

초기 탐색 위치의 효율적 선택에 의한 고속 움직임 추정

*남 수 영, *김 석 규, **임 채 환, *김 남 철
*경북대학교 전자공학과, **삼성전자 무선사업부

Fast Motion Estimation Using Efficient Selection of Initial Search Position

Soo-Young Nam, Seog-Gyu Kim, Chae-Whan Lim, and Nam-Chul Kim
Dept. of Electronics Engineering, Kyungpook National University
E-mail : salgoo@vcl.knu.ac.kr

Abstract

In this paper, we present a fast algorithm for the motion estimation using the efficient selection of an initial search position. In the method, we select the initial search position using the motion vector from the subsampled images, the predicted motion vector from the neighbor blocks, and the (0,0) motion vector. While searching the candidate blocks, we use the spiral search pattern with the successive elimination algorithm(SEA) and the partial distortion elimination(PDE). The experiment results show that the complexity of the proposed algorithm is about 2~3 times faster than the three-step search(TSS) with the PSNR loss of just 0.05[dB]~0.1[dB] than the full search algorithm PSNR. The search complexity can be reduced with quite a few PSNR loss by controlling the number of the depth in the spiral search pattern.

I. 서론

H.263[1], MPEG[2] 등의 다양한 비디오 부호화 표준에서는 움직임 벡터 추정에 블록정합 알고리즘(block matching algorithm)이 사용되는데, 전통적인 블록정합 알고리즘인 전역탐색 블록정합 알고리즘(FBMA : full search BMA)[3]은 탐색 영역(search range) 안에서 모든 움직임 후보 벡터들을 탐색하기 때문에 많은 계산량을 요구하므로 동영상 부호화에 큰 부담이 된다. 그러므로 계산량의 부담을 줄일 수 있는 고속의 알고리즘들이 연구되어 왔다.

FBMA에 대해 성능저하 없이 계산량만을 줄이는 알고리즘으로 연속소거 알고리즘(successive elimination algorithm)[4], 부분정합 왜곡 소거법(partial distortion elimination)[5] 등이 있고, 후보 블록을 탐색하는 패턴에는 회전 탐색 패턴(spiral search)[6], 다이아몬드 탐색 패턴(diamond search)[7] 등이 있다. 성능 저하를 감소하면서 계산량을 대폭 줄인 고속 알고리즘으로는 3단계 탐색법(three-step search)[8], 2차원 대수 탐색법(2-dimensional

logarithm search)[9] 등이 있다. 그러나 후자의 고속의 알고리즘은 FBMA에 대해 성능저하가 매우 크고, 전자와 같이 성능저하가 없는 고속 알고리즘은 계산량을 줄이는 데 한계가 있다. 따라서 약간의 성능 저하를 허용하면서, 계산량을 후자의 고속 알고리즘과 같거나 그 이상으로 줄임으로써 실시간 동영상 전송에 적합한 알고리즘의 연구가 필요하다.

본 논문에서는 초기 탐색 위치를 효율적으로 선택한 후, 탐색영역 내에서 회전 탐색하면서 연속소거 알고리즘과 부분정합 왜곡 소거법을 사용하여 FBMA에 대하여 0.1dB 이하의 미소한 성능저하를 보이면서 3단계 탐색법에 비해 계산량이 크게 감소되는 알고리즘을 제안한다. 초기 탐색 위치는 부표본화된 프레임으로부터 얻은 움직임 벡터와 주위 블록의 움직임을 이용한 예측 움직임 벡터, 그리고 탐색 영역의 중심점에 해당하는 각각의 위치 중에서 정합 오차가 가장 작은 것을 선택한다. 제안된 방법에서 얻은 초기 탐색 위치는 FBMA에서 얻은 움직임 벡터에 해당하는 위치에 매우 근접하기 때문에 연속소거 알고리즘과 부분정합 왜곡 소거법의 효과를 더욱 높이며, 성능저하가 거의 없이 탐색영역을 대폭 줄일 수 있음을 확인하였다.

