

예측 움직임 벡터와 블록 정합 오류 특성을 이용한 고속 움직임 추정 알고리즘

정봉수¹, 전병우²
성균관대학교, 정보통신 공학부

Fast Motion Estimation Algorithm using Predictive Motion Vector and Block Matching Error Characteristics

Bongsoo Jung, Byeungwoo Jeon
Dept. of Information and Computer Engineering, Sungkyunkwan University
E-mail : bsjung@ece.skku.ac.kr¹, bjeon@yurim.skku.ac.kr²

Abstract

움직임 벡터의 상관도, 움직임 벡터의 분포특성, 블록 정합 오류의 특성은 탐색 패턴과 탐색 방법을 결정하는 중요한 요소이다. 일반적으로 움직임 벡터는 주로 탐색영역의 가운데를 중심으로 수평 혹은 수직축에 주로 분포한다. 또한 탐색 영역 내의 정합 오류 값의 분포를 보면 움직임 벡터의 분포와 비슷한 형태로 정합 오류의 값들이 수평 혹은 수직 방향으로 최소 정합 오류 값의 위치로 단조 감소해 나간다. 본 논문에서는 이러한 블록 정합 오류의 특성을 이용한 새로운 탐색 방법을 제안 하며 주변 블록의 움직임 벡터의 상관도를 이용하여 초기 탐색 지점을 선택하는 고속 움직임 추정 알고리즘을 제안한다. 또한 모의실험을 통하여 기존의 여러 움직임 추정 알고리즘과 비교하여 PSNR 의 감소는 거의 없으면서 매크로블록당 평균 탐색포인트와 수행 시간의 향상을 얻을 수 있음을 확인한다.

H.261/263/264 같은 여러 영상 압축 표준에서 프레임간의 시간적 중복성을 없애기 위한 중요한 방법으로써, 동영상 압축효율에 가장 큰 영향을 주는 부분이다. 하지만 움직임 추정은 부호화기에서 가장 많은 계산량을 요구하는 부분이기도 하다. 움직임 추정 방법 중 전역 탐색(Full search)방법은 탐색범위 내에서 모든 위치를 탐색하기 때문에 최적의 움직임 벡터를 찾을 수 있지만 계산량이 많이 요구 되어 실시간 동영상 압축 시스템에 적용하기에는 문제가 있다. 따라서 이를 개선하기 위해 많은 고속 움직임 추정 알고리즘들이 제안되었다. 이중 탐색점을 줄이는 방법으로 3 단계탐색(TSS) [1], 새로운 3 단계탐색(NTSS) [2], 4 단계탐색(4SS) [3], 다이아몬드탐색(DS) [4], 그리고 CDS (Cross-Diamond Search) [5] 알고리즘들이 있다. 또한 주변 매크로블록의 움직임 벡터의 상관도를 이용한 MVFAST[8]와 PMVFAST[9] 알고리즘들도 있다.

블록 정합 움직임 추정 방식에서 영상의 움직임 벡터는 대부분 탐색영역의 중심에 분포한다. 논문 [7]의 실험에 따르면, 부드러운 움직임 영상의 경우 5x5의 탐색영역의 중심에 약 80%의 움직임 벡터가 분포한다. 따라서 이러한 움직임 벡터의 특성을 이용한 다이아몬드 탐색 알고리즘이 제안 되었다[4]. 또한 논문 [5]에서는 5x5의 탐색 영역의 중심에 분포한 움직임 벡터들의 대략 96%가 십자형태의 위치에 분포하는 것을 확인 하였

I. 서론

움직임 추정 및 보상은 MPEG-1/2/4 와 ITU-T

*본 논문은 2003년도 산업기반 기술개발사업(NiPC) 연구비 지원에 의한 것임.

다. 따라서 기존의 다이아몬드 탐색보다 탐색 격자점의 수를 줄인 십자형 탐색 패턴을 사용한 십자형 다이아몬드 탐색 방법(Cross-Diamond Search)이 제안되었다[5]. 이들 방식은 급격한 움직임이 있는 영상의 경우에는 움직임 벡터가 탐색 범위인 5x5 를 벗어나는 경우가 많이 발생하여 적합 하지 않다.

본 논문에서는 주변 블록의 움직임 벡터의 상관성을 이용한 초기 탐색 지점 선택과 움직임 벡터 분포와 블록 정합 오류의 특성을 이용한 새로운 탐색 방법을 적용한 고속 움직임 추정 알고리즘을 제안한다.

