

매크로블록 간 움직임유사도를 이용한 고속 PDE 알고리즘

유태경* · 정용재* · 문광석* · 김종남*

*부경대학교

A Fast Partial Distortion Elimination Algorithm Using Motion Similarity of Macro-blocks

Tae-Kyung Ryu* · Yong-Jae Jeong* · Kwang-Seok Moon* · Jong-Nam Kim*

*Pukyong National University

E-mail : toydev@pknu.ac.kr

요 약

본 논문에서는 인접블록간의 움직임 유사도를 이용하여 불필요한 후보블록을 보다 빠르게 제거하는 PDE기반의 고속 블록매칭 알고리즘을 제안한다. 제안한 방법은 기존의 방법보다 불필요한 계수를 효율적으로 제거하기 위하여 인접 블록간의 영상의 유사성에 기초하여 네 개의 인접 매크로블록 가운데 최대 복잡도를 가지는 서브블록의 누적 분포 함수(cumulative distribution function-CDF)와 서브블록별 복잡도가 집중되지 않도록 하기위하여 normalized스캔 방법에 사용하여 효율적으로 계산량을 감소하였다. 제안한 알고리즘은 화질의 저하 없이 기존의 PDE 알고리즘에 비해 55% 이상의 계산량을 줄였으며, MPEG-2 및 MPEG-4 AVC를 이용하는 비디오 압축 응용분야에 유용하게 사용될 수 있을 것이다.

키워드

비디오코딩, 움직임 예측, 부분 에러 제거, 누적 분포 함수,

1. 서론

움직임 추정(motion estimation)은 동영상 압축 표준에서 비디오 시퀀스간의 공간적 중복성을 활용하는 중요한 역할을 한다. 대부분의 움직임 추정 알고리즘에서 움직임벡터를 계산하기 위한 방법으로 블록 매칭 알고리즘 가운데 가장 확실한 방법은 주어진 탐색영역안의 가능한 모든 위치를 검색하는 전 영역 탐색 방법이다. 이 방법은 최적의 성능을 보이지만 매우 많은 계산량을 가지는 단점이 있다. 이러한 특징이 실시간 비디오 시스템에서 전 영역 탐색알고리즘을 사용하지 않는 이유가 된다. 여기에 추가해서 움직임 추정은 실시간 비디오 코딩에서 중요한 장애 요인으로 작용한다. 이러한 문제점을 해결하기 위해 지난 십수 년간 고속 알고리즘에 대한 연구가 진행되어져 왔는데, 이들 알고리즘들은 크게 두 그룹으로 나누어 질 수 있다. 하나는 전 영역 탐색 방식에 비해 예측 화질의 손실을 갖는 것이고, 다른 하나는 예측 화질의 손실을 갖지 않는 방식이다. 전자는 다음의 세부 그룹으로 다시 나누어 질 수 있다.

다이아몬드 서치(diamond search-DS), 육각형 서치(hexagon based search-HEXBS)등이 있다[1-2]. 무손실 움직임 예측 기법에는 기존 블록과 후보 블록의 블록합을 이용한 후보 제거 알고리즘(successive elimination algorithm-SEA) 및 변형된 알고리즘들[3], 그리고 부분 매칭 에러값을 이용한 후보 제거(partial distortion elimination-PDE) 알고리즘 및 그 변형 알고리즘 등이 있다[4-7].

본 논문에서는 기존의 무손실 기반 움직임 추정방법인 PDE 알고리즘의 계산량을 보다 많이 줄이기 위하여 PDE기반으로 인접한 매크로블록과의 유사도를 고려하여 서브블록의 누적비율(cumulative distribution function- CDF)[7]을 고려하여 최소 SAD값에 적용하고 서브블록별로 보다 균일한 오차값을 적용하기 위하여 normalized 스캔방법의 초기 오차 보정을 통하여 보다 빠르게 후보블록을 제거하는 고속 PDE알고리즘을 제안한다. 제안한 방법을 고속 움직임 추정 방법에 적용하여 보다 빠른 움직임 추정이 가능하다.