II. 블록 정합에 의한 움직임 추정

2.1 블록정합 알고리즘

블록정합 알고리즘에서는 영상을 일정한 크기의 작은 블록 단위로 나눈 다음 현재 프레임의 참조 블록과 이전 프레임의 후보 블록 사이의 정합오차(또는 유사도)를 비교하여 최소값(또는 최대값)을 갖는 상대위치를 찾아 물체의 움직임을 찾는다.

블록정합 알고리즘에서 현재 프레임의 참조 블록과 이전 프레임의 후보 블록 사이의 정합오차 계산에는 일반적으로 다음과 같이 주어지는 MSE(mean square error)와 MAD(mean absolute difference)를 사용한다.

$$MSE(i, j) = \langle (X - Y_{i,j})^2 \rangle \quad (1)$$

$$MAD(i, j) = \langle |X - Y_{i,j}| \rangle \quad (2)$$

(1)식과 (2)식에서 이동 범위 (i, j) 는 탐색영역 내로 제한되고, $\langle \cdot \rangle$ 연산은 블록 A 및 B에 관하여 다음과 같이 수행된다.

$$\langle |A - B| \rangle = \frac{1}{B^2} \sum_{l=0}^{B-1} \sum_{k=0}^{B-1} |A(k, l) - B(k, l)|$$

$$\langle (A - B)^2 \rangle = \frac{1}{B^2} \sum_{l=0}^{B-1} \sum_{k=0}^{B-1} (A(k, l) - B(k, l))^2$$

2.2 연속 소거 알고리즘

Li와 Salari에 의해 제안된 연속 소거 알고리즘[4]은 현재 프레임의 참조 블록과 이전 프레임의 후보 블록들 각각의 평균을 구하고, 참조 블록 평균과 후보 블록 평균의 차를 구하여 초기 정합오차와 비교, 부등식을 만족하지 않는 후보 블록들을 제거해가며 최소 MAD를 갖는 움직임 벡터를 탐색한다.

연속소거 알고리즘에서 사용된 부등식[10]은 다음과 같다.

$$|\langle X \rangle - \langle Y_{i,j} \rangle| \leq MAD(i, j) \quad (3)$$

탐색영역 내에서 이전까지 구한 $MAD(i, j)$ 중 최소인 값을 MAD_0 라 할 때, 다음의 관계를 만족하는 후보 블록만이 최소 MAD를 가질 수 있으므로, 이 관계를 만족하는 후보 블록에 대해서만 탐색을 진행한다.

$$MAD(i, j) \leq MAD_0 \quad (4)$$

즉, (3)식과 (4)식을 결합하여 얻은 다음 식을 만족하는 후보 블록만 탐색한다.

$$|\langle X \rangle - \langle Y_{i,j} \rangle| \leq MAD_0 \quad (5)$$

따라서 FBMA에 대하여 PSNR을 저하시키지 않으면서, 계산량을 줄일 수 있다.

2.3 부분 정합 왜곡 소거법

부분정합 왜곡 소거법[5]에서는 블록간 정합오차를 계산하는 과정에서 부분 정합오차가 이전까지의 최소 정합오차보다 크면, 그 블록에 대한 정합오차 계산을 그만두고 다음 후보 블록으로 옮겨 탐색을 계속한다. 이때, 부분 정합오차는 다음과 같이 주어진다.

$$PMAD_{i,j}(m, n) = \frac{1}{B^2} \sum_{l=0}^m \sum_{k=0}^n |X(k, l) - Y_{i,j}(k, l)| \quad (6)$$

(6)식은 참조 블록과 (i, j) 의 움직임 벡터를 갖는 후보 블록에 대하여 (m, n) 위치 화소까지의 부분 정합오차이고, 각 블록에서 대응되는 위치의 화소 값을 절대차 누적시킨 것이므로 단조증가 함수이다. 따라서 다음과 같이 부분 정합오차가 이전까지 구해진 최소 정합오차 MAD_0 보다 크면, 후보 블록의 $MAD(i, j)$ 는 MAD_0 보다 크다.

$$PMAD_{i,j}(m, n) \geq MAD_0 \quad (7)$$

따라서 현재의 후보 블록을 버리고 다음 후보 블록에서 탐

색을 계속하여, 계산량을 줄인다.

