

예측 움직임 벡터를 이용한 고속 블록 추정 알고리즘

김진욱, 박태근

가톨릭대학교 정보통신공학부

e-mail : white3479@hanmail.net, parktg@catholic.ac.kr

A Fast Motion Estimation Algorithm using Predictive Motion Vector

Jin-Wook Kim, Tae-Geun Park

School of Information and Communication Engineering

Catholic University of Korea

Abstract

움직임 추정은 비디오신호의 압축에 중요한 역할을 한다. 본 논문은 효율적으로 움직임 벡터를 찾기 위하여 움직임 벡터의 시간적, 공간적 유사성을 이용하였다. 벡터를 검색하기 이전에 움직임 벡터의 검색 범위를 크게 9개의 영역으로 나눈 후, 이전 프레임에서 동일한 위치, 현재 프레임의 현재 매크로블록의 상위, 상우와 좌측의 매크로블록에서의 움직임 벡터까지 총 4개의 움직임 벡터를 이용하여 9개의 영역 중 한 영역을 제 1 후보, 그를 둘러싼 영역을 제 2 후보라 정하고 극소점들(Local Minima)을 피하였다. 모의실험을 통한 결과 NDS(New Diamond Search) 알고리즘에 비하여 매크로블록 당 평균 탐색 포인트 수가 5.79 포인트 감소하고, MSE는 최대 104.23 감소한 것을 확인할 수 있었다.

I. 서론

통신 및 방송 분야에서, 고품질 TV, 주문형 비디오, 화상회의 그리고 CD-ROM 보관기록 등에서와 같이

디지털 비디오 및 오디오 신호를 제한된 대역의 채널을 통하여 전송하거나 저장매체에 저장하기 위해서는 이를 충분히 압축할 수 있는 부호화가 필요하다.

움직임 추정 및 보상은 MPEG-1/2/4[1-2] 와 ITU-T H.261/263/264[3-4] 같은 여러 영상 압축 표준에서 프레임간의 시간적 중복성을 없애기 위한 중요한 방법으로서, 동영상 압축효율에 가장 큰 영향을 주는 부분이다. 하지만 움직임 추정은 부호화기(Encoder)에서 가장 많은 계산량을 요구하는 부분이기도 하다.

움직임 추정은 블록 매칭 알고리즘(BMA)을 이용하여 수행된다. 이 블록 매칭 알고리즘 중 전역탐색(Full search)방법은 탐색범위 내에서 모든 위치를 탐색하기 때문에 최적의 움직임 벡터를 찾을 수 있으나 계산량이 많이 요구되어 실시간 동영상 압축 시스템에 적용하기에는 문제가 있다. 따라서 이를 개선하기 위해 많은 고속 움직임 추정 알고리즘들이 제안되었다. 이 중 탐색 포인트를 줄이는 방법으로, 넓은 범위에서 점차적으로 좁게 탐색하는 3단계탐색(TSS)[5], 다이아몬드 탐색을 적응적인(adaptive) 탐색으로 발전시킨 뉴다이아몬드탐색(NDS)[6], 탐색 포인트 수는 앞의 알고리즘들 보다 줄인 CSA[7] 알고리즘 등이 있다.

본 논문에서는 효율적으로 움직임 벡터를 찾기 위한 방법으로 움직임 벡터의 시간적, 공간적 유사성을 이용하였다. 벡터를 검색하기 이전에 극소점들을 피하기 위하여 움직임 벡터의 검색 범위를 크게 9개의 영역으로 나눈 후 두 영역을 적절하게 검색하여 좋은 효율을 유지할 수 있었다.

본 연구는 한국과학재단 특정기초연구(R01-2005-000-11054-0) 지원으로 수행되었음

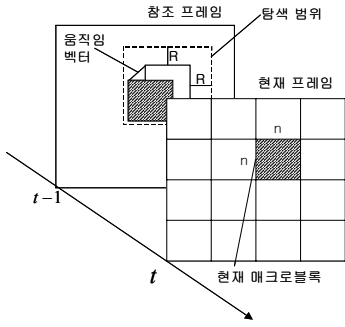


그림 1. 움직임 벡터

II. 제안한 움직임 추정 알고리즘

2.1 움직임 추정 알고리즘

일반적으로 MBA에서 유사한 블록의 기준으로 SAD (Sum of Absolute Difference) 값을 사용한다. 다음 식은 t 번째 프레임 내에서 (u, v) 에 위치한 크기 n 인 매크로블록(MB)에서의 SAD 값의 수식이다.

