

적응적인 최적 임계값 선택을 통한 고속 움직임 예측 방법

김근용, 김승환, 호요성
광주과학기술원 정보통신공학과
전화 : 062-970-2258 / 핸드폰 : 016-9273-3928

Adaptive Threshold Technique for Fast Block Motion Estimation

Geun-Yong Kim, Seung-Hwan Kim, Yo-Sung Ho
Gwangju Institute of Science and Technology (GIST)
E-mail : {gykim, kshkim, hoyo}@gist.ac.kr

Abstract

Motion estimation is an essential part of video coding. Although many motion estimation algorithms have been proposed, they do not consider various characteristics of the local motion in video sequences, which causes performance degradation. In this paper, we propose a new motion estimation algorithm using an adaptive threshold technique to consider the local motion. Experimental results show that the proposed algorithm provides improved coding efficiency than the previous ones.

I. 서론

최근 디지털 TV를 이용한 다양한 멀티미디어 서비스의 수요가 날로 증가하고 있으며, 이에 부응하여 영상 정보를 효율적으로 전송하기 위한 움직임 예측 방법이 많이 사용된다. 현재까지 제안된 움직임 예측 방식들 중에서 전역탐색 블록정합 방법이 많이 사용되지만, 이 방법은 탐색 영역내의 모든 경우를 계산하므로 매우 복잡하며 시간이 많이 소요된다. 이러한 복잡성을 줄이기 위해 이차원 로그 탐색 알고리즘, 삼단계 탐색 알고리즘, 교차 탐색 알고리즘 등이 제안되었지만, 이들 방식들은 정해진 탐색 형식으로 인해 최적화된 움직임 벡터를 찾지 못하는 경우가 종종 있다.

본 논문에서는 MVFAST (motion vector field adaptive search technique) [1] 알고리즘을 이용하여 이웃한 매크로 블록의 움직임 벡터를 바탕으로 현재 매크로 블록의 움직임 벡터를 예측하는 새로운 블록정합 방법을 제안한다. 제안된 방식은 이웃한 매크로 블록의 움직임

벡터와 SAD (sum of absolute difference) 값을 바탕으로 현재 매크로 블록의 움직임 벡터를 예측하며, 적응적인 최적 임계값 선택 기법을 사용한다. 제안한 알고리즘은 기존에 제안한 알고리즘에 비해 계산 속도가 빠르게 향상된 PSNR (peak signal-to-noise ratio) 결과를 보인다.

II. MVFAST 알고리즘

2.1 MVFAST 알고리즘의 기본 구조

MVFAST 알고리즘은 이웃 매크로 블록의 움직임 벡터와 SAD 값을 이용하여 현재 매크로 블록의 움직임 벡터를 예측하여 속도를 증가시킨 탐색 알고리즘이다. 일반적인 SAD 값은 다음과 같이 계산된다.

$$SAD = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} |I_t(i, j) - I_{t-1}(i, j)| \quad (1)$$

여기서 N, M 은 매크로 블록의 크기를 나타내며, I_t 는 현재 프레임의 매크로 블록, I_{t-1} 는 이전 프레임에 속한 매크로 블록을 나타낸다. 먼저 (0, 0) 움직임 벡터와 그림 1과 같이 이웃한 매크로 블록(좌, 상, 우상)의 움직임 벡터들로부터 현재 매크로 블록의 움직임 벡터를 위한 후보를 정한다. 이렇게 예측된 움직임 벡터를 중심으로 개선된 다이아몬드 탐색을 수행한다.

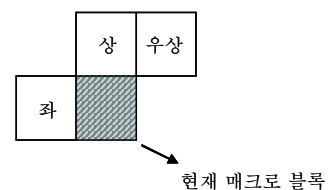


그림 1. 이웃한 매크로 블록들

그림 2(a)에서 보인 바와 같이 MVFAST 알고리즘은

기존 다이아몬드 탐색법 [2]에서 사용하는 큰 다이아몬드 형태 외에, 그림 2(e)와 같은 작은 다이아몬드 형태도 사용한다. 다이아몬드 탐색은 최소 SAD 값이 존재하는 방향으로 진행되며 그림 2(c), 그림 2(d), 그림 2(f)와 같이 최소 SAD 값이 가운데에 위치하면 그 점을 최소점이라 판단하여 탐색을 종료한다 [1]. 이때 작은 다이아몬드 탐색 형태가 사용되었다면 탐색이 종료된다. 만약 큰 다이아몬드 탐색 형태가 사용되었다면 최종 탐색은 작은 다이아몬드를 사용하여 그림 2(b)와 같이 수행된다.