2.4 회전 탐색 패턴

회전 탐색 패턴[6]은 움직임 벡터가 탐색영역의 중심에 많이 분포함을 이용하여 탐색영역의 중심으로부터 정사각형 모양으로 회전하면서 후보 블록을 탐색한다. 이 때, 탐색영역의 중심이란 현재 프레임의 참조 블록과 대응되는 이전 프레임 후보 블록의 위치를 말한다. 최소 정합오차를 갖는 후보 블록의 위치가 탐색 영역의 중심에 가깝게 있을 때, 회전 탐색 패턴을 연속소거 알고리즘이나 부분정합 왜곡 소거법과 함께 사용하면, (5)식과 (7)식에서 MAD_0 가 최소 정합오차에 빨리 가까워지고 부등식이 제한하는 범위를 더욱 줄여 계산량을 줄일 수 있다.

III. 초기 탐색 위치 선택을 이용한 고속 움직임 추정

일반적으로 회전 탐색에서 초기 탐색 위치는 탐색영역의 중심 (0,0)벡터에 대응되는 위치를 사용한다. 그러나 움직임이 큰 영상의 경우 이러한 탐색은 최소 정합오차를 갖는 움직임 벡터를 찾는 데 많은 계산량이 요구된다. 그러므로 움직임을 예측하여 초기 탐색 위치를 적절히 선택하고, 연속소거 알고리즘이나 부분정합 왜곡 소거법을 사용하면, (5)식과 (7)식이 제한하는 범위를 효과적으로 줄일 수 있으므로 계산량을 대폭 줄일 수 있다. 제안된 방법에서는 부표본화된 프레임으로부터 얻은 움직임 벡터, 주위 블록의 움직임을 이용한 예측 움직임 벡터, 그리고 탐색 영역의 중심에 해당되는 (0,0)벡터에 각각 대응되는 탐색위치 중에서 가장 작은 정합오차를 갖는 것을 초기 탐색 위치로 선택하고, 이 위치를 시작으로 회전 탐색을 진행하여 움직임을 추정함으로써 계산량을 대폭 줄이면서도 성능저하는 크지 않도록 하였다.

3.1 초기 탐색 위치의 선택

부표본화를 이용한 초기 움직임 벡터의 선택은 그림 1과 같이 이루어진다. 먼저 현재 프레임과 이전 프레임을 각각 수평 및 수직 방향으로 2 : 1 부표본화하여 얻은 영상에 대하여, 블록정합 알고리즘으로 각 블록의 움직임 벡터를 구한다. 이 때, 정합되는 블록의 크기 및 탐색 영역은 원영상의 것들에 비해 1/4로 축소된 상태이며, 회전 탐색 패턴으로 연속소거 알고리즘과 부분정합 왜곡 소거법을 사용한다. 다음으로, 이렇게 얻은 부표본화된 영상의 움직임 벡터를 두 배로 확대하여 초기 움직임 벡터 (x_1, y_1) 를 얻는데, 이것은 원 블록의 대략적 움직임을 잘 나타내게 된다.

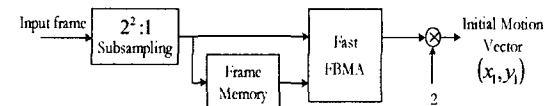


그림 1. 부표본화를 이용한 초기 움직임 벡터 선택

한편, H.263 등의 동영상 부호화 표준에서는 주위 블록의 움직임을 이용한 예측 움직임인 벡터[1]를 사용하는데, 이것은 참조 블록과 주위 블록의 움직임이 높은 상관성을 갖는다는 것을 바탕으로 주위 블록의 움직임 벡터에서 가운데 값

을 선택하여 예측 움직임을 얻는다. 그림 2는 이때 사용되는 주위 블록의 움직임 벡터들을 나타낸다. 여기서 점선은 프레임(picture or GOB)의 경계를 나타내며, (a), (b), 그리고 (c)는 각각 참조 블록이 프레임의 경계에 위치하여 주위 블록이 없는 경우의 주위 움직임 벡터를 결정하는 방법을 보여준다. 참조 블록의 예측 움직임 벡터 MV 의 수직, 수평 성분은 각각 다음과 같이 구한다.

$$x_2 = \text{Median}(x_{MV1}, x_{MV2}, x_{MV3}) \quad (8-a)$$

$$y_2 = \text{Median}(y_{MV1}, y_{MV2}, y_{MV3}) \quad (8-b)$$

여기서 x_{MV_i} 및 y_{MV_i} 는 주위 블록의 움직임 벡터 MV_i 의 x 및 y 성분을 나타낸다. 이렇게 얻어진 예측 움직임 벡터는 여러 블록들이 같거나 유사한 방향성을 갖는 경우에는 매우 효과적이지만, 각 블록의 움직임간에 상관성이 적은 영상에서는 잘못된 예측을 하는 경우도 있다.