II. 기존의 고속 전역 탐색 알고리즘

1. PDE(partial distortion elimination)

PDE알고리즘은 후보 블록의 부분 매칭 에러를 가지고 최소 에러와 비교함으로써 불필요한 계산을 줄이는 것이다. 즉, 매칭 에러의 중간 합이 그때까지의 최소 에러보다 크다면 나머지 계산을 할 필요가 없다는 것이다. 식 (1)은 일반적인 PDE에서 사용되는 SAD값의 행별 부분비교를 나타낸 것이다.

$$\sum_{i=1}^k \sum_{j=1}^N |f_t(i, j) - f_{t-1}(i+x, j+y)| \quad k=1, 2, \dots, N \quad (1)$$

식 (1)에서 N 은 블록 크기를 의미한다. 앞에서 언급했듯이, $k < N$ 일 때 부분 매칭 에러 합이 그때까지의 최소 에러보다 크다면 $k+1$ 부터 N 까지의 행에 대한 매칭 에러 계산을 절약 할 수 있다. PDE 방법은 전 영역 탐색 방법뿐만 아니라 다른 고속 탐색 방법에서도 계산을 효율적으로 줄이기 위해 사용되어질 수 있다. PDE 알고리즘의 계산량 감소는 주어진 탐색영역에서 전체 최소 에러를 가능한 빨리 찾는 것과 해당 매칭 블록에서 매칭 에러가 큰 영역을 먼저 계산함으로써 얻을 수 있다. 기존의 변형된 PDE 알고리즘에서 제시된 방법들은 후보 블록을 보다 효율적으로 빨리 제거하기 위하여 서브블록을 여러 가지 형태로 변형하거나 매칭 순서를 변형하여 보다 빨리 최소 SAD 값보다 큰 값이 나오도록 하였다. 서브블록의 형태는 MB을 크게 가로방향 또는 세로방향 그리고 4×4 형태의 정방 서브블록으로 나누고 서브블록의 각각의 픽셀들을 모아 부분계수로 만드는 방법 등이 있다.

2. NPDS(normalized partial distortion search)

무손실 움직임 추정 방법의 경우 손실 방법에 비해 계산량의 감소가 적은 단점을 가지고 있다. 이런 계산량 감소의 단점을 해결하기 위하여 예측화질의 손실을 일부 감수하더라도 계산량 감소를 이루려는 연구들이 진행되었다. 이 방법들 가운데 대표적인 것이 NPDS(normalized partial distortion search)방법이다. 이 방법은 기존의 PDE방법이 부분 SAD값과 최소 SAD값을 비교하여 불가능한 후보벡터를 제거하였다면 NPDS방법은 보다 빠른 후보블록 제거를 위해 부분SAD값이 서브블록별로 균일하게 분포하여 하나의 서브블록에 몰리지 않도록 normalized 매칭 스캔 방법을 사용하여 부분계수를 정렬한다.[6] 서브블록별 SAD값(d_p)과 부분 SAD값은 수식 (2)와 (3)과 같이 구할 수 있다. 고속 움직임 제거 알고리즘은 식(4)와 같이 부분 SAD값과 SAD_{\min} 값의 부분 값으로 연속 비교하여 불필요한 후보 벡터를 제거

한다.

$$d_p = \sum_{k=0}^3 \sum_{l=0}^3 \left| f_t(i+4k+s_p, j+4l+t_p) - f_{t-1}(i+4k+s_p+x, j+4l+t_p+y) \right| \quad (2)$$

$$SAD_p = \sum_{i=1}^p d_i \quad (3)$$

$$16SAD_p \geq p SAD_{\min} \quad (4)$$

하지만 이 방법의 경우 식(4)를 적용하여 후보 벡터를 제거할 경우 d_p 의 값이 SAD값의 1/16로 균일하게 분포하지 않고 SAD_p 가 초기에 큰 값을 갖는 경우 최소 움직임 벡터가 제거될 수 있는 경우가 발생하는 단점이 있다.