$$SAD^t(u, v) = \sum_{i=1}^n \sum_{j=1}^n Diff(MB^t(u, v), MB^{t-1}(u+i, j+v))$$

2.2 시간적, 공간적 특성을 이용한 예측 영역(P영역)의 선택

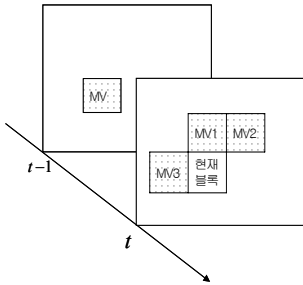


그림 2. P영역의 예측

본 논문에서는 시간적, 공간적 유사성을 이용하여 현재 탐색해야 할 MB의 MV 위치가 크게 9개의 공간 중 어느 공간인지를 다음과 같이 예측한다.

그림 2에서의 $MV^{t1}, MV^{t2}, MV^{t3}, MV^{t-1}$ 총 4개의 MV를 이용하여 MV_{mean} 을 구한다.(그림 2)

$$MV_{mean} = Mean(MV1, MV2, MV3, MV^{t-1})$$

탐색 범위를 크게 9개의 영역으로 나누어 그 중 어느 영역에 속하는가를 정하여 이를 예측한 하나의 영역(P영역, 진한부분)이라 한다.

2.3 예측 영역을 이용한 보합 영역(I영역)의 선택

위에서의 P영역만으로는 극소점들(Local Minima)에 빠질 위험이 있고, 완벽하게 예측할 수 없기 때문에 좋은 성능을 유지하기 어렵다. 본 논문에서는 다음과 같이 이를 둘러싸고있는 영역(I영역, 흐린부분)을 적절하게 탐색한다.

P영역(진한부분)은 그림 3과 같이 a),b), ... ,i) 총9가지 경우가 존재하며, 이에 따라 그림 3과 같이 I영역(흐린부분)은 정해진다.

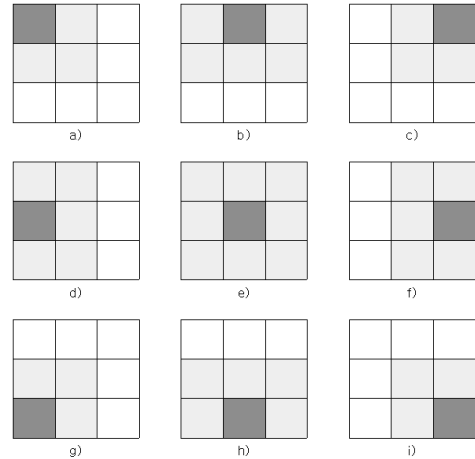


그림 3. P영역에 따르는 I영역의 관계

2.4 제안한 알고리즘

STEP1

P영역선택 : 시간적 공간적 상관관계를 이용하여 9개의 영역 중 한 영역을 선택한다.
I영역선택 : P영역이 선택됨에 따라서 1차적으로 둘러싸고있는 부분을 선택한다.
다음과 같이 P영역에서 탐색한 SAD 값중 가장작은 값을 MAD^P 라 한다.

$$MAD^P = MIN(SAD^{P1}, SAD^{P2}, \dots, SAD^{P8})$$

$$MAD^I = MIN(SAD^{I1}, SAD^{I2}, \dots, SAD^{I8})$$

$$MAD^* = MIN(MAD^P, MAD^I)$$

if $MAD^* < TH0$ then Final STEP

else STEP2

STEP2

P, I영역 사이의 차이가 충분히 많이 난다면, 다음과 같이 MAD 값이 큰 부분은 생략한다.

if $Diff(MAD^P, MAD^I) > TH1$ then

if $(MAD^* \equiv MAD^P)$ then STEP4-A

else if $(MAD^* \equiv MAD^I)$ then STEP4-B

else STEP3

STEP3

STEP 2에서 확실히 판단할 수 없으므로 I영역만 4 포인트 더 탐색한다.

$$MAD^I = \min(SAD^I, SAD^{I1}, \dots, SAD^{I4})$$

$$MAD^* = \min(MAD^P, MAD^I)$$

if $MAD^* < TH0$ then Final STEP
else STEP 4 - C

STEP 4

[STEP 4 - A] P영역만 탐색한다.

i)

$$MAD^P = \min(SAD^P, SAD^{P1}, \dots, SAD^{P4})$$

if $MAD^P < TH0$ then Final STEP

else $MAD^P = \min(SAD^P, SAD^{P5}, \dots, SAD^{P8})$

ii)

$$MAD^* = MAD^P$$

Final STEP 수행

[STEP 4 - B] I영역만 탐색한다.

i)

$$MAD^I = \min(SAD^I, SAD^{I1}, \dots, SAD^{I4})$$

if $MAD^I < TH0$ then Final STEP

else $MAD^I = \min(SAD^I, SAD^{I5}, \dots, SAD^{I8})$

ii)

$$MAD^* = MAD^I$$

Final STEP 수행

[STEP 4 - C] P, I영역 모두 탐색한다.