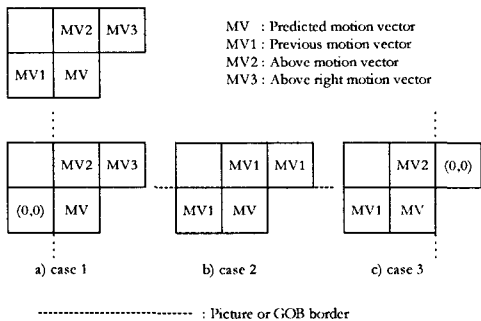


그림 2. 초기 탐색 위치의 예측[1]

따라서 블록의 대략적 움직임을 잘 나타내는 부표본화를 이용한 초기 움직임 벡터 (x_1, y_1) 와 주위 블록의 움직임을 이용한 예측 움직임 벡터 (x_2, y_2) , 그리고 움직임 벡터가 탐색 영역의 중심에 많이 분포함을 이용하기 위한 (0,0) 벡터를 모두 사용하여 초기 움직임 예측을 하는 것이 더 효과적이다. 제안된 방법에서는 위에서 얻은 세 가지 초기 탐색 벡터들 중에서 다음과 같이 가장 작은 정합 오차를 갖는 움직임 벡터 (x, y) 에 대응되는 위치를 초기 탐색 위치로 이용하였다.

$$(x, y) = \arg \min_{i=0,1,2} MAD(x_i, y_i) \quad (9)$$

여기서 $(x_0, y_0) = (0, 0)$ 이다. 그림 3은 연속 영상 STEFAN.QCIF 33프레임에 대하여 제안된 방법으로 얻은 초기 움직임 벡터와 FBMA로 얻은 움직임 벡터 사이의 차벡터 분포를 나타낸다. x축은 차벡터의 수평 성분 절대값, y축은 차벡터의 수직 성분 절대값, 그리고 z축은 전체 차벡터 수에 대한 같은 값을 갖는 차벡터 수의 백분율[%]을 나타낸다. 그림 3으로부터 제안된 방법이 움직임 벡터에 대응되는 탐색 위치에 상당히 근접한 초기 탐색 위치를 선택하는 방법으로 매우 효과적인 것임을 알 수 있다.