III. 제안한 알고리즘

비디오 영상에서 이전 프레임과 현재 프레임간의 상관관계를 실험적으로 고려해 보면 프레임간의 움직임은 매우 작다. 정적인 영상의 경우는 거의 움직임이 없는 경우도 있다. 이러한 비디오영상의 특성을 제대로 고려하기 위하여 매칭 스캔 순서를 정하는 방법으로 나선형 스캔(spiral scan)을 적용하여 최소 SAD값이 우선적으로 나올 수 있도록 고려하였다. 기존의 PDE방법의 계산량 감소를 위해서는 NPDS 방법에서 사용하였던 서브블록과 SAD_{\min} 값의 서브블록별 부분 값을 비교하여 계산량 감소를 이룰 수 있었다. 하지만 NPDS방법의 경우 서브블록의 오차값이 SAD_{\min} 값의 부분 값 보다 커지는 상황이 발생하여 예측화질의 손실이 발생한다. 이러한 예측화질의 손실을 막고 계산량 감소를 이루기 위해서는 새로운 매칭 스캔 방법이 필요하다.

제안한 방법은 단일 프레임 내에서 인접한 매크로블록간에는 움직임의 유사성이 존재한다는 것을 고려하였다. 따라서 그림 1과 같이 하나의 프레임 영상에서 인접한 4개의 매크로블록간의 유사성을 고려하여 참조 매크로블록을 위치에 따라 4가지 종류의 참조 블록형태(UpperLeft, UpperRight, Upper, Left)로 구성하고 각 참조블록의 복잡도를 참조하여 현재 후보블록을 적응적으로 제거한다. 우선 참조블록으로 사용되는 매크로블록의 움직임 추정의 경우 기존의 복잡도 기반 움직임 추정방법으로 최소SAD값을 가지는 후보점의 서브블록별 복잡도를 기준으로 서브블록 매칭순서를 구하여 무손실 움직임 추정방법을 수행한다. 일반적으로 네 개의 후보블록가운데 3개 이상의 인접한 참조블록이 없을 경우 참조블록으

로 가정하고 사용한다. 참조 매크로블록을 가지고 유사성을 고려하는 경우 식 (5)와 같이 네 개의 참조 매크로블록의 CDF값 가운데 서브블록별 최대값을 구하여 후보블록별로 SAD_{min} 값과 CDF값의 곱과의 비교를 통해 보다 빠르게 후보블록을 제거할 수 있다.

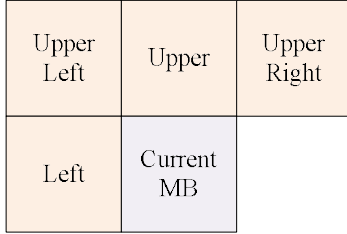


그림 1. 참조 블록과 CDF를 적용할 매크로 블록

$$\max CDF[p] = \max(CDF_{upper_left}[p], CDF_{upper}[p], CDF_{upper_right}[p], CDF_{left}[p]) \quad (5)$$

서브블록의 크기는 4×4 크기를 갖는 형태로 서브블록에서 각각 하나의 픽셀을 가져오는 normalized 방법을 개선한 방법을 사용하였다. 기존의 서브블록 매칭방법인 1×16, 16×1, 4×4의 크기를 갖는 서브블록 매칭은 서브블록의 복잡도가 하나의 서브블록에 집중되는 경우가 발생한다. 이러한 복잡도의 집중은 NPDS 기반 방법에서의 예측화질 손실의 원인이 된다. 제안한 방법은 이러한 서브블록별 복잡도를 각 서브블록별로 고르게 분포하도록 하기위하여 normalized 스캔방법을 적용하여 서브블록 매칭시 CDF값이 하나의 서브블록에 모여 원하지 않는 후보블록이 제거되지 않도록 normalized 방법에 디터링 방법을 적용하고 최초 서브블록의 복잡도를 기반으로 서브블록의 매칭순서를 정하였다. 초기 매칭 블록의 오차를 고려하여 제안한 방법은 초기 매칭 시 발생하는 오차를 보정할 수 있는 계수 값($\alpha=1/32$)을 더하였다. 제안한 방법은 식(6)과 같이 구할 수 있다.

$$pSAD < (\max CDF[p] + \alpha) SAD_{min} \quad (6)$$

$$p = 1, 2, \dots, 16$$

IV. 실험 및 결과

제안한 알고리즘과 기존의 알고리즘의 성능을 비교하기 위해, "foreman", "car phone", "trevor", "claire", "grandmother"의 비디오 시퀀스를 실험하였다. 이들 시퀀스에서, "foreman" 과 "car phone"은 다른 테스트 시퀀스에 비해 비교적 큰

움직임을 가지고 있다. 한편 "claire", "grandmother"는 이전 두 테스트 시퀀스보다 비교적 작은 움직임을 갖는다. "trevor" 시퀀스는 이들 두 부류의 중간 정도 움직임을 갖는다고 할 수 있다. 매칭 블록의 크기는 16×16이며, 탐색 영역의 범위는 ± 7 화소이다. 프레임의 크기는 QCIF(176×144)이다. 실험 결과는 계산된 평균 행의 수와 PSNR로 나타내었다. 계산된 평균 행의 수는 하나의 스캔 블록에서 계산 되는 평균 행의 수치이다.