II. 제안 움직임 추정 알고리즘

대부분 블록 정합 움직임 추정에서 가장 정합이 잘 되는 위치를 결정하기 위해 블록 정합 오류 평가기준으로 SAD(Sum of Absolute Differences)를 사용하고 있다. (x, y) 에 위치하는 $N \times N$ 크기의 블록에 대한 SAD 는 식(1)과 같이 계산된다.

$$SAD(dx, dy) = \sum_{m=x}^{x+N-1} \sum_{n=y}^{y+N-1} |I_t(m, n) - I_{t-1}(m+dx, n+dy)| \quad (1)$$

여기서 I_t 는 현재 프레임, I_{t-1} 는 이전 프레임 이며, (dx, dy) 는 (x, y) 로 부터 변위 즉 움직임 벡터이다.

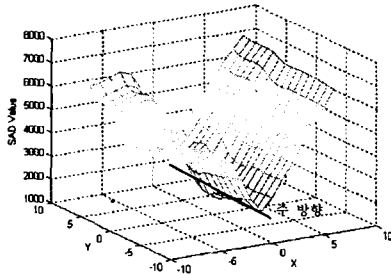


그림 1 탐색 영역 에서 블록 매칭 에러 값의 분포 예

그림 1 은 탐색 영역 내에서 전역 탐색을 수행하여 각 위치의 SAD 값의 분포를 나타낸 것이다. 그림 1 에서 보는 것과 같이 블록 정합 오류를 나타내는 SAD 값의 분포는 특정 방향성분을 나타내면서 최적의 예러 지점으로 단조롭게 감소하는 형태를 나타낸다. 이 방향은 움직임 벡터의 분포와 비슷한 형태를 보여 준다. 논문 [6] 에서 여러 영상에 대하여 탐색 범위 내에서 전역 탐색을 수행하여 움직임 벡터 분포의 실험 결과를 보면 가운데를 중심으로 십자형태의 분포를 나타낸다. 따라

서 제안 알고리즘의 탐색 방법은 기존의 다이아몬드 탐색(DS)이나 십자형 다이아몬드 탐색(CDS) 방법과 달리 초기에는 작은 다이아몬드와 같은 5 포인트를 탐색하여, 중심의 SAD 가 최소가 아니면 주변 4 포인트 중의 최소인 지점의 방향으로 1 포인트를 확장하여 탐색한다. 이때 확장 포인트가 더 이상 최소 SAD 값으로 수렴 하지 않으면, 상/하 혹은 좌/우의 탐색하지 않은 지점을 탐색하여 중심 지점이 최소 SAD 값을 가질 때까지 탐색한다. 그림 2 는 제안된 탐색 패턴의 예를 보여주고 있다.

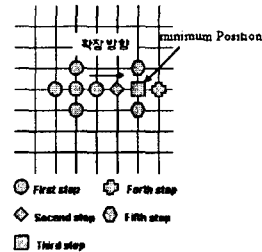


그림 2 제안된 탐색 패턴

제안된 고속 움직임 추정 알고리즘은 새로운 탐색 패턴을 수행 할 때 탐색 영역의 중심부터 탐색하지 않고 현재 블록의 움직임 벡터를 예측하여 예측 지점을 중심으로 탐색을 수행한다. 현재 블록의 움직임 벡터를 예측하기 위하여 그림 3 과 같이 좌측(MV1), 상(MV2), 우상(MV3), 이전 프레임에 대응되는 움직임 벡터 ($MV_{collocated}$), 제로 움직임 벡터, 그리고 MV1, MV2, MV3 의 중간값의 움직임 벡터(MV_{median})를 후보 움직임 벡터로 사용하여 최소의 SAD 값을 가지는 벡터를 예측벡터(Predictive MV)로 선택한다. 여기서 MVFAST 와 같이 $(0,0)$ 지점의 SAD 값이 문턱치값 ($T=512$) 보다 적으면 탐색을 중단 하고 최적 움직임 벡터를 $(0,0)$ 으로 한다. 또한 PMVFAST 와 비슷하게 MV_{median} 와 $MV_{collocated}$ 가 같고, MV_{median} 위치의 SAD 값이 $MV_{collocated}$ 위치의 SAD 값보다 작으면 탐색을 중단하고 중간값 움직임 벡터를 최적의 벡터로 한다.

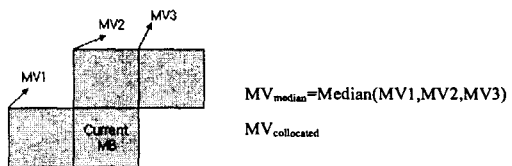


그림 3 후보 움직임 벡터

본 논문에서 제안된 움직임 탐색 알고리즘의 수행 순서는 다음과 같이 요약된다.