3.2 제안된 알고리즘의 흐름도

제안된 알고리즘에서는 그림 4와 같이 한 프레임에 대한

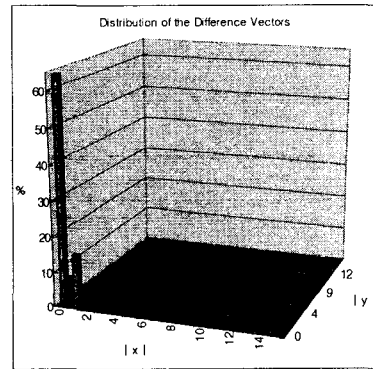


그림 3. 제안된 방법으로 얻은 초기 움직임 벡터와 FBMA로 얻은 움직임 벡터 사이의 차벡터 분포

움직임 벡터 추정을 시작하기 전에 전처리 과정으로 각 참조 블록의 평균 $\langle X \rangle$ 와 각 후보 블록의 평균 $\langle Y_{i,j} \rangle$ 을 구한다. 그리고 블록의 대략적인 움직임을 잘 나타내는 부표본화로 얻은 움직임 벡터 (x_1, y_1) 에 대하여 $MAD(x_1, y_1)$ 를 먼저 구하고, 이 값을 최소 정합오차 MAD_0 로 하여 이웃 블록의 움직임을 이용한 예측 움직임 벡터 (x_2, y_2) 를 이용한 $MAD(x_2, y_2)$ 와 $MAD(x_0, y_0)$ 를 구하여 가장 작은 정합 오차를 갖는 움직임 벡터를 선택한다. 이렇게 얻어진 초기 탐색 위치를 중심으로 회전 탐색을 시작하며, 이전까지 얻은 최소 정합오차 MAD_0 는 $MAD(x, y)$ 로 한다. 회전 탐색 패턴으로 연속소거 알고리즘(SEA)을 이용하여 탐색할 후보 블록을 선택하고, 선택된 후보 블록과의 정합 오차를 계산하는 과정에서 블록내 각 줄에 대하여 부분정합 왜곡 소거법(PDE)를 이용하여 정합 오차 계산을 진행할 것인가를 결정한다.

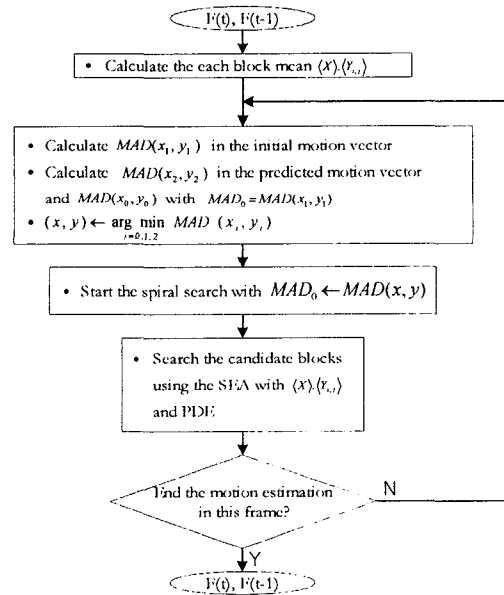


그림 4. 제안된 알고리즘의 흐름도

IV. 실험 결과 및 검토

본 논문에서 제안된 방법과 기존 방법의 성능 평가를 위하여 설정된 실험환경은 표 1과 같다. 그리고 성능평가의 척도는 다음과 같이 PSNR을 사용하였고, 복잡도(계산량) 비교를 위하여 덧셈과 나눗셈의 사용된 연산 수를 사용하였다.

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

(10)식에서 MSE는 다음과 같이 얻어진다.

$$MSE = \frac{1}{A \times B} \sum_{i=0}^{A-1} \sum_{j=0}^{B-1} \{X(i,j) - \hat{X}(i,j)\}^2 \quad (11)$$

(11)식에서 $X(i,j)$ 는 현재 프레임의 (i,j) 위치의 화소값을, $\hat{X}(i,j)$ 는 움직임 추정 및 보상에 의하여 얻은 현재 프레임의 (i,j) 위치 화소값을 나타내며, A 와 B 는 각각 프레임의 가로, 세로 화소 수를 나타낸다.

표 1. 실험 환경

사용 언어	Borland C++ Builder 4.0
프레임 크기	QCIF(176 × 144)
블록 크기	16×16 화소
탐색 범위	[-15,+15]×[-15,+15]
원영상 프레임 속도	30프레임/초
처리 프레임 속도	10프레임/초
프레임 수	33프레임

표 2는 영상 SUZIE.QCIF 33프레임에 대하여 FBMA와 3단계 탐색법, 연속소거 알고리즘, 그리고 제안된 알고리즘을 사용한 실험의 성능과 복잡도이다. 여기서 제안된 알고리즘은 선택한 초기 탐색 위치를 중심으로 회전 탐색 패턴의 탐색 범위 $[-S,+S] \times [-S,+S]$ 에서 S 를 6과 2로 하여 탐색 영역을 제한하였다. 표 3은 STEFAN.QCIF에 대하여 각 알고리즘을 사용한 실험의 성능과 복잡도이다.

실험 결과로부터 S 를 6으로 한 제안된 알고리즘을 사용하였을 경우 FBMA에 비하여 0.06dB 이하의 적은 평균 PSNR 저하만이 발생하면서, 연산수는 평균적으로 3단계 탐색법보다 적거나 거의 같다는 것을 볼 수 있다. 그리고 S 를 2로 하였을 경우에는 각 영상에 대하여 0.1dB 이하의 평균 PSNR 저하를 발생시키면서 3단계 탐색법에 비하여 30%~50%의 연산수를 요구하였다.