표 1-2는 30fps, 10fps에서의 모든 영상 시퀀스의 서브 블록에 대해 전역탐색방법, 순차적인 PDE 방법과 복잡도 기반의 PDE방법, Hadamard 방법과 제안한 방법을 적용하여 계산된 평균 행의 수를 보여 준다. 모든 수치는 복잡도를 측정하기 위해 모든 부가적인 계산을 고려한 값이다. 고속 알고리즘을 전혀 사용하지 않은 원래의 전 영역 탐색 알고리즘은 표에서 보는 바와 같이 평균 계산 행의 수가 매칭 블록의 크기인 16이 된다. 표 1에서 30fps의 경우 제안한 알고리즘은 기존의 PDE알고리즘보다 50~63%의 계산량 감소를 보였다. 표 2의 경우 제안한 알고리즘을 10fps의 영상에 적용하면 53~65%의 계산량 감소를 나타낼 수 있었다.

그림2는 10fps "foreman"영상에서 각 방법별 영상 시퀀스에서 서브 블록에 대해 다양한 방법으로 실험된 순차적 매칭 스캔과 제안한 방법을 사용하여 계산된 평균 행의 수를 보여 준다. 제안한 방법을 적용하면 각 후보 블록별로 종래의 순차적인 방법에 비해 매칭 에러를 계산하는데 있어 서브 블록 당 매칭에러 비교를 적용했을 때 불필요한 계산을 현저히 줄였다.

표 3은 30fps, 10fps에서 기존의 방법과 제안한 방법에서의 PSNR 차이이다. 이 차이는 제안한 방법을 적용하였을 경우 발생하는 오차의 정도이다. 제안한 방법은 전역탐색 방법과 비교하여 영상의 품질의 차이가 거의 없었다. 따라서 본 논문에서는 계산의 복잡도를 제거하기위하여 참조 서브블록과 현재 서브블록간의 SAD_{min} 값과의 CDF를 이용하여 블록별로 적응적 임계치를 구하여 화질의 감소 없이 우수한 계산량의 감소를 얻을 수 있었다.

표 1. 30fps 영상 시퀀스에서 서브 블록에 대해 다양한 매칭 스캔의 계산된 평균 행의 수.

| Seqs. Algs. | Foreman | Car phone | Trevor | Claire | Grand ma |
|-----------------|-------------|-----------|--------|--------|----------|
| | Original FS | 16.00 | 16.00 | 16.00 | 16.00 |
| PDE(sequential) | 4.03 | 4.39 | 3.13 | 3.88 | 4.11 |
| PDE(complexity) | 2.87 | 3.70 | 2.40 | 3.20 | 3.44 |
| PDE(hadamard) | 2.77 | 3.66 | 2.38 | 3.20 | 3.45 |
| proposed | 1.58 | 1.74 | 1.43 | 1.09 | 1.93 |

표 2. 10fps 영상 시퀀스에서 서브 블록에 대해 다양한 매칭 스캔의 계산된 평균 행의 수.

| Algs. | Seqs. | | | | |
|-----------------|---------|-----------|--------|--------|----------|
| | Foreman | Car phone | Trevor | Claire | Grand ma |
| Original FS | 16.00 | 16.00 | 16.00 | 16.00 | 16.00 |
| PDE(sequential) | 5.05 | 7.30 | 4.51 | 4.40 | 4.61 |
| PDE(complexity) | 3.89 | 6.55 | 3.59 | 3.49 | 3.79 |
| PDE(hadamard) | 3.71 | 6.47 | 3.50 | 3.49 | 3.79 |
| proposed | 2.16 | 2.03 | 1.79 | 1.49 | 1.98 |

