

움직임 분류와 직접 탐색 패턴을 통한 고속 블록 움직임 추정 알고리즘

박순철, 후메라 리사, 최태선
광주과학기술원

e-mail : spark@gist.ac.kr, humaira@gist.ac.kr, tschoi@gist.ac.kr

A Fast Block Motion Estimation Algorithm Based On Motion Classification And Directional Search Patterns

Soon-Chul Park, Humaira Nisar, Tae-Sun Choi
Gwangju Institute of Science and Technology

Abstract

This paper suggests a simple scheme of block motion estimation in which the search pattern selection is based on the classification of motion content available in the spatio temporal neighboring blocks. The search area is divided into eight sectors and the search pattern selection is also based on the direction of predicted motion vector. Experimental results show that the proposed algorithm has achieved good predicted image quality measured in terms of PSNR and has very less computational complexity.

I. 서론

추정 및 보상은 여러 영상 압축 표준에서 많이 이용되고 있다. 블록 정합 알고리즘(Block Matching Algorithm)중에 전역 탐색(Full Search)방법은 하드웨어 구현이 용이하기 때문에 비디오 코딩 응용에 널리 이용되고 있지만 반면에 많은 계산량이 요구된다. 따라서 이를 개선하기 위해 많은 고속 블록 정합 움직임 추정 알고리즘들이 제안 되었다[1]-[2]. 본 논문에서는 영상에서 시공간적 상관성을 동시에 고려하여 초기 탐

색 지점을 선택하는 예측 움직임 추정 기술을 제안한다.

II. 제안한 알고리즘

시공간적인 상관 정보(spatio temporal correlation information)를 사용하여 움직임 벡터를 예측할 수 있다[1]. 첫 단계로 탐색의 초기 지점을 선택하기위해 움직임 벡터들을 탐색한다. 그림 1 과 같이 현재 블록의 움직임 벡터를 예측하기 위하여 좌(MV_n(SL)), 상(MV_n(SA)), 이전 프레임에 대응되는 우(MV_{n-1}(TR)), 하(MV_{n-1}(TB)), 그리고 움직임 벡터(MV_{n-1}(P))의 중간 값을 후보 움직임 벡터로 사용하여 최소 SAD(Sum of Absolute Difference) 값을 가지는 벡터를 예측벡터(P(X_n))로 선택한다.

$$P(X_n) = \text{median}(MV(SL), MV(SA), MV(TR), MV(TB), MV(P)) \quad (1)$$

첫 단계에서 예측벡터를 구했으므로 초기 탐색영역에서 예측된 지점으로 넘어간다.

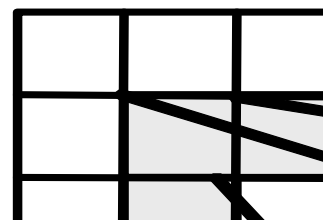


그림 1. 시공간 상관 정보에 대한 인접 블록들

블록의 움직임 내용을 정의하기 위해서 예측 움직임 벡터의 크기를 사용한다. 실험의 탐색 범위는 + 7로 결정하였다. 따라서 움직임 벡터의 x, y요소의 크기도 0에서 7까지 변한다. 블록들은 움직임 정도에 따라 움직임이 없는 블록, 작은 블록($P(X_n) \leq 1$), 중간정도 블록($P(X_n) \leq 3$), 그리고 큰 블록($P(X_n) > 3$)으로 분류하였다.

예측정보는 현재 블록의 움직임 방향을 예측하기 위해 사용되고, $P(X_n)$ 와 $MV_{n-1}(P)$ 로부터 구할 수 있다. 그림 2(a)에서 나타내는 것처럼 전체 공간은 8개의 구역(sector)으로 나뉜다. 탐색 패턴은 4가지 경우와 8개 구역으로 나누어진다.

움직임이 없는 블록과 작은 블록에 있는 어떤 움직임을 찾기 위한 알고리즘이 그림 2 (b)에서 나타난다. 먼저 탐색 중앙의 주위의 4 포인트를 체크한다. 만약 중심 포인트가 최소 SAD이면 탐색을 중단한다. 그렇지 않다면 다음단계로 가서 새로운 3 포인트를 체크하고 나서 탐색을 중단한다. 중간 블록과 큰 블록인 경우 $P(X_n)$ 의 관계와 움직임의 방향은 다음과 같이 정의한다:

- Sector I: $-22.5 < \text{ang} < 22.5$, Sector II: $22.5 < \text{ang} < 67.5$,
- Sector III: $67.5 < \text{ang} < 112.5$, Sector IV: $112.5 < \text{ang} < 157.5$,
- Sector V: $157.5 < \text{ang} < 202.5$, Sector VI: $202.5 < \text{ang} < 247.5$,
- Sector VII: $247.5 < \text{ang} < 292.5$, Sector VIII: $292.5 < \text{ang} < 337.5$.

