턴의 중심점으로 지정하여 탐색 후보점을 추가하고 최소 BDM을 계산한다. 이 때, 계산된 최소 BDM 점이 탐색패턴의 중심점에 위치하면 5단계로 진행하고, 아니면 4단계를 반복 수행한다.

5단계: 탐색패턴을 이전 단계에서 계산된 최소 BDM 점을 중심으로 수직, 수평 방향의 1화소 이웃점을 포함하는 작은 납작한 육각 탐색패턴(SFHSP: Small Flat-Hexagon Search Pattern)으로 변경하고, 이 단계에서 구한 최소 BDM 점이 최소 정합 오차를 가지는 동작벡터의 최종해로 결정한다.

제안하는 알고리즘은 탐색 영역의 중심점을 포함하는 그림 7의 (a)와 같은 큰 크로스 패턴의 9개의 탐색 후보점으로 최소 BDM을 계산한다. 탐색패턴의 중심점이 최소 BDM이면 탐색을 중단하고 중심점을 동작벡터로 결정한다. 아니면 탐색 영역의 중심점을 중심으로 하는 큰 다이아몬드 탐색패턴(LDSP)의 (±1,±1)에 위치하는 네 개의 탐색 후보점 중 그림 7의 (b)와 같이 이전 단계에서 구해진 최소 BDM 점에 인접한 두 개의 탐색 후보점을 추가하여 최소

BDM을 계산한다. 만약 최소 BDM이 크로스 패턴의 중간 날개에 위치하면 탐색을 중단하고 동작벡터로 결정한다. 최소 BDM이 아니면 그림 7의 (c)와 같이 계산된 최소 BDM 점을 탐색패턴의 중심점으로 설정하고 육각형의 꼭지점에 해당하는 주위 6개의 탐색 후보점을 포함하는 큰 납작한 육각 탐색패턴을 구성하고 최소 BDM을 계산한다. 계산된 최소 BDM 점이 탐색패턴의 중심점에 위치할 때까지 그림 7의 (d)와 같이 큰 납작한 육각 탐색패턴의 구성과 최소 BDM 계산을 반복한다.

최종적으로 최소 BDM 점이 탐색패턴의 중심점이 되었을 때 그림 7의 (e)와 같이 최소 BDM 점을 중심으로 수직, 수평 방향의 1화소 이웃점을 포함하는 작은 납작한 육각 탐색패턴으로 변경하고, 최소 BDM 계산 결과 구해진 최소 BDM 점이 동작벡터의 최종 해로 결정된다.

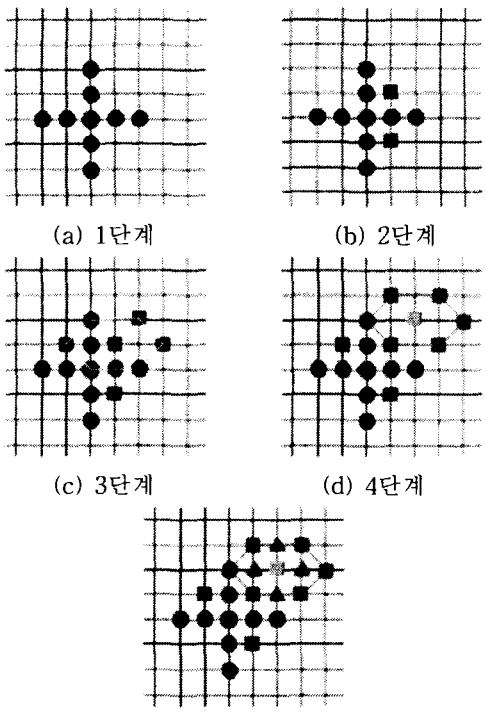


그림 7. 크로스패턴과 납작한 육각 패턴 기반 탐색 알고리즘의 탐색경로

Fig 7. Search Path of Cross and Flat-Hexagonal Pattern Search Algorithm

4. 실험 결과

제안한 크로스 패턴과 납작한 육각 탐색 패턴 기반 알고리즘의 성능을 평가하기 위하여 다음 표 4-1과 같이 서로 다른 해상도를 가지는 CIF(해상도 352X288) 동영상, SIF(해상도 352X240) 동영상, 그리고 QCIF(해상도 176X144) 동영상을 각각 6개씩 선택하여 총 18개의 실험 동영상을 각각 100프레임씩을 기준으로 실험하였다. 실험 결과를 바탕으로 최적의 크로스-육각 패턴 기반 알고리즘을 최종적으로 제안하고자 한다.

