

고속 블록 정합을 위한 납작한 육각패턴 기반 탐색 알고리즘

남현우^o 위영철 김하진
아주대학교 대학원^o 아주대학교 정보통신대학
{manner^o, ycwee, hjkimn}@ajou.ac.kr

A Search Algorithm based on Flat-Hexagon Pattern for the Fast Block Matching

Hyeon-Woo Nam^o Young-Cheul Wee Ha-Jine KIMN
Graduate School, Ajou University^o College of Information Technology, Ajou University

요 약

서로 다른 형태와 크기를 가지는 탐색패턴과 움직임 벡터의 분포는 블록 정합 기법에서 탐색 속도와 화질을 좌우하는 중요한 요소이다. 본 논문에서는 납작한 육각패턴을 이용한 새로운 고속 블록 정합 알고리즘을 제안한다. 이 방법은 작은 육각패턴을 이용하여 적은 탐색점으로 움직임이 적은 벡터를 우선 찾은 다음에 움직임이 큰 벡터에 대해서는 납작한 육각패턴을 이용하여 고속으로 움직임 벡터를 찾게 하였다. 실험결과, 제안된 알고리즘은 육각패턴 탐색기법에 비하여 움직임 벡터 예측의 속도에 있어서 약 11~51% 이상의 높은 성능 향상을 보였으며 화질 또한 PSNR 기준으로 약 0.05~0.74dB 의 향상을 보였다.

1. 서 론

동영상의 빠른 전송 또는 효율적인 저장을 위해서는 동영상 내에 존재하는 시간적, 공간적 중복성을 동영상 분석 기법을 통해 제거하는 압축이 필요하다.

동영상 압축을 위해 움직임을 분석하기 위한 방법으로 화소 재귀적 알고리즘과 블록 정합 알고리즘이 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 단위의 움직임 벡터를 추출하는 블록 정합 알고리즘이 널리 사용되고 있으며 대표적인 알고리즘은 전역 탐색(FS: full search)이다. 전역 탐색은 과정이 간단하고 하드웨어 구현이 용이하며 정합오차가 가장 작은 움직임 벡터를 찾을 수 있지만 많은 계산량이 필요한 단점이 있다. 이러한 전역 탐색의 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], 다이아몬드 탐색(DS: diamond search)[4], 육각패턴 탐색(HEXBS: hexagon-based pattern search)[5], 크로스-다이아몬드 탐색(CDS: cross-diamond search)[6], 작은 크로스-다이아몬드 탐색(SCDS: small-cross-diamond search)[7] 등의 다양한 고속 블록 정합 알고리즘(FBMA: fast block matching algorithm)이 개발되었다. 이들 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(minimum 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 찾는 데 요구되는 계산량을 줄일 수 있도록 초기에 작은 육각패턴을 이용하여 적은 탐색점으로 움직임이 적은 벡터를 탐색하고 이때 찾지 못한 움직임이 큰 벡터에 대해서는 납작한 육각패턴을 이용하여 고속으로 탐색하는 방

법을 제안한다.

2. 움직임 예측과 제안하는 탐색패턴

동영상 압축에 있어서 움직임 벡터를 찾고 그 위치에 있는 이전 블록과의 차이값을 압축/저장하는 시간적인 중복성을 어떻게 찾아내는지의 여부가 효율을 높이는 척도가 된다.

움직임 예측은 계산 복잡도 및 하드웨어 구현에 있어서 용이한 블록 단위의 움직임 예측이 널리 사용된다. 블록 단위의 움직임 예측은 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 벡터로 결정된다.

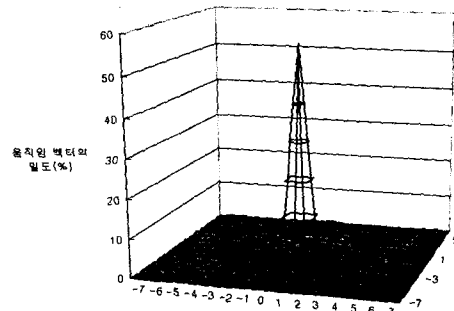


그림 1 움직임 벡터의 분포

동영상에서는 연속하는 두 프레임간의 움직임에 많은 시간적 중복성이 존재하므로 참조 프레임의 움직임 정보를 현재 프레임에서 동일한 위치의 매크로 블록의 탐색 시작점으로 사용함으로써 적은 탐색점들을 사용하여 움직임 벡터를 구할 수 있

고, 양호한 보상결과를 얻을 수 있다.

