

기울기 기반 적응 샘플링을 통한 블록 움직임 추정

박보운, 배황식, 김종욱, 정정화
한양대학교 전자공학과
e-mail : pypark@hanmir.com

Motion Estimation Using Gradient-based Adaptive Sampling Block Matching

Po-Yun Park, Hwang-Sik Bae, Jong-Woog Kim, Jong-Wha Chong
Dept. of Electrical Engineering, Hanyang University

요 약

본 논문에서 제안한 알고리즘은 동영상의 한 프레임내에서 계산할 화소를 적응적으로 샘플링하여 전체 계산량을 감소시키면서도, 우수한 성능을 나타낸다. 동영상의 움직임 추정에 있어서, 전역 탐색 블록 정합은 최적의 성능을 나타내지만, 많은 계산량을 갖는 단점이 있기 때문에 계산량을 줄이기 위한 많은 알고리즘들이 발표되었다. 본 알고리즘은 한 프레임내에서 영상이 복잡한 부분에서는 많은 수의 화소를 정합에 이용하며, 단순한 부분에서는 적은 수의 화소를 샘플링 하여 보다 적은 계산량으로 움직임 벡터를 산출한다. 제안된 알고리즘은 전역 탐색 블록 정합 알고리즘에 대해서 연산량을 31.52%로 줄였으며, 97.76%의 PSNR 을 보여주었다. 이는 종래의 부분 탐색 알고리즘들에 비해 뒤지지 않는 성능을 나타낸다.

1. 서론

동영상 데이터의 시간적인 중복성을 줄이기 위해서 사용되는 움직임 추정 (motion estimation)은 움직임 벡터(motion vector)를 만들어내어 프레임간의 차를 부호화하게 해준다.

움직임 벡터를 만들어내기 위한 움직임 추정에는 여러 방법이 있는데, 그 중 블록 정합 방식(block matching algorithm)은 과정이 간단하고 예측 효율과 추정의 정확도 등을 고려했을 때 전체적으로 좋은 특성을 보이며, 하드웨어로 구현하는데도 용이하기 때문에 많은 연구가 이루어지고 있다. 블록 정합 방식 중 전역 탐색 방식(full search)의 경우 탐색 영역 안에서 모든 화소와의 비교 계산을 하기 때문에, 정합 에러가 가장 적은 움직임 벡터를 찾을 수 있으나 많은 계산량이 필요한 단점이 있다. 이에 계산량을 줄이면서 빠른 탐색속도를 얻기 위한 3 단계 탐색법 (3SS)[2], 2D-LOG 탐색법[3], 블록기반 경사 하강 탐색(BBGDS)[4] 등의 알고리즘이 제시되어 왔지만 이러한 알고리즘들은 평균 절대 오차(Mean Absolute Error, MAE) 공간에

서 오차가 전역적 최소해에 근접할수록 단조 감소한다는 가정을 바탕으로 하고 있어서 탐색 영역내에 여러 최소점이 있을 경우 국소 최소점에 고립될 가능성이 있다.

본 논문에서는 이러한 문제점을 개선시키기 위해서 검색 영역 안에서 가능한 모든 블록들을 검색하되, 샘플링에 의해 선택된 화소들만 계산에 활용함으로써 계산량을 줄이는 방법을 제안한다. 프레임을 구성하는 각 화소 간의 기울기를 비교해서, 기울기의 변화량이 큰 화소를 샘플링하여 연산 대상으로 지정하고 움직임 벡터를 찾아낸다. 샘플링은 선택된 화소들의 개수에 따라 적응적으로 이루어진다.

본 논문은 다음과 같이 기술된다. 제 2 장에서는 블록 정합 알고리즘과 전역 탐색 정합에 대하여 기술하였고, 제 3 장에서 제안하는 적응 샘플링을 통한 블록 움직임 추정에 대해 기술된다. 이어서 제 4 장에서는 제안된 알고리즘에 의한 실험 결과와 종래의 전역 탐색 알고리즘 등과의 성능 비교를 하였으며, 제 5 장에서 결론을 맺는다.