움직임은 일반적으로 수평, 수직방향으로 발생하므로 구역의 경계를 0-45도로 정의하는 대신에 -22.5에서 22.5도로 정의 하였다. 첫 단계의 방향 탐색 패턴은 그림 2 (c)에 나타났다. 두 번째 단계에서는 탐색 중앙 주위에 수직, 수평으로 위치한 포인트들을 확인한다. 전체 과정을 그림 2 (d)에 나타내었다.

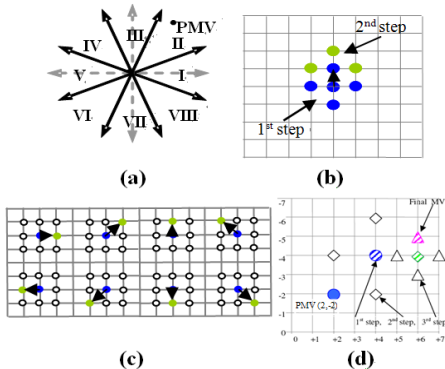


그림 1 (a) $P(X_n)$ 의 움직임 방향에 관한 구역의 구성 (b) 움직임이 없거나 작은 블록에 대한 탐색 방법 (c) 움직임 정도가 중간이거나 큰 블록에 대한 방향 탐색 패턴 (d) 움직임 정도가 중간이거나 큰 블록에 대한 전체 탐색 패턴 과정

III. 실험

제안된 알고리즘은 다양한 영상으로 실험하였다. 블록의 크기는 16x16화소이고, 탐색 영역의 크기는 ± 7

이다. SAD는 블록 왜곡 지표(BDM)으로써 사용된다. 제안된 알고리즘은 Full Search(FS), Three Step Search(TSS), Four Step Search(FSS), Diamond Search(DS), Adaptive Road Pattern Search(ARPS)와 같은 잘 알려진 움직임 추정 알고리즘들에 대하여 비교하였다. 표 1 및 2에는 평균 PSNR 및 움직임 블록에 대한 평균 NSP(number of search points)와 speed up ratio의 성능 비교를 나타내었다.

표 1 PSNR 비교

Carphone, Foreman(QCIF-300),Hall-Monitor, Coastguard (CIF-300), Salesman, Ms Am(CIF-150), Flower (SIF-150), Mobile (SIF-100)

Sequence	FS	TSS	FSS	DS	ARPS	Proposed
Carphon	31.4	30.72	30.85	30.72	31.07	30.89
Foreman	31.27	30.66	30.73	30.78	30.87	31.29
Coastgua	30.43	26.87	29.73	30.25	30.40	30.37
Salesman	35.38	34.63	34.96	34.86	35.32	35.22
Miss Am	37.71	36.14	36.62	36.82	37.59	37.38
Hallmoni	34.80	34.51	34.60	34.60	34.74	34.66
Flower	23.97	23.29	21.2	23.59	23.78	23.79
Mobile	22.67	22.40	22.36	22.50	22.65	22.64
Average	30.95	29.9	30.13	30.52	30.80	30.78

표 2 평균 NSP

Sequence	FS	TSS	FSS	DS	ARPS	Proposed	Speedup w.r.t. FS
Carphone	184.6	21.62	16.04	15.89	5.98	5.82	31.71
Foreman	184.6	21.68	15.59	16.47	8.22	5.72	32.27
Coastgua	204.3	23.38	12.64	17.87	8.72	7.67	26.63
Salesman	204.3	23.21	17.28	15.67	6.35	5.28	38.69
Miss Am	204.3	23.46	19.3	16.14	8.4	6.28	32.53
Hallmonit	204.3	23.26	16.69	15.62	6.06	5.36	38.11
Flower	202.4	23.23	14.15	17.79	8.89	6.65	30.44
Mobile	202.4	23.02	17.82	15.89	7.43	5.03	40.24
Average	198.9	22.86	16.19	16.42	7.51	5.98	33.26
Av. Speed up w.r.t. FS	1.0	8.70	12.29	12.11	26.50	33.29	---

IV. 결론

본 논문에서는 고속 블록 정합 움직임 추정 알고리즘을 제안하였다. 시공간 상관 정보 기반의 제안된 알고리즘은 화질과 계산속도면에서도 좋은 성능을 나타냈다.

참고문헌

[1] K. R. Namuduri, Senior Member, IEEE, "Motion estimation using spatio-temporal contextual information", IEEE Transactions On Circuits And Systems For Video Technology, Vol. 14, No. 8, August 2004.
 [2] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. on Communications, Vol. COM-29, No. 12, pp. 1799-1808, Dec 1981.