표 2. 다른 해상도를 가진 실험 동영상

Table 2. Experimental Sequences with Different Resolution

해상도	실험에 사용된 동영상(괄호 안은 전체 프레임 수)
CIF(352x288)	Akiyo(300) Foreman(400) Coastguard(300) Miss America(150) Mother and Daughter(300) Stefan(100)
SIF(352x240)	Garden(150) Popp(150) Mobile(140) Football(150) Susie(120) Tennis(112)
QCIF(176x144)	Akiyo(300) Table(300) Claire(150) Miss America(150) Mother and Daughter(200) Stefan(300)

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

움직임 추정에 사용된 매크로 블록의 크기는 16×16 픽셀이며, 탐색영역의 변위는 ±7을 적용하여 Pentium Ⅳ 1.6GHz와 256MB 메모리가 장착된 컴퓨터상에서 실험을 수행하였다.

성능 비교 평가 함수로는 영상 화질의 품질을 평가하기 위해 평균 절대값 오차(MAD: mean absolute difference)와 평균 제곱 오차(MSE: mean squared error)를 이용한 PSNR(peak signal-to-noise ratio)를 이용하였다.

$$MAD = (\frac{1}{M \times N}) \sum_{i=1}^{M} \sum_{j=1}^{N} | O_t(x,y) - E_t(x,y) |$$

(1)

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}, MSE = (\frac{1}{M \times N}) \sum_{i=1}^{M} \sum_{j=1}^{N} [O_t(x,y) - E_t(x,y)]^2$$

(2)

식 (1)과 (2)에서 M과 N은 각각 영상의 가로와 세로의 크기를 나타내며, $O_t(x,y)$는 원영상의 화면을 나타내고, $E_t(x,y)$는 움직임 추정 화면을 나타낸다.

정합 오차 측정 함수로는 절대값 오차의 합(SAD: sum of absolute difference)을 이용하였으며 식(3)과 같다.

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

(3)

식 (3)에서 MB는 매크로 블록의 가로와 세로의 크기를 나타내며, $I_t(k,l)$은 현재 프레임을 나타내고, $I_{t-1}(k+i, l+j)$은 이전 프레임을 나타낸다.

표 3과 표 5에 실험 동영상의 해상도별로 그룹화하여 실험 동영상별로 전체 프레임에 대한 기존의 탐색 알고리즘과 제안하는 탐색 알고리즘들의 실험 결과를 나타내었다. 표 3에는 각 실험 동영상에서 각 매크로 블록의 동작벡터 추정시 사용된 평균 탐색점의 수를 NS(Number of Search point) 항목으로 나타내었고, 계산된 평균 탐색점 수 NS를 이용하여 계산된 속도 향상의 비율을 SPEEDUP 항목에 표시하였다. 또한 표 5에서는 실험 동영상의 해상도별로 그룹화하여 각 동영상 프레임의 원영상과 동작추정에 의해 생성된 영상의 각 화소 사이의 평균 절대 차분을 MAD 항목에 나타내었고, 각 동영상 프레임의 원영상과 동작추정에 의해 생성된 영상의 각 화소 사이의 MSE를 이용해 계산된 PSNR[dB]을 실험에 적용한 알고리즘별로 나타내었다.

다음 표 3은 제안하는 크로스 패턴과 납작한 육각 패턴 기반 탐색 알고리즘에 대한 구현 결과를 바탕으로 움직임 벡터의 탐색에 필요한 탐색점의 수와 탐색 속도의 성능 향상에 대해 실험 동영상들이 속한 해상도별로 계산된 평균값을 나타내었다.

표 3. 각 해상도별 NS 와 SPEEDUP 결과의 평균값

Table 3. Average Value of NS and SPEEDUP Result of Each Resolution

Method	CIF		SIF		QCIF	
	NS	SPEE-DUP	NS	SPEE-DUP	NS	SPEE-DUP
FS	204.280	1.000	202.050	1.000	184.560	1.000
DS	15.362	13.503	15.365	13.269	12.557	14.834
CDS	12.277	17.254	12.872	15.914	9.848	19.337
HEXBS	11.998	17.153	12.087	16.806	10.298	18.017
PROPOSED	11.973	17.627	12.573	16.250	9.663	19.593

표 3의 각 해상도별 탐색점의 수와 탐색속도 향상에 대한 실험 결과의 평균값들을 통해 제안하는 탐색 알고리즘들을 분석해보면 비교 대상인 CDS 기법에 비해서는 해상도에 관계없이 탐색속도의 향상을 나타내고 있다. 하지만 HEXBS와의 비교결과에서는 움직임이 큰 동영상을 많이 포함하고 있는 SIF 영상의 경우 초기 크로스 패턴의 사용에 의해 탐색점의 수가 소폭 증가한 결과를 나타내었다.

비교대상 기법인 CDS와 HEXBS 기법과 제안한 알고리즘들과의 분석결과를 다음과 같이 표 4로 정리하였으며 표 4의 NS와 SPEEDUP 항목의 값은 백분율로 계산한 값이다.