V. 결론

본 논문에서는 부표본화를 이용하여 얻은 초기 움직임 벡터와 주위 블록의 움직임을 이용한 예측 움직임 벡터, 그리고 (0,0) 벡터를 이용하여 초기 탐색 위치를 효율적으로 선택하는 고속 알고리즘을 제안하였다. 제안된 알고리즘에서는 선택한 초기 탐색 위치를 중심으로 회전 탐색 패턴으로 연속소거 알고리즘을 사용하여 후보 블록을 선택하고, 선택된 후보 블록과의 정합 오차를 계산하는 과정에서 부분정합 왜곡 소거법을 사용하여 계산량을 대폭 줄였다.

제안된 방법으로 실험한 결과 회전 탐색의 탐색 영역을 6으로 하였을 경우 FBMA에 대하여 평균 0.06[dB]이하의 PSNR 저하를 일으키면서 3단계 탐색법과 비슷한 평균 복잡도를 나타내고, 탐색 영역을 2로 하였을 경우 FBMA에

표 2. 보상된 영상에 대한 평균 PSNR과 평균 복잡도 비교 (SUZIE.QCIF)

Algorithms	FBMA	TSS	Salari's SAE	Proposed algorithm(S-6)	Proposed algorithm(S-2)
PSNR[dB]	22.40	21.81	22.40	22.34	22.31
Compl [10 ⁵]	594	41.7	143.4	42.2	20.3
-exity [%]	100	7.0	24.1	7.1	3.4

표 3. 보상된 영상에 대한 평균 PSNR과 평균 복잡도 비교 (STEFAN.QCIF)

Algorithms	FBMA	TSS	Salari's SAE	Proposed algorithm(S-6)	Proposed algorithm(S-2)
PSNR[dB]	31.69	31.22	31.69	31.66	31.63
Compl [10 ⁵]	594	41.7	67.7	29.5	14.1
-exity [%]	100	7.0	11.4	5.0	2.4

대하여 평균 0.1[dB] 이하의 PSNR 저하만을 일으키면서 3단계 탐색법의 복잡도에 대하여 35%~50% 정도의 복잡도를 나타내었다. 따라서, 제안된 방법은 전송 환경에 따라 탐색 영역을 적절히 조절하여 적은 성능저하만으로 고속의 움직임 탐색을 가능하게 하는 알고리즘이다.

References

- [1] ITU-T DRAFT H.263, "Video Coding for Low Bitrate Communication," May. 1996.
- [2] ISO-IEC JTC1/SC2/WG11, "Preliminary text for MPEG video coding standard," ISO, Aug. 1990.
- [3] Y. H. Fok and O. Au, "A Fast Block Matching Algorithm in Feature Domain," *Proc. of IEEE Workshop on Visual Signal Processing and Communications*, Melbourne, pp. 199-202, Sep. 1993.
- [4] W. Li and E. Salari, "Successive Elimination Algorithm for Motion Estimation," *IEEE Trans. on Image Processing*, vol. 4, no. 1, pp. 105-107, Jan. 1995.
- [5] H. S. Wang and R. M. Mersereau, "Fast Algorithms for the Estimation of Motion Vectors," *IEEE Trans. on Image Processing*, vol. 8, no. 3, pp. 435-438, Mar. 1999.
- [6] T. Zahariadis and D. Kalivas, "Fast Algorithm for the Estimation of Block Motion Vectors," *Proc. of ICECS on Electronics, Circuits, and Systems '96*, vol. 2, pp. 716-719, 1996.
- [7] S. Zhu; K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. on Image Processing*, vol. 9, no. 2, pp. 287-290, Feb. 2000.
- [8] J. Lu and M. L. Liou, "A simple and efficient search algorithm for block-matching motion estimation," *IEEE Trans. on Circuits Syst. Video Technol.*, vol. 7, pp.429-433, Apr. 1977.
- [9] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. on Commun.*, vol.COMM-29, pp. 1799-1808, Dec. 1981.
- [10] D. M. Young and R. T. Gregory, *A Survey of Numerical Mathematics*. New York: Dover. pp.759-101, vol. 2, 1998.