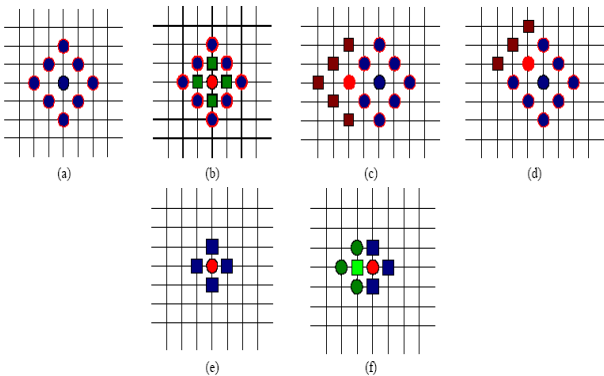


그림 2. MVFAST 움직임 탐색 형태

2.2 탐색 시작점의 설정

MVFAST 알고리즘에서 어떤 다이아몬드 형태를 사용할 것인가의 선택은 현재 매크로 블록의 움직임 벡터가 가지게 될 움직임 정도에 의해 결정된다. 식 (2)는 현재 매크로 블록이 가지게 될 움직임 정도를 나타낸다 [3].

$$\begin{cases} Low, & \text{if } L < L1 \\ Medium, & \text{if } L1 \leq L \leq L2 \\ High, & \text{if } L > L2 \end{cases} \quad (2)$$

식 (2)에서 L 값은 그림 1과 같이 이웃한 세 매크로 블록의 움직임 벡터 중 크기가 가장 큰 것의 값을 갖는다. L 값이 L1 값보다 작다면 현재 매크로 블록의 움직임 벡터는 작을 것이라고 예측되고, (0, 0)을 시작점으로 작은 다이아몬드 탐색을 시작한다. L 값이 L2 값보다 크면 움직임이 클 것이라고 예측되고 (0, 0)과 이웃한 매크로 블록의 움직임 벡터를 후보로 SAD 값들을 비교한다. 이 중 최소 SAD 값을 가지는 움직임 벡터를 시작점으로 작은 다이아몬드 탐색을 시작한다. 그 외의 경우는 (0, 0)을 시작점으로 큰 다이아몬드 탐색을 수행한다. 만약 (0, 0)의 SAD 값이 임의의 고정된 임계값보다 작으면, 움직임 벡터를 (0, 0)으로 설정하고 탐색을 종료한다. 일반적으로 임계값은 512가 사용된다 [1].

III. 제안한 최적 임계값 선택 방법

3.1 MVFAST 알고리즘의 문제점

MVFAST 알고리즘은 움직임 벡터 (0, 0)의 SAD 값이 고정된 임계값 보다 작을 경우, 다른 탐색없이 (0, 0)을 움직임 벡터로 정하고 탐색을 종료하게 된다. 그러나 고정 임계값 512는 영상의 국부적인 특성을 반영하지 못하기 때문에, PSNR 향상과 탐색의 조기 종료를 위해서는 영상에 적응적인 임계값 설정이 필요하다. 또한 초기에 예측되어 비교되는 (0, 0) 움직임 벡터는 움직임이 많은 영상의 경우, 잘못된 시작점을 선택하게 되어 수행 속도를 저하시키는 요인이 될 수도 있다.

3.2 효과적인 움직임 벡터 예측과 임계값 선택

본 논문에서는 이런 문제점들을 해결하기 위해 다음과 같은 방법을 사용하였다.

첫째, 시작점 예측의 정확도를 높이기 위해 또다른 예측 움직임 벡터를 선택한다. 이웃 매크로 블록의 SAD 값 중 최소값을 (0, 0)의 SAD 값과 비교한다. 이 중 작은 SAD 값을 가진 움직임 벡터를 시작점으로 선택한다.