2. BMA(Block based Matching Algorithm)

동영상의 연속한 프레임들 사이에는 유사성이 존재한다. 이전 프레임(previous frame)과 현재 프레임(current frame)을 블록 단위로 비교를 하여, 가장 일치하는 블록의 위치를 움직임 벡터로 산출하여, 현재 프레임을 복원할 때, 이전 프레임의 일치되는 블록을 이용하는 방식이 그림 1에서 보여지는 블록 정합 알고리즘(block based matching algorithm)이다[1]. 이 중에 전역 탐색 블록 정합(Full Search Block Matching, FSBM)은 현재 움직임 벡터를 찾을 블록과, 비교할 탐색 영역(search window)내에 있는 모든 화소에 대해서 정합 비교를 하기 때문에, 정합이 가장 잘 되는 블록의 위치를 찾아내 최적의 움직임 벡터를 산출해낸다.

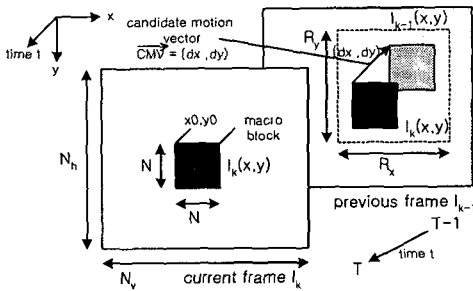


그림 1. block based matching algorithm

이 때, 화소값의 차이를 (식 1)을 이용하여 MAE 를 구한 후, 이 값들을 비교하여 가장 적은 차이를 나타내면 정합이 가장 잘 되는 블록이므로 이 위치를 움직임 벡터로 지정하게 된다.

$$MAE = \frac{1}{N \cdot N} \sum_{m=x}^{x+N-1} \sum_{n=y}^{y+N-1} |I_k(m,n) - I_{k-1}(m+dx, n+dy)| \quad (식 1)$$

탐색 영역이 $[p-1, -p]$ 라 가정하면, 한 블록에 대해서 $((p-1) - p)^2$ 만큼의 정합 연산이 필요하다. 초당 프레임을 f_s 라 하고 프레임의 행과 열의 길이를 N_h, N_w 로 하면 초당 정합 연산을 구하는 식은 (식 2)와 같이 나타난다.

$$N_w N_h (2p-1)^2 \cdot f_s \quad (식 2)$$

이러한 방식으로 움직임 추정을 할 경우 모든 화소에 대해 정합 여부를 수행하므로, 최적의 결과를 얻을 수 있으나, 움직임 추정에 사용되는 일반적인 사양으로도 많은 계산량이 필요하다. 현재 찾고자 하는 블록이 16x16 화소로 되어있고, 720x480 의 한 프레임에는 1350 개의 블록이 존재하고, 탐색 범위를 나타내는 p 를 8 화소의 범위로 지정하고, 초당 30 장의 프레임을 처리할 경우는 한 블록당 다음 (식 2)와 같은 수의 연산량을 갖게 된다.

$$1350 \cdot 30 \cdot (7 - (-8))^2 = 9112500 \quad (식 3)$$

또한 정합 연산에는 기본적으로 감산(differenciation), 누적(accumulation), 절댓값(absolute) 연산등이 사용되고, 이를 화소 단위의 계산량을 보면 (식 4)와 같이 된다.

$$N_w N_h (2p-1)^2 \cdot f_s \cdot 3 \cdot 16^2 \quad (식 4)$$

(식 3)에서 구한 값을 (식 4)에 대입시켜보면 연산량은 약 7GOPS(billion operation per second)에 까지 이른다. FSBM 은 많은 연산을 필요로 하기 때문에 부분 탐색이나, 부분 정합 등을 이용한 알고리즘들이 많이 발표되고 있다.

3 장에서는 이러한 블록 정합에 사용되는 대상 화소들을, 화소값을 이용한 적응적인 샘플링을 이용하여 추출된 화소들로부터 제한하여 연산에 사용함으로써 계산량을 줄이는 알고리즘을 기술한다.

3. 기울기 기반 적응 샘플링 블록 정합 (Gradient-based Adaptive Sampling Block Matching, GASBM)

일반적으로 영상을 샘플링 할 경우 그림 2에서 보이는 바와 같이 샘플링 비율(sampling ratio)에 따라 일정한 비율로 몇 화소 마다 하나의 화소를 읽을 것인지 정하게 된다. 예를 들어 1/4 서브 샘플링을 한다고 하면 16x16 으로 이루어진 매크로 블록(Macro Block : MB)의 경우 상하 좌우로 4 화소 마다 하나의 화소를 읽게 된다.