표 4를 통한 비교 분석 결과 탐색점의 수와 탐색 속도 향상의 측면에 있어서는 크로스 패턴과 납작한 육각 탐색패턴을 이용한 제안 알고리

즘이 탐색속도에 있어 향상을 성능을 나타내었다.

표 4. CDS와 HEXBS와의 해상도별 비교 결과 1

Table 4. Comparison Result 1 of Each Resolution with CDS and HEXBS

Method	CIF		SIF		QCIF	
	NS	SPEED UP	NS	SPEED UP	NS	SPEED UP
CDS와의 비교	2.471	2.160	2.318	2.111	1.878	1.326
HEXBS와의 비교	0.208	2.760	-4.026	-3.308	6.166	8.745

표 4의 비교 결과를 분석해 보면 QCIF 영상의 경우는 해상도가 작은 관계로 움직임 벡터의 탐색영역 중심에 분포할 확률이 다른 영상에 비해 높기 때문에 넓은 탐색영역에 존재하는 움직임 벡터를 탐색하는데 적합한 육각 탐색 기법에 비해 높은 탐색속도 향상을 보여주고 있다.

표 5. 각 해상도별 MAD 와 PSNR 결과의 평균값

Table 5. Average Value of MAD and PSNR Result Average of Each Resolution

Method	CIF		SIF		QCIF	
	MAD	PSNR	MAD	PSNR	MAD	PSNR
FS	3.435	34.255	6.876	26.179	2.621	35.943
DS	3.671	33.871	7.092	25.834	2.762	35.678
CDS	3.788	33.672	7.226	25.557	2.798	35.608
HEXBS	3.799	33.516	7.397	25.468	2.845	35.508
PROPOSED	3.688	33.830	7.151	25.704	2.784	35.629

제안한 알고리즘을 이용하여 복원된 영상의 화질 성능을 비교하기 위해 MAD와 PSNR에 대해서도 실험 동영상들이 속한 해상도별로 계산된 평균값을 표 5에 나타내었다.

표 5의 각 해상도별 MAD와 PSNR에 대한 실험 결과의 평균값들을 통해 제안하는 탐색 알고리즘들을 분석해보면 비교 대상인 CDS 기법과 HEXBS 기법에 대해서 해상도에 관계없이 높은 성능 향상을 보여주었다.

비교대상 기법인 CDS와 HEXBS 기법과 제안한 알고리즘들과의 분석결과를 다음과 같이 표 6으로 정리하였다. 이 때, 표 6의 MAD 항목의 값은 백분율로 계산한 값이고 PSNR은 차이 값을 dB로 표시하였다.

표 6. CDS와 HEXBS와의 해상도별 비교 결과 2

Table 6. Comparison Result 2 of Each Resolution with CDS and HEXBS

Method	CIF		SIF		QCIF	
	MAD	PSNR	MAD	PSNR	MAD	PSNR
CDS와의 비교	2.662	0.158	1.038	0.147	0.494	0.020
HEXBS와의 비교	2.926	0.314	3.330	0.235	2.138	0.121

표 6의 비교 결과를 분석해 보면 해상도와 관계 없이 탐색영역 중심 주위에 분포하는 움직임이 작은 움직임 벡터와 여기에서 찾아지지 않는 움직임이 큰 움직임 벡터를 탐색할 수 있도록 탐색패턴을 설계함으로해서 복원된 화질의 향상에 많은 기여를 하였다.

전체적인 실험 결과를 볼 때, 모든 실험 동영상에 대해서 제안하는 알고리즘이 기존의 탐색 알고리즘들에 비해 탐색점의 수가 감소함으로써 탐색속도의 향상을 가져오는 것을 알 수 있다. 제안 알고리즘 은 CDS에 비해 탐색점의 수에 있어서 약 1.9~2.5%가 의 성능 향상을 나타내었다. 또한 HEXBS에 대해서는 탐색점의 수가 약 0.2~6.2% 정도 감소되었다. 탐색속도 면에서는 약 1.3~8.8% 정도의 성능 향상을 나타내었다.

동영상의 화질에 대한 성능 비교 평가 함수로 사용된 MAD와 MSE를 이용해 계산된 PSNR의 실험 결과에 대해서도 제안하는 알고리즘과 비교를 해보면 CDS에 비해 MAD가 약 0.5~2.7% 정도 감소되었고 PSNR은 약 0.02~0.16dB 정도 향상되었다. 또한 HEXBS에 대해서도 MAD가

약 2.1~3.3% 정도 감소되었고 PSNR에서는 약 0.12~0.31dB 정도의 성능 향상을 나타내었다.