둘째, 적응적으로 임계값을 선택한다. 이를 위해 식 (3)과 같이 이웃 매크로 블록 간 SAD 값의 변화율인 S_r 값을 정의한다.

$$S_r = MAX(S1, S2, S3) - MIN(S1, S2, S3) \quad (3)$$

식 (3)에서 S1, S2, S3는 각각 이웃 매크로 블록들의 SAD 값이다. S_r 값을 바탕으로 식 (4)와 같이 탐색 초기 종료를 위한 적응적인 임계값을 선택한다.

$$\begin{cases} Th = LowThreshold, & \text{if } S_r < Th_L \\ Th = HighThreshold, & \text{if } S_r > Th_H \\ Th = MediumThreshold, & \text{others} \end{cases} \quad (4)$$

전역 움직임은 영상내에서 카메라의 움직임과 같은 전체영상의 움직임을 나타낸다. 하지만 전역 움직임으로는 한 영상의 움직임을 모델하는데는 한계가 있다. 즉, 한 영상내에 여러개의 객체가 있는 경우는 각 객체가 지나는 국부적인 움직임이 존재하며, 같은 객체 안에서도 부분적으로 서로 다른 국부적인 움직임이 존재한다.

본 논문에서는 이러한 국부적인 움직임을 매크로 블록 단위까지 고려하기 위해 이웃 매크로 블록들의 SAD 값의 연관성을 이용한다. 실험을 통해 국부적인 움직임을 가지고 있는 경우에 이웃 매크로 블록들의 움직임 벡터가 비슷하고 SAD 값의 변화율이 낮을 확률이 높다는 것을 확인하였다. 따라서 영상내 객체의 국부적인 움직임이 일정한 경우가 많으면 탐색의 조기 종료를 위해 기존에 사용하던 512 값 대신 큰 임계값을 사용한다.

셋째, 탐색 속도 증가를 위해 이웃한 매크로 블록의 움직임 벡터가 동일하고 $S_r < Th_L$ 인 경우는 탐색을

조기에 종료시킨다. 이와 같은 경우는 현재 매크로 블록도 이웃한 매크로 블록들과 동일한 움직임 벡터를 가지고 있다고 판단한다. 이는 매크로 블록들 사이의 SAD 값과 움직임 벡터의 연관성에 기반한다 [4].

3.3 적응적인 임계값 선택 움직임 예측 방법

그림 3은 제안한 움직임 예측 알고리즘의 순서도이다.