이 때 영상을 복잡한 그림이 있는 부분과 변화가 별로 없는 단순한 부분으로 나눈 다음, 복잡한 부분에서는 많은 수의 화소값을 읽어들이고, 단순한 부분에서는 적은 수의 화소값을 읽어 들여, MAE 를 구하면 전체 영상을 균일하게 샘플링 하는 것보다 적은 샘플링 만으로도 효과적으로 최소 MAE 를 찾을 수 있다. 복잡한 영상과 단순한 영상은 각각의 화소값을 인접 화소값과 비교했을 경우 차이가 많이 나는 것이 일정 영역에 얼마나 존재하느냐에 따라 나눌 수 있다.

3.1 적응 샘플링(adaptive sampling)

p_k 를 밝기(luminance) 값으로 나타낸 현재 화소값이라고 하고 p_{k-1} 는 화소를 읽어들이는 순서에 따라 현재 화소인 p_k 를 읽기 바로 이전에 읽어 들인 인접 화소로 한다. 예를 들어 x 축 방향으로 화소를 검색할 경우 p_{k-1} 는 x 축 방향의 이전 인접 화소를 가르킨다. 현재 화소에서의 기울기 G_c 와 인접한 화소에서의 기울기 G_p 는 (식 5), (식 6)와 같이 표현된다.

$$G_p = p_{k-1} - p_{k-2} \quad (식 5)$$

$$G_c = p_k - p_{k-1} \quad (식 6)$$

샘플링 할 화소를 정하기 위해서는 (식 5), (식 6)에서 표현된 기울기의 변화량을 이용하는데 그 변화량은 현재 계산한 기울기에서 이전에 계산해 두었던 기

울기를 뺀 $G_c - G_p$ 값이다.

$$|G_c - G_p| > THR \quad (식 7)$$

(식 7)에서 나타낸 바와 같이 기울기의 변화량인 $G_c - G_p$ 값의 절대치가 경계값보다 크면 그 화소의 위치를 샘플링 위치(sampling point)로 지정해 준다.

이러한 과정을 참조 프레임(reference frame)에 적용한다. 일반적으로 x 축 방향으로 화소값을 읽기 때문에 x 축 방향의 기울기의 변화값을 이용해 샘플링 위치를 정하였는데, 이 경우 하나의 라인에 있어 처음 화소와 제일 마지막 화소는 기울기의 변화량을 구할 수가 없다. 따라서 이 두 화소는 샘플링 위치로 지정이 되지 않기 때문에 강제로 샘플링 위치로 지정한다.

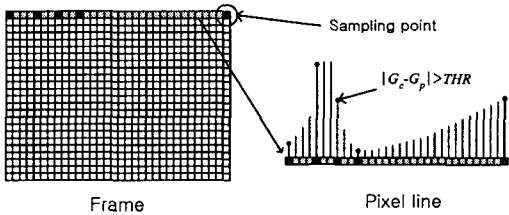
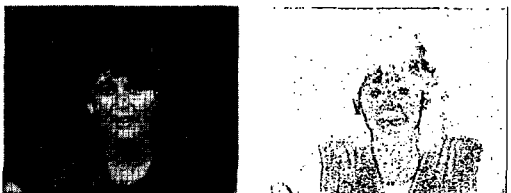


그림 3. 기울기 기반 적응 샘플링 방법

그림 3은 (식 7)을 이용하여 프레임 내의 한 행을 x 축 방향으로 검색하여 샘플링 위치를 찾아낸 예이다. 그런데 이렇게 기울기의 변화량을 이용해 샘플링 위치를 추출해내는 경우 밝기의 변화가 적은 부분에서는 샘플링 수도 적기 때문에 정확한 밝기값을 얻기 힘들다. 이런 부분은 변화가 적은 부분이므로 전체적인 밝기만 확인하면 되기 때문에, 이 부분에 한하여 MAE 를 구할 때 이전까지 구한 샘플링 화소를 무시하고 3 장 도입부에 설명된 일반적인 샘플링 방법을 이용해 일정 개수만큼의 화소만을 샘플링 한다.