전체적인 실험결과를 보면 제안한 탐색 알고리즘들보다 16~20배 이상 많은 탐색점을 사용하여 동작벡터를 추정하는 전역 탐색과도 화질 면에서 많은 차이가 나지 않으면서 탐색 속도는 향상되었다는 것을 볼 수 있다. 특히 제안하는 알고리즘은 비교대상이 된 HEXBS나 CDS에 비해 화질 면에서 우수한 동작추정 성능을 가지면서, 탐색속도 면에서도 향상된 성능을 나타냈다.

5. 결론

본 논문에서는 동영상 프레임의 시간적 상관성에 따라 움직임 벡터의 분포를 바탕으로 하여 적응적으로 탐색원점과 탐색패턴을 바꾼 새로운 탐색 알고리즘을 제안하였다. 초기에 탐색영역의 중심 주위에 분포하는 움직임 벡터에 대한 빠른 탐색을 위한 크로스패턴과 수평방향으로 많이 분포하는 움직임 벡터의 빠른 탐색을 위한 납작한 육각 탐색 패턴을 결합함으로써 실험을 통하여 알 수 있듯이 제안된 알고리즘은 움직임 추정에 필요한 탐색점의 수를 약 0.2~6.2% 감소시킴으로써 탐색속도 면에서 향상을 보였고, 화질 면에서도 가장 우수한 성능을 가지는 전역 탐색에 근접하는 결과를 얻을 수 있었다. 육각 패턴 탐색과 비교하였을 경우에 움직임 보상 추정된 화질에 있어서 약 0.02~0.31dB 정도의 성능을 향상시켰으며 다른 기존의 탐색 알고리즘들과 비교할 때에도 우수한 성능을 보였다.

본 논문에서는 움직임 벡터의 추정 과정에서 탐색영역에서의 탐색 시작점을 (0,0) 으로 고정시켰다. 인접 블록이나 이전 프레임의 움직임 벡터를 고려하여 탐색 시작점의 좌표를 가변적으로 설정할 수 있는 적응성을 고려하여 본 논문에서 제안한 탐색 알고리즘을 사용하여 움직임 추정을 한다면 보다 빠르게 움직임 벡터를 찾을 수 있을 것이다.

참고 문헌

[1] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated Interframe Coding for Video Conferencing", in Proc. National Telecommunications Conference, New Orleans, LA, pp.G5.3.1-G5.3.5, Nov. 1981.

[2] 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 Technology, Vol. 6, No. 3, pp.313-317, June 1996.

[3] S. Zhu, K. K. 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.

[4] C. Zhu, X. Lin, L. P. Chau, K. P. Lim, H. A. Ang, C. Y. Ong, "A novel hexagon-based search algorithm for fast block motion estimation", Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on , Vol.3, 1593-1596, May 2001.

[5] C. Zhu, X. Lin, L. P. 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.

[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 Technology, Vol. 12, No. 12, pp.1168-1177, Dec., 2002.

[7] C. H. Cheung, L. M. Po, "A Novel Small-Cross-Diamond Search Algorithm

for Fast Video Coding and Videoconferencing Applications", Image Processing. 2002. Proceedings. 2002 International Conference on, Vol. 1, pp.I-681-I-684, Sep., 2002.

[8] H. W. Nam, I. Y. Park, Y. C. Wee, H. J. Kimn, "A New Cross and Hexagonal Search Algorithm for Fast Block Matching Motion Estimation", KIPS, Vol.10-B, No.7, Dec., 2003.

[9] H. W. Nam, Y. C. Wee, H. J. Kimn, "A Fast Block Matching Algorithm by using the Unit-Diamond and Flat-Hexagonal Search Pattern", KISS, Vol.10, No.1, Feb., 2004.

남 현우

1986. 3~1993. 2 : 아주대학교 공과대학 컴퓨터공학과 학사졸업
1994. 3~1996. 2 : 아주대학교 대학원 컴퓨터공학과 석사졸업
1996. 3~1999. 2 : 아주대학교 대학원 컴퓨터공학과 박사수료
2000. 3~현재 : 강남대학교 및 시립인천전문대학 강사
2003. 5~현재 : (주)이테크 기술이사
관심분야 : 컴퓨터그래픽스, 이미지프로세싱, 멀티미디어

김 종경

1980 .3 - 1984. 4 : 광운대학교 전자계산소 주임
1984. 4 - 1999. 7 : 경희대학교 정보처리처 팀장
1999. 3 - 2003. 7 : 인천대학교, 경기대학교 강사
1999. 12 - 현재 : (주) 원포텍 이사
2003 . 3 - 현재 : 시립 인천 전문대학 컴퓨터정보과 겸임교수
1990. 2 : 경희대학교 산업정보대학원 전자계산공학 (공학석사)
1999. 7 : 아주 대학교 일반대학원 컴퓨터공학과 (박사수료)
관심분야 : 멀티미디어 응용 및 시스템구조, 소프트웨어 시스템 디자인, 운영체제