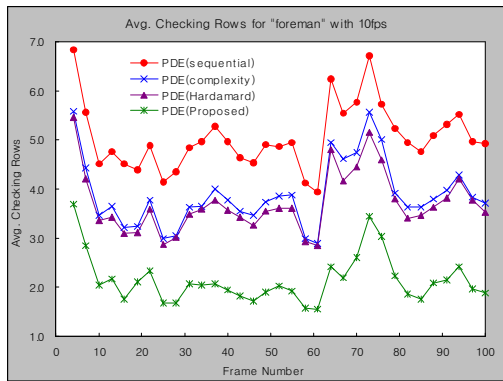


그림 2. 10fps "foreman" 영상 시퀀스에서 기존의 순차적인 스캔방법과 제안한 방법의 계산된 평균 행의 수.

표 3. 30fps, 10fps 영상 시퀀스에서 제안한 방법과 FS의 PSNR차

| Algs. | Seqs. | | | | |
|-------|---------|-----------|--------|--------|----------|
| | Foreman | Car phone | Trevor | Claire | Grand ma |
| 30fps | 0.001 | 0.002 | 0.022 | 0.002 | 0.001 |
| 10fps | 0.002 | 0.001 | 0.014 | 0.021 | 0.010 |

V. 결 론

본 논문에서는 기존의 PDE기반의 블록매칭 알고리즘의 계산량을 줄이고 동시에 동일한 예측화질을 얻기 위해, 단일프레임 내에서 매크로 블록간의 움직임 유사도를 이용한 개선된 서브블록 매칭스캔 알고리즘을 제안하였다. 기존의 PDE 방법을 개선하기 위하여 인접 매크로블록의 각 서브블록별 CDF를 고려하여 최소 SAD값에 적용하여 후보블록을 서브 블록 단위로 효율적으로 제거하고 예측화질의 손실을 줄이기 위하여 서브블록당 복잡도를 줄이는 normalized 스캔 방법을 사용하여 전역 탐색방법과 거의 유사한 PSNR을

유지하면서 효율적으로 불필요한 계산량을 줄였다. 제안한 알고리즘은 화질의 저하 없이 기존의 PDE 알고리즘에 비해 55% 이상의 계산량을 줄였으며, MPEG-2 및 MPEG-4 AVC를 이용하는 비디오 압축 응용분야에 유용하게 사용될 수 있을 것이다.

감사의 글

본 연구는 중소기업청이 주관하는 산학협력실 지원사업 과 중소기업청 산학연공동기술개발사업 (선도형) 사업의 지원으로 수행되었음.

참 고 문 헌

- [1] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Processing*, vol. 4, pp. 105-107, Jan. 1995.
- [2] X.Q. Gao, C.J. Duanmu, and C.R. Zou, "A multilevel successive elimination algorithm for block matching motion estimation," *IEEE Trans. Image Processing*, vol. 9, pp. 501-504, Mar. 2000.
- [3] T.G. Ahn, Y. H. Moon and J.H Kim, "Fast fullsearch motion estimation based on Multilevel Successive Elimination Algorithm," *IEEE Trans. Circuits System for Video Technology*, vol. 14, pp. 1265-1269, Nov. 2004.
- [4] J.N. Kim, and et al., "A fast full-search motion-estimation algorithm using representative pixels and adaptive matching scan," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 10, pp. 1040-1048, Oct. 2000.
- [5] J.N. Kim, S.C. Byun, Y.H. Kim and B.H. Ahn, "Fast Full Search Motion Estimation Algorithm Using Early Detection of Impossible Candidate Vectors", *IEEE Trans. Signal Processing*, vol. 50, pp. 2355-2365, Sept. 2002.
- [6] C.K. Cheung and L. M. Po, "Normalized partial distortion search algorithm for block motion estimation," *IEEE Trans. Circuit System for Video Technol.*, vol. 10, No. 3, pp. 417-422, Apr. 2000.
- [7] S. Jin, H. Lee and J. Jeong, "Fast Partial Distortion Elimination Algorithm for Lossless and Lossy Motion Estimation Using Hadamard Transform and Probability Model," *Data Compression Conference*, 2008, pp. 523 - 523, Mar. 2008.