(a) Flower Garden 원영상 (b) (a)의 샘플링 영상



(c) Miss America 원영상 (d) (c)의 샘플링 영상

그림 4. 샘플링 위치를 찾은 영상 예

이 방법으로 전체 프레임을 모두 검색하여 샘플링 위치를 표시한 영상이 그림 4에 예시되어 있다. 그림 4의 (b)와 (d)는 적응 샘플링 방식으로 위치를 지정한 화소를 나타낸 영상이다.

3.2 적응 샘플링 방식에서의 MAE의 계산

3.1에서 제안한 방식으로 이전 프레임에 샘플링 위치를 정하였으나, 현재 프레임의 MB에는 이전 프레임에서 얻은 샘플링 위치의 좌표에 해당하는 현재 MB의 화소값을 읽어 들인다. 이전 프레임에서 검색 영역 안의 후보 MB(candidate MB)에는 샘플링 위치가 정해져 있으므로 후보 MB와 현재 프레임의 MB를 비교할 때 현재 프레임의 MB에서도 이 위치의 화소만을 읽어 들여 두 MB 사이의 MAE를 구한다.

$$MAE = \frac{1}{N_p} \sum_{m=x}^{x+N-1} \sum_{n=y}^{y+N-1} P_{mn} \cdot \{|I_x(m, n) - I_{x-1}(m+dx, n+dy)\}$$

$$P_{mn} = \begin{cases} 1 & \text{if sampling point on } (m+dx, n+dy) \text{ exists} \\ 0 & \text{otherwise} \end{cases} \quad (식 8)$$

(식 8)에서 P_{mn} 은 이전 프레임에서 지정된 샘플링 위치의 화소에서 1이고, 그렇지 않다면 0이다. N_p 은 MB 내부에서 샘플링 하는 화소의 개수이다. MAE를 읽기 전에 먼저 MB 안에서 샘플링 한 화소의 개수를 구할 때, 샘플링한 화소의 개수가 경계값 이상으로 크지 않다면 3.1에서 설명한 바와 같이 이 MB에 한하여 지금까지 구한 샘플링 화소를 무시하고 일반적인 샘플링 방법을 이용해 일정 개수만큼의 화소만을 샘플링 하도록 한다.

지금까지의 방법을 순서대로 나타내면 그림 5와 같다.

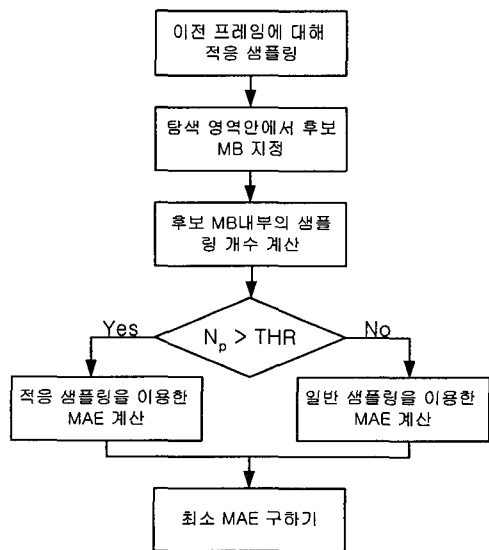
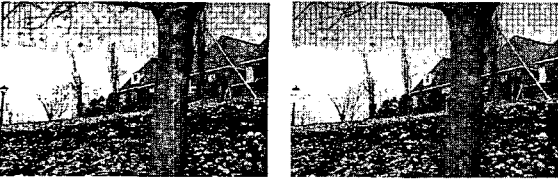


그림 5. GASBM의 순서도

그림 6은 제안한 알고리즘을 이용하여 움직임 벡터를 구하고 이를 이용해 현재 프레임에 복원한 영상을 보여준다.



(a) Flower Garden 원영상 (b) 복원한 영상

그림 6. 원영상과 복원한 영상

4. 결과와 고찰

제안한 알고리즘의 검증에는 Flower garden, Football, Miss America, Table tennis의 1~30 프레임의 영상을 이용하였다. 3.1에서 설명한 샘플링 위치를 찾기 위한 기울기의 변화량에 대한 경계값과 3.2에서 설명한 샘플링 방식을 정하기 위한 샘플링한 화소 수의 경계값은 실험적으로 구하여 모든 영상에 동일하게 적용하였다.