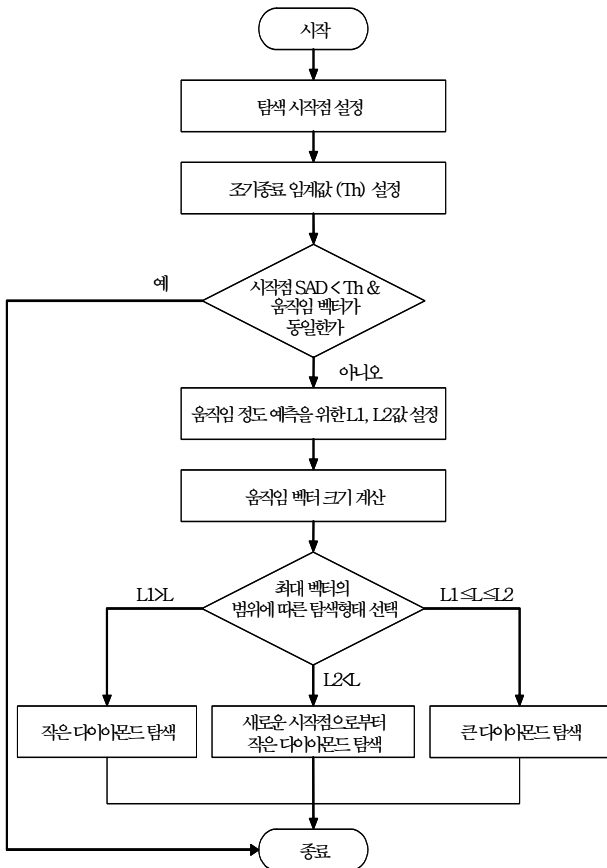


그림 3. 적응적인 임계값 선택 움직임 예측 알고리즘

위 순서도에서 보는 바와 같이 처음에 (0, 0)의 SAD 값을 구하여 이웃한 매크로 블록들의 SAD 값과 비교한다. 이 중 최소 SAD 값을 가지는 움직임 벡터를 탐색 시작점으로 설정한다. 다음은 이웃 매크로 블록들 사이의 SAD 값의 변화율을 바탕으로 조기종료를 위해 임계값 Th를 설정한다. 임계값은 식 (4)와 같이 현재 영상에 얼마나 많은 국부적인 동일 움직임이 있느냐에 따라 각각 다른 값을 가지게 된다. 이를 통하여 설정된 임계값과 시작점 SAD 값을 비교하여 탐색을 조기에 중단할 것인가를 결정하게 된다. 만약 시작점 SAD 값이 설정된 임계값 보다 작을 경우는 시작점이 최종 움직임 벡터로 결정되고 탐색이 종료된다. 그렇지 않을 경우는 다이아몬드 탐색을 시작하게 된다. 이를 위하여 우선 현재 움직임 벡터의 움직임 정도를 예측하

기 위한 L1 값과 L2 값을 설정한다. 이렇게 설정된 L1 값과 L2 값을 이웃 매크로 블록의 최대 움직임 벡터 크기 (L)와 비교하여 탐색을 수행할 다이아몬드 형태를 결정한다.

최대 움직임 벡터의 크기에 따라 $L1 > L$ 인 경우는 시작점을 중심으로 작은 다이아몬드 탐색을 수행하고 $L1 \leq L \leq L2$ 인 경우는 시작점을 중심으로 큰 다이아몬드 탐색을 수행한다. 또한 $L2 < L$ 인 경우는 이웃한 세개의 매크로 블록의 움직임 벡터들을 시작점 후보로 하여 각각에 대해 SAD 값을 계산한다. 이 중 최소 SAD 값을 가지는 움직임 벡터를 새로운 시작점으로 설정하고 작은 다이아몬드 탐색을 수행한다.

IV. 실험결과 및 분석

제안한 알고리즘의 성능을 비교하기 위하여 기존의 탐색 알고리즘 중 전역 탐색법, 삼단계 탐색법, MVFAST 탐색법에 대해서 실험을 실시하였다. 실험을 위해 CIF 포맷의 Foreman, Coastguard, Football, News, Container, Flowergarden, Mobile 등을 사용했으며, 각각 초기 100 프레임을 이용하였다. 탐색 영역은 가로와 세로 각각 -14~+15 화소의 크기를 가지도록 설정하였다. 부호화 비트율은 Foreman과 Coastguard, Football, News, Container는 384k 비트로 설정했고, Flowergarden와 Mobile은 1M 비트로 설정했다. 실험 영상은 MPEG4 simple profile 소스코드를 이용하여 부호화 되었으며 각 알고리즘의 성능을 비교하기 위하여 전체 탐색점의 개수와 PSNR 값을 사용하였다. 일반적인 PSNR 값은 다음과 같이 계산된다.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (5)$$

MSE (mean square error)는 평균 자승 오차이며 한 프레임에 대해 수행된 MSE는 다음과 같이 정의된다.

$$MSE = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} |I_i(i, j) - I_i'(i, j)|^2}{N \times M} \quad (6)$$

여기서 N, M 은 프레임의 크기를 나타내며, I_i 는 원 프레임, I_i' 는 복호화된 예측 프레임을 나타낸다. 제안한 탐색법과 기존 탐색법의 PSNR 값과 전체 탐색점의 개수를 표 1과 표 2에 각각 나타내었다.

표 1. 각 알고리즘의 PSNR 비교(dB)

구분	전역 탐색	삼단계 탐색	MVFAST	제안한 알고리즘	Th_L/Th_H
테스트영상					
Foreman	33.81	33.07	33.67	33.81	256/1024

Coastguard	28.53	28.05	28.51	28.55	1024/4096
Football	36.95	36.61	36.88	36.89	256/2048
News	37.86	38.04	37.79	37.80	1024/4096
Container	34.98	34.98	34.98	34.98	256/1024
Flowergarden	29.18	28.99	29.03	29.10	1024/4096
Mobile	26.54	26.47	26.56	26.58	1024/4096