그림 1과 같이 실험 동영상들에 대해 탐색영역의 거리를 ± 7 로 두었을 때, 움직임 벡터의 분포를 보면 탐색영역의 중심점으로부터 반경 2픽셀 내에 많은 분포를 가지지만 수평방향의 움직임 벡터의 분포가 수직방향의 분포보다 많다. 그러므로 본 논문에서는 탐색영역의 중심점과 중심점으로부터 반경 1픽셀 내의 이웃점으로 구성된 그림 2의 (a)와 같은 작은 육각패턴을 적용하여 초기 탐색점 수를 감소시키며 기존의 육각패턴 탐색에서 발생할 수 있는 탐색영역의 일부가 탐색대상에서 제외되는 현상을 고려한 그림 2의 (b)와 같은 납작한 육각패턴을 제안한다. 이 탐색패턴들을 이용함으로써 움직임 벡터의 빠른 탐색과 화질 향상에 기여하게 된다.

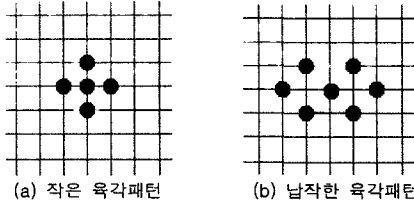


그림 2 작은 육각패턴과 수평 방향(X축)의 움직임 벡터를 고려한 납작한 육각패턴

3. 제안하는 작은 육각패턴과 납작한 육각패턴 탐색 알고리즘

1단계: 탐색 영역의 (0,0)을 중심점으로 그림 7의 (a)와 같이 1픽셀 간격의 4개의 추가점을 포함하는 5개의 탐색점으로 구성된 작은 육각패턴의 탐색점에 대하여 최소 BDM을 계산한다. 중심점이 최소 BDM이면 중심점을 움직임 벡터(MV=(0,0))로 결정하고 탐색을 중단한다. 아니면 2단계로 진행한다.
 2단계: 1단계에서 계산된 중심점이 아닌 최소 BDM 점을 중심점으로 하는 그림 7의 (b)와 같은 납작한 육각패턴을 구성하여 최소 BDM을 계산한다. 새로 계산된 최소 BDM 점이 중심점에 위치하면 4단계로 진행하고, 아니면 3단계로 진행된다.
 3단계: 이전 단계에서 발견된 최소 BDM 점을 중심점으로 지정하고 새로운 납작한 육각패턴을 구성한다. 새로 계산된 최소 BDM 점이 중심점에 위치하면 4단계로 진행하고, 그렇지 않으면 3단계를 반복 수행한다.
 4단계: 최종적으로 작은 육각패턴을 구성하여 최소 BDM을 계산하고, 이 단계에서 구해진 최소 BDM 점이 움직임 벡터의 최종 해가 된다.

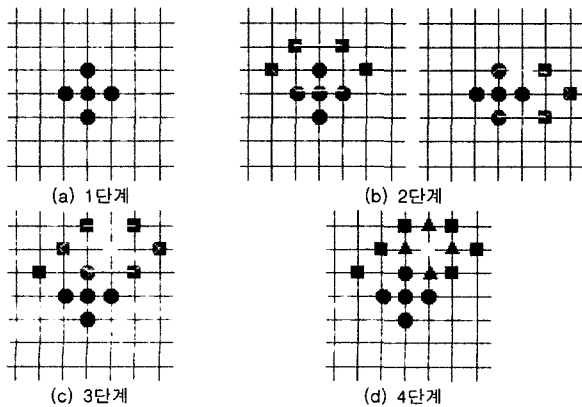


그림 3 제안하는 알고리즘의 탐색경로

4. 실험 결과

제안된 기법의 성능을 평가하기 위하여 2개의 CIF(352X288) 영상 Akiyo, Foreman과 SIF(352X240) 영상 Football, 그리고 QCIF(176X144) 영상 Table에 대해 각각 100프레임씩을 대상으로 실험하였고, 비교 탐색 기법으로는 FS, DS, CDS, SCDS, HEXBS, 그리고 제안한 탐색기법을 사용하였다. 움직임 예측에 사용된 매크로 블록의 크기는 16x16 픽셀이며, 탐색영역의 변위는 ± 7 를 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 이용하였으며, 정합 오차 측정 함수로는 절대값 오차의 합(SAD: sum of absolute difference)을 이용하였다. 또한 제안하는 기법의 성능 향상을 측정하기 위해 블록 당 탐색점의 수를 기존 방법들과 비교하였다.