제안한 알고리즘을 통해 얻은 움직임 벡터를 이용하여 영상을 복원시키고 PSNR과 연산량을 계산하여 다음과 같이 표 1, 2, 3으로 나타내었다.

(dB)	Flower	Football	MissA	Tennis
FSBM	22.58	25.16	37.95	28.97
3SS	21.97	23.62	37.82	27.51
GASBM	22.19	24.54	37.15	28.20

표 1. PSNR(dB)

(%)	Flower	Football	MissA	Tennis	Average
FSBM	100	100	100	100	100
3SS	97.30	93.88	99.66	94.96	96.45
GASBM	98.27	97.54	97.89	97.34	97.76

표 2. PSNR의 비교(%)

시뮬레이션한 3SS, FSBM과 GASBM의 PSNR은 표 1로 나타내었다. 기존의 타 알고리즘들과 PSNR을 비교하기 위해 탐색 영역의 크기, 프레임의 수 등의 동일한 조건 하에서의 실험 결과가 필요하나, 각 알고리즘의 특성이 다르기 때문에 FSBM의 PSNR 값에 대한 상대 성능을 % 값으로 비교하여 표 2와 같이 정리하였다.

PSNR 값을 비교해 보면 평균적으로 GASBM이 3SS보다 나왔으나 움직임이 적은 MissA에서는 3SS에 비해 좋지 않은 결과가 나왔고 움직임이 큰 Flower나 Tennis의 경우는 그보다 좋은 결과가 나왔

다. GASBM은 전체적으로 3SS보다 개선되고 비교적 균일한 PSNR을 나타내므로, 영상에 따라 움직임 벡터의 정확도 차이가 큰 3SS에 비해 안정적인 정확도를 얻을 수 있다. 따라서 다양한 영상을 처리하는 움직임 추정 시 제안된 GASBM이 더 안정적인 효율을 나타낸다고 할 수 있다.

표 3은 FSBM을 기준으로 하여 그에 대한 비율로 나타낸 각 알고리즘들의 연산량이다.

(%)	Flower	Football	MissA	Tennis	Average
FSBM	100	100	100	100	100
3SS	16.65	16.50	16.54	16.55	16.56
GASBM	48.85	17.49	14.91	51.37	31.52

표 3. 연산량(%)

표 3에서 보는 바와 같이 복잡한 형태의 영상이 많은 Flower의 경우 연산량이 FSBM의 49% 가량으로 3SS보다 많았으나 단순한 형태의 MissA의 경우는 이에 비해 계산량이 적은 것을 볼 수 있다.

5. 결론

전역 탐색 블록 정합은 최적의 성능을 나타내지만, 많은 계산량을 갖는 단점이 있기 때문에 다양한 고속 탐색 알고리즘들이 연구되었다. 본 논문에서 제안한 GASBM 알고리즘은 동영상의 한 프레임 내에서 계산할 화소를 적응적으로 샘플링하여 정합을 수행한다. 화소값의 차를 이용한 기울기의 변화량을 활용하여 영상이 복잡한 부분에서는 많은 수의 화소를 샘플링하고, 단순한 부분에서는 적은 수의 화소를 샘플링하여, 전체 계산량을 감소시키고 기존의 고속 탐색 알고리즘보다 안정되고 개선된 PSNR 값을 얻었다. 따라서 제안된 GASBM은 다양한 영상을 처리하는 부호기에서 더 안정적인 효율을 나타낼 수 있다.

참고문헌

- [1] Peter Kuhn, "Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation", Kluwer Academic Publishers, 1999
- [2] T. Koga, K. Iinuma, A. Iijima, and T. Ishiguro, "Motion-Compensated Interframe Coding for Video Conferencing," in Proc. NTC81, New Orleans, LA, 1981, pp. C9.6.1-9.6.5.
- [3] J.R.Jain and A.K. Jain, "Displacement measurement and its application in interframe image coding", IEEE Trans. commun., vol.COM-33, pp.888-896, Aug, 1985
- [4] L.-K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," IEEE Trans. Circuits Syst. Video Technol., vol. 6, no. 4, pp. 419-422, 1996.
- [5] Jer Min Jou, Pei-Yin Chen, and Jian-Ming Sun, "The Gray Prediction Search Algorithm for Block Motion Estimation", IEEE Transactions on Circuit and Systems for Video Technology, Vol. 9, No. 6, september 1999 843