i) Step 1: 그림 3의 후보 움직임 벡터(MV_{median} , zero MV , $MV_{collocated}$, $MV1$, $MV2$, $MV3$) 위치의 SAD 값을 계산한다. 이때, (0,0)의 SAD 값이 경계치 T (본 논문에서는 512 사용) 보다 작으면 탐색을 중단한다. 또는 MV_{median} 와 $MV_{collocated}$ 이 같고, MV_{median} 위치의 SAD 값이 $MV_{collocated}$ 위치의 SAD 값보다 작으면 탐색을 중단한다. 여기서 예측 움직임 벡터(Predicted MV)를 찾는다.

ii) Step 2: Step1에서 찾은 예측된 움직임 벡터를 중심으로 그림 2의 동,서,남,북 각 주변위치의 SAD를 계산한다. 이때 예측된 움직임 벡터 지점의 SAD가 최소이면 탐색을 중단하고 그렇지 않으면 다음 단계를 수행한다.

iii) Step 3: 최소 SAD 지점의 확장 방향 1 포인트의 SAD를 계산한다. 이때 확장 방향 1 포인트의 SAD 값이 감소하지 않을 때까지 최소 SAD 지점을 갱신하면서 탐색을 수행한다. SAD 값이 감소하지 않으면 Step 4를 수행한다.

iv) Step 4: Step 3의 확장 방향에 수직으로 상/하 혹은 좌/우의 2 포인트를 탐색한다. 이때 중심 위치가 최소이면 탐색을 중단하고 그렇지 않으면 최소 SAD 지점을 갱신하고 Step 3을 수행한다.

그림 4는 제안된 알고리즘의 수행 과정의 예를 그림으로 나타내었다.

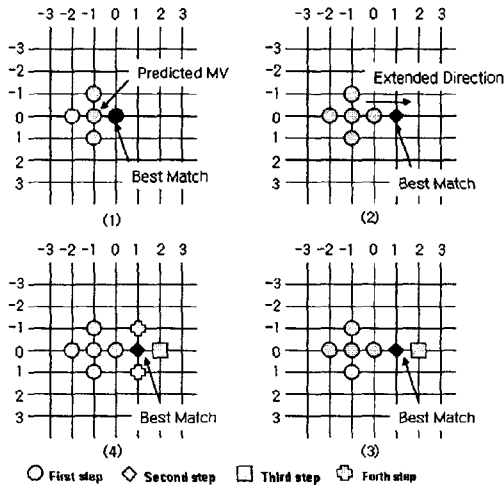


그림 4 제안된 알고리즘의 수행 예(Predicted MV=(0,-1))

III. 실험 결과

제안된 고속 움직임 추정 알고리즘의 성능 평가를 위해 MPEG-4 부호화기 참조 소프트웨어에 기존의 다이아몬드 탐색(DS), 십자형 다이아몬드 탐색(CDS), MVFAST, PMVFAST를 구현하여 성능을 비교하였다. 그리고 표 1에 나타난 실험영상과 조건하에 실험 하였다. 객관적 화질 평가를 위해서 PSNR(peak signal to noise ratio)를 사용하였고 계산량 비교를 위하여 매크로블록당 평균 탐색 포인트의 수를 비교 하였다. 또한 계산량의 비교에서 실제 수행 시간의 비교가 매우 중요하므로 팬텀 4/2.4GHz, WindowsXP에서 각각 5번을 수행한 후 수행시간의 평균을 측정하였다.

표 3은 제안 알고리즘과 전역 탐색(FS), 다이아몬드 탐색(DS), 십자형 다이아몬드 탐색(CDS), MVFAST, PMVFAST의 각 알고리즘을 PSNR과 평균 매크로 블록의 탐색 포인트, 그리고 소프트웨어 수행 시간을 나타내었다. 표 3을 보면 제안된 고속 움직임 추정 알고리즘이 기존의 고속 움직임 알고리즘과 비교하여 PSNR의 손실이 거의 없으면서 매크로블록당 탐색 포인트와 수행시간의 성능 향상을 나타내었다. 평균 매크로블록당 탐색 포인트는 십자형 다이아몬드 탐색과 MVFAST에 비교하여 대략 70%와 40%의 계산량 감소를 가져오며 전역 탐색 알고리즘과 비교하여 250배 이상의 탐색 포인트 감소를 가져온다. 제안 알고리즘을 PMVFAST 알고리즘과 비교하면 평균 10%미만의 탐색 포인트 감소를 가져 오는데, 이는 Hall monitor와 Mother and daughter 영상과 같이 움직임이 수평이나 수직에 많이 분포하지 않는 경우는 오히려 더 많은 탐색을 하기 때문이다. 하지만 알고리즘의 복잡도에서 PMVFAST보다 간략하기 때문에 실제 수행시간은 더 적은 시간을 나타내었다. MVFAST와 PMVFAST 알고리즘과 수행시간을 비교하면 대략 35.5%와 11.5%의 성능 향상이 있다.