실험영상에 대한 실험 결과는 표 1, 그림 4, 그리고 그림 5에 각각 나타내었다. 표 1에서는 실험영상별로 전체 프레임에 대한 기존의 탐색기법과 제안하는 탐색기법의 실험결과를 나타내었으며 그림 4과 그림 5에서는 두 실험영상의 프레임별 블록 당 평균 탐색점 수와 프레임별 픽셀 당 MAD 값을 각각 그래프로 나타내었다.

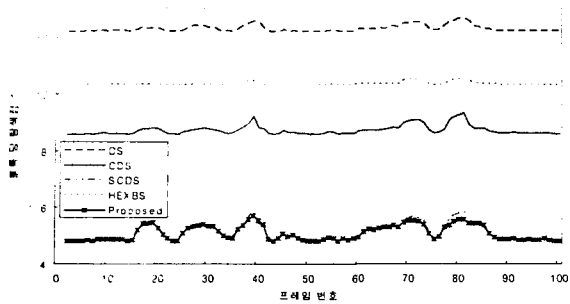
표 1 각 실험영상에 대한 실험결과 (a) 성능 비교 평가 함수의 결과값

BMA	Akiyo (CIF)		Foreman (CIF)		Football (SIF)		Table (QCIF)	
	MAD	PSNR	MAD	PSNR	MAD	PSNR	MAD	PSNR
FS	0.605	42.173	2.909	32.888	10.242	22.392	5.021	26.076
DS	0.606	42.150	2.998	32.619	10.758	21.857	5.160	25.745
CDS	0.606	42.127	3.086	32.215	11.207	21.494	5.229	25.597
SCDS	0.606	42.127	3.087	32.207	11.259	21.433	5.259	25.524
HEXBS	0.619	41.849	3.279	31.887	11.117	21.628	5.506	25.278
Proposed	0.606	42.147	3.015	32.486	11.046	21.567	5.259	25.532

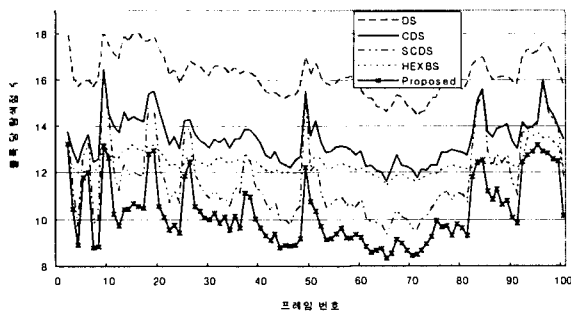
(b) 블록 당 탐색점 수와 속도향상 비율

BMA	Akiyo (CIF)		Foreman (CIF)		Football (SIF)		Table (QCIF)	
	Ns	SpUp	Ns	SpUp	Ns	SpUp	Ns	SpUp
FS	204.28	1.000	204.28	1.000	202.05	1.000	184.56	1.000
DS	12.29	16.628	15.99	12.775	16.17	12.494	13.83	13.344
CDS	8.72	23.422	13.48	15.155	13.41	15.063	11.63	15.863
SCDS	5.08	40.250	12.52	16.316	11.66	17.328	10.19	18.110
HEXBS	10.35	19.744	12.36	16.534	12.38	16.318	11.19	16.490
Proposed	5.08	40.227	10.98	18.610	10.27	19.674	8.99	20.522