표 2. 각 알고리즘의 전체 탐색점 개수 비교

구분 테스트영상	전역 탐색	삼단계 탐색	MVFAST	제안한 알고리즘	Th_L/ Th_H
Foreman	35283600	980100	339947	335783	256/1024
Coastguard	35283600	980100	391859	290037	1024/4096
Football	35283600	980100	395251	394484	256/2048
News	35283600	980100	128635	123921	1024/4096
Container	35283600	980100	157183	148950	256/1024
Flowergarden	35283600	980100	286592	220515	1024/4096
Mobile	35283600	980100	282377	251085	1024/4096

표 1과 표 2에 나타난 바와 같이 제안한 알고리즘은 탐색점의 개수가 적음에도 불구하고 향상된 PSNR 결과를 보여준다. 특히 주목해야 할 사항은 식 (3)에 정의된 Th_L 값과 Th_H 값의 설정이다. Th_L 값과 Th_H 값은 영상내에서 국부적인 움직임이 어느 정도 존재하는지를 결정하는 임계값이다. 실험을 통해 확인한 결과 Foreman이나 Football과 같이 영상의 움직임이 일정하지 않은 경우는 Th_L 값은 256, Th_H 값은 1024나 2048을 설정할 때 좋은 결과를 얻을 수 있었다. 또 Coastguard나 News, Flowergarden과 같이 영상에서 객체의 움직임이 많지 않은 경우나 카메라의 움직임이 일정한 경우 Th_L 값은 1024, Th_H 값은 4096으로 설정할 때 속도와 PSNR 값 양 측면에서 좋은 결과를 얻을 수 있었다.

객체의 움직임이 일정하여 국부적인 동일 움직임이 많은 경우는 이웃 매크로 블록 사이의 SAD 값의 변화율을 통해 움직임 벡터를 예측할 수 있다. 따라서 Th_L 값과 Th_H 값을 큰 값으로 설정하여 조기에 탐색을 종료시킬 확률을 높이며, 객체의 움직임이 일정하지 않아 국부적인 움직임이 많지 않은 경우는 Th_L 값과 Th_H 값을 높게 설정하여 조기 종료 확률을 낮춘다. 이를 통해 PSNR 값의 손실없이 탐색점의 개수를 줄일 수 있었다. 모든 영상에 대하여 식 (4)에서 정의한 임계값 Th는 각각 MAX_Th, 512, 256으로 설정하였다. MAX_Th는 충분히 큰 값을 의미한다. 곧 이웃한 매크로 블록의 움직임 벡터가 모두 같고 Sr 값이 Th_L 값보다 작은 경우는 조기에 탐색을 종료한다. 또한 현재

매크로 블록 움직임 벡터의 움직임 정도를 예측하는 임계값 L1과 L2는 각각 1과 2로 설정하였다 [1].

V. 결론

본 논문에서는 이웃한 매크로 블록들 사이의 상호 연관성을 이용하여 영상 압축을 위한 움직임 예측 알고리즘을 제안하였다. 제안한 알고리즘에서는 이웃한 매크로 블록들의 SAD 값에 적응적으로 탐색의 조기 종료를 위한 임계값을 결정하였다. 또한 현재 움직임 벡터를 예측하기 위해 이웃 매크로 블록들의 SAD 값의 변화율을 이용하였다. 실험을 통하여 제안한 알고리즘은 기존의 고속 탐색법과 비교하여 PSNR과 탐색 속도에 있어서 향상된 성능을 가지고 있음을 알 수 있었다.

감사의 글

본 연구는 광주과학기술원(GIST)과 광주과학기술원 실감방송연구센터를 통한 대학 IT연구센터(ITRC), 그리고 교육부 두뇌한국21 (BK21)정보기술사업단의 지원에 의한 것입니다.

참고문헌

- [1] P. I Hosur and K. K. Ma, "Motion Vector Field Adaptive Fast Motion Estimation," International Conference on Information, Communications and Signal Processing, Dec. 1999.
- [2] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation," International Conference on Information, communications and Signal Processing, vol. 1, pp. 292-296, 1997.
- [3] Weiguozheng, Ishfaq Ahmad and Ming Lei Liou, "Adaptive motion search with elastic diamond for MPEG-4 video coding," International Conference on Image Processing, vol. 1, pp. 337-380, Oct. 2001.
- [4] Zhibo Chen, Peng Zhou and Yun He, "Fast Motion Estimation for JVT," JVT-G016, March 2003.