표 2. 실험 조건

Search range	± 16
Rate control	VM5+
Image (format, Hz, bps)	Hall monitor (QCIF, 30Hz, 10kbps), Mother and daughter (QCIF, 30Hz, 24kbps), Silent (QCIF, 30Hz, 24kbps), Coastguard(CIF, 30Hz, 112kbps), Mobile and calendar(CIF, 30Hz, 300kbps) Foreman(CIF, 30Hz, 512kbps), Stefan(CIF, 30Hz, 2048kbps)

표 3 제안된 알고리즘의 성능 비교

Sequence		FS	DS	CDS	MVFAST	PMVFAST	Proposed
Hall monitor	PSNRY(dB)	29.65	29.62	29.56	29.61	29.62	29.66
	Search point per MB	1024.00	13.05	9.13	4.06	2.45	2.60
	Execution time(ms)	95.342	2	1.424	0.972	0.708	0.698
Mother and daughter	PSNRY(dB)	33.37	33.32	33.29	33.24	33.27	33.26
	Search point per MB	1024.00	13.23	9.61	4.12	2.79	2.85
	Execution time(ms)	95.318	2.016	1.692	1.092	0.794	0.594
Silent	PSNRY(dB)	35.11	35.12	35.1	35.08	35.04	35.13
	Search point per MB	1024.00	13.76	10.36	6.04	3.88	3.74
	Execution time(ms)	101.674	2.154	1.85	1.264	0.866	0.813
Coastguard	PSNRY(dB)	33.37	33.32	33.29	33.24	33.27	33.26
	Search point per MB	1024.00	16.15	14.18	9.97	6.57	5.18
	Execution time(ms)	410.416	9.696	8.804	6.794	4.492	3.943
Foreman	PSNRY(dB)	33.75	32.98	32.96	33.49	33.79	33.52
	Search point per MB	1024.00	18.65	17.29	8.45	5.35	5.27
	Execution time(ms)	417.07	11.016	9.696	5.784	4.538	4.224
Stefan	PSNRY(dB)	33.86	32.45	32.45	33.67	33.9	33.8
	Search point per MB	1024.00	19.83	18.91	8.73	5.34	4.94
	Execution time(ms)	461.526	12.278	11.752	6.234	4.714	4.024
AVERAGE	PSNRY(dB)	33.19	32.80	32.78	33.06	33.15	33.11
	Search point per MB	1024	15.78	13.25	6.89	4.40	4.10
	Execution time(ms)	263.56	6.53	5.87	3.69	2.69	2.38

III. 결론

본 논문에서는 움직임 벡터의 공간적인 상관도를 이용한 예측 움직임 벡터와 블록 정합 오류 특성을 이용한 고속 움직임 추정 알고리즘을 제안하였다. 제안 알고리즘은 실험을 통하여 기존의 여러 고속 움직임 탐색 알고리즘과 비교하여 PSNR 감소가 거의 없으면서 탐색 포인트와 실제 수행시간이 많이 감소하는 것을 확인하였다. 전역 탐색 알고리즘과 비교하여 평균 PSNR 은 0.08dB 의 감소가 있지만 탐색 포인트는 대략 250 배 이상 감소하고 수행 시간은 대략 110 배 감소하였다.

참고문헌

[1] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing", in *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, pp.G5.3.1-G5.3.5, Nov 1981.

[2] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol.4, pp.438-443, Aug. 1994.

[3] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans.*

Circuits Syst. Video Technol., vol.4, pp.313-317, Aug. 1994.

[4] Shan Shu and Kai-Kuang Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation," *IEEE Trans. on Image Processing*, Vol. 9, no. 2, pp.287-290, Feb 2000.

[5] Chun-Ho Cheung and Lai-Man Po, "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation," *IEEE Trans. Circuit Syst. Video Technol.*, Vol. 12, no. 12, pp.1168-1177, Dec 2002.

[6] K.-K. Ma and G Qiu, "Unequal-arm adaptive rood pattern search for fast block-matching motion estimation in the JVT/H.26L," *IEEE Internat. Conf. on Image Processing*, Spain, Sep. 2003.

[7] Jo Yew Tham, Ranganath, S, Ranganath and M. Kassim, A.A, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans.Circuits Syst. Video Technol.*, Vol. 8, no. 4, pp.369-377, Aug 1998.

[8] P. I. Hosur and K. K. Ma, "Motion vector field adaptive fast motion estimation," *Proc. Second Intern. Conf. on Information, Communications and Signal Processing(ICICS'99)*, Singapore, Dec. 1999.

[9] A.M. Tourapis, O.C.Au, and M.L.Liou, "Predictive motion vector field adaptive search technique (PMVFAST)-Enhancing block based motion estimation," *Proc. SPIE Visual Commun. Image Process.*, San Jose, CA, Jan. 2001.