STEP 4 - A i)수행

STEP 4 - B i)수행

$$MAD^* = \min(MAD^P, MAD^I)$$

Final STEP 수행

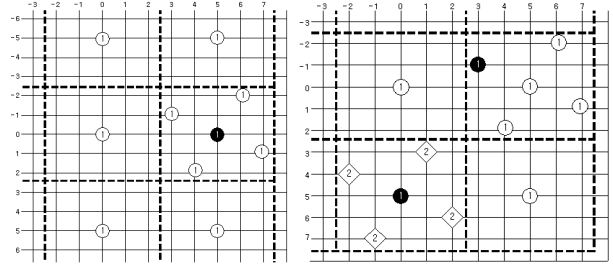
Final STEP

[Final STEP]

$$MV = \arg(MAD^*)$$

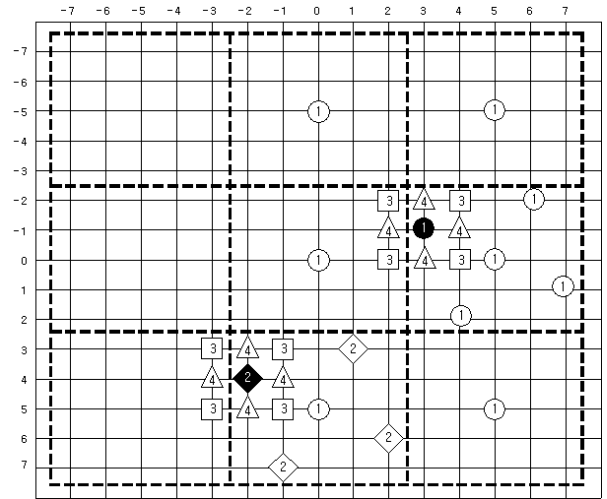
MV출력

Football의 4번째 프레임에서 (17,8)위치의 MB에서
는 $MV^4_1 = (4, 4)$, $MV^4_2 = (7, 4)$, $MV^4_3 = (6, 0)$,
 $MV^3_{(17,8)} = (-1, -7)$ 이므로 $MV_{mean} = (4, 0.25)$ 이다.
따라서 P, I영역은 그림 2-f)와 같은 형태로 선택이 되
어, 그림4-a), b), c)와 같은 순서로 탐색한다.



a) STEP 1

b) STEP 2



c) STEP 3, 4

①STEP 1 ◇STEP 2 □STEP 3 △STEP 4

그림 4. 제안된 알고리즘의 수행 예

III. 실험 결과

본 논문에서는 다양한 테스트 영상을 사용하여 기존의 예측 방법들과 제안한 예측 방법의 성능을 테스트 하였다. 각 방법을 통하여 부호화된 영상의 MSE와 움직임 벡터 탐색 과정에서의 탐색한 포인트 수를 비교 하였다. CIF(352×288) 영상에서 60 Frame 단위로 실험하였으며, 탐색 범위는 ±7로 수행하였다. 제안된 알고리즘과 성능평가를 위하여 전역탐색방법(FS)과 3SS, NDS, CSA와 비교하였다. 제안한 알고리즘의 MSE값은 다른 알고리즘에 비하여 최대 104.23 감소 했으며 움직임 벡터 탐색 포인트 수는 최대 5.79 포인트 감소하였다.(표 1)

IV. 결론

본 논문에서는 움직임 벡터의 시간적, 공간적인 상관관계를 이용한 예측 움직임 벡터를 이용한 제 1 후보영역과 극소점들(Local Minima)을 피하기 위한 제 2의 후보영역을 적절하게 탐색하는 알고리즘을 제안하