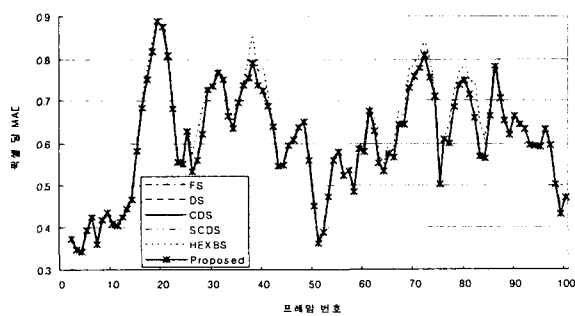
표 1의 실험결과를 볼 때, 모든 실험영상에 대해서 제안하는 탐색기법이 기존의 탐색기법들에 비해 탐색점의 수가 현저하게 감소함으로써 탐색속도의 향상을 가져오는 것을 알 수 있다. 또한 영상의 화질에 대한 성능 비교 평가 함수로 사용된 MAD와 MSE를 이용해 계산된 PSNR의 실험결과를 볼 때, 제안한 탐색기법은 많은 탐색점을 필요로 하는 전역 탐색과도 많은 차이가 나지 않으며 비교대상이 된 육각패턴 탐색이나 크로스-다이아몬드 탐색에 비해서도 근사한 움직임 예측성능을 가지면서, 탐색속도 면에서는 우수한 성능 향상을 보였다.



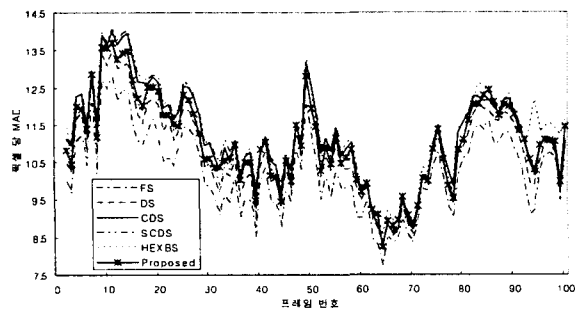
(a) 움직임이 적게 포함된 Akiyo(CIF) 영상



(b) 움직임이 많이 포함된 Football(SIF)영상
그림 4 실험영상의 프레임별 블록 당 평균 탐색점 수



(a) 움직임이 적게 포함된 Akiyo(CIF) 영상



(b) 움직임이 많이 포함된 Football(SIF)영상
그림 5 실험영상의 프레임별 픽셀 당 MAD

그림 4의 결과를 통해 프레임간의 시간적 상관성이 많이 존재하는 움직임이 적게 포함된 영상이거나 프레임 간에 존재하는 시간적 상관성이 상대적으로 적은 움직임이 많이 포함된 영

상일지라도 기존의 탐색기법에 비해 움직임 예측에 필요한 탐색점의 수가 현저하게 감소된 것을 볼 수 있으며 이는 탐색속도 측면에서 높은 성능 향상을 보인다는 것을 알 수 있다.

또한 그림 5의 결과를 볼 때, 전역 탐색과 차이가 작은 탐색기법이 움직임 벡터 예측성능이 우수함을 나타내는 것으로서 움직임이 많이 포함된 영상과 움직임이 적게 포함된 영상들이 조금씩 다른 결과를 나타내지만 결과적으로는 제안하는 탐색기법이 크로스-다이아몬드 탐색을 적용한 결과와 근소한 차이만을 가지며 육각패턴 탐색보다는 우수한 실험결과를 나타냈다.

5. 결론

본 논문에서는 움직임 벡터의 분포를 바탕으로 하여 초기에 작은 육각패턴을 이용한 중심 주위에 분포하는 움직임 벡터의 빠른 탐색과 납작한 육각패턴을 이용한 수평방향으로 많이 분포하는 움직임 벡터의 빠른 탐색을 결합함으로써 탐색속도 면에서 높은 향상을 보였고, 화질 면에서도 가장 우수한 성능을 가지는 전역 탐색에 근접하는 결과를 얻을 수 있었다.

움직임 벡터의 예측 과정에서 발생할 수 있는 예외적인 요소를 모두 고려하여 가장 적응적인 현재 프레임의 매크로 블록 탐색원점을 예측하고 본 논문에서 제안한 탐색기법을 사용하여 움직임 예측을 한다면 보다 빠르게 움직임 벡터를 찾을 수 있을 것이며, 우수한 보상 결과를 얻을 수 있을 것으로 기대된다.

참고문헌

- [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] R. Li, Bing Zeng, "A New Three-Step Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Circuits & System for Video Technology, 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 Technology, Vol. 6, No. 3, pp.313-317, June 1996.
- [4] 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.
- [5] Ce Zhu, Xiao Lin, 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.
- [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.1-681-1-684, Sep., 2002.