Sequence	FS	3SS	NDS	CSA	Proposed
Foreman	153.4	175.31	213.5	205.7	167.67
Football	311.39	337.1	369.2	378.12	330.58
Bus	161.13	265.08	336.63	553.95	233.74
Crew	56.16	61.03	64.5	71.96	62.48
평균	170.52	209.63	245.96	302.43	198.2

a) 블록 당 평균 MSE

Sequence	FS	3SS	NDS	CSA	Proposed
Foreman	225	25	22.6	14.66	17.16
Football	225	25	16.89	15.31	18.85
Bus	225	25	20.31	16.78	22.46
Mobile	225	25	20.61	16.12	18.37
평균	225	25	20.1	15.72	19.21

b) 블록 당 평균 탐색 포인트 수

표 1. 움직임 예측 알고리즘에 따른 성능 비교

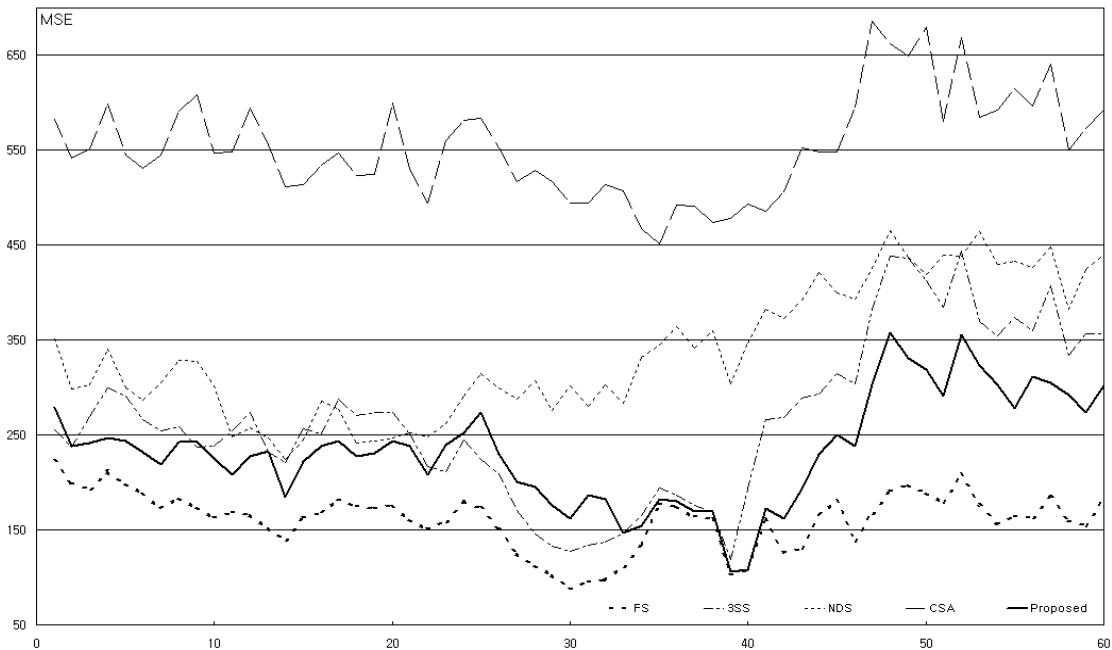


그림 5. Bus 에서의 MSE 분포

였다. 제안된 알고리즘은 실험을 통하여 기존의 여러 고속 움직임 탐색 알고리즘과 비교하였을 때 탐색 포인트 수는 최대 5.79포인트 감소하였고, MSE도 최대 104.23 감소하면서 성능 역시 향상한 것을 확인하였다.

감사의 글

저자들은 본 연구를 위하여 설계 환경을 제공하여 준 IDEC(IC Design Education Center)에 감사드립니다.

참고문헌

[1] ISO/IEC 11172-2."Information Technology-Coding of Moving Picture and Associated Audio for digital Storage Media at Up to about 1.5 Mbits/sec, Part 2: Video," Aug. 1993.
 [2] ISO/IEC 13818-2, " Information Technology- Generic Coding of Moving Picture and Associated Audio, Part 2: Video," Nov. 1995.

[3] ITU Telecom. Standardization Sector of ITU, "Video coding for audiovisual services at p x 64 kbits/s," ITU-T Recommendation H.261, 1990.
 [4] ITU Telecom. Standardization Sector of ITU, "Video coding for low bitrate communication," ITU-T Recommendation H.263, March 1996.
 [5] L. M. Po and W. C. Ma, "A new three-step search algorithm for block motion estimation," IEEE trans. Circuits Syst. Video Tehnol., vol.4, pp.438-443, Aug. 1994.
 [6] M. Ghanbari, "The Cross-Search Algorithm for Motion Estimation," IEEE trans. Commun., vol.38 NO, 7, pp.950-953, July 1990.
 [7] Shan Zhu and Kai-Kuang Ma, "A New Diamond Search Algorithm for Fast Block Matching Motion Estimation" IEEE trans. on image processing, vol.9, NO.2 February 2000.