

One-bit 변환을 기반으로 한 고속의 가변 블록 크기 움직임 예측 알고리즘

신동식*, 한재혁*, 박원배*, 안재형*

*충북대학교 정보통신공학과
e-mail:dsshin98@hotmail.com

Fast Variable-size Block Matching Algorithm for Motion Estimation Based on One-bit Transformation

Dong-Shik Shin*, Jea-Hyeck Han*, Won-Bae Park*, Jae-Hyeong Ahn*

*Dept of Computer and Communication Engineering
Chungbuk National University

요 약

본 논문에서는 One-bit 변환을 기반으로 한 고속의 가변 블록 크기 움직임 예측 알고리즘을 제안한다. 제안된 방법은 블록 내의 평균값을 이용하여 8bit 화소값을 1bit로 변환한 후 움직임 예측을 수행한다. One-bit 변환을 통한 영상의 단순화는 움직임 추정의 계산적 부담을 감소시켜 빠른 탐색을 가능하게 한다. 그리고 블록 내의 움직임 정도를 미리 판별하여 이를 기반으로 한 적응적 탐색이 불필요한 탐색을 제거하고 움직임이 큰 블록에서는 정합과정을 심화시켜 보다 정확한 움직임 예측을 수행한다. 본 제안된 방식을 가지고 실험한 결과 한 프레임당 적은 수의 블록으로 고정된 크기의 블록을 가진 전역 탐색 블록 정합 알고리즘(full search block matching algorithm; FSBMA)보다 예측 에러를 적게 발생시켜 평균적으로 0.5dB 정도의 PSNR 개선을 가져왔다. 특히, 움직임이 많은 영상에서 뛰어난 효과를 나타냈다.

1. 서론

멀티미디어 데이터는 오늘날 구축된 네트워크를 통해 폭넓게 사용되어지고 있다. 그러나 이러한 데이터는 매우 커다란 저장용량과 고속의 전송속도를 요구한다. 이를 해결하기 위해 멀티미디어 데이터 압축과 관련된 많은 연구가 되어오고 있다. 그 중에 하나가 동영상 압축이다. 동영상 압축은 많은 시간적 상관관계를 가지고 있는 프레임간의 시간적 중복성을 제거하는 움직임 예측을 통해 이루어진다. 이 움직임 예측에는 간단하고 효과적인 방식인 블록 정합 알고리즘(block matching algorithm; BMA)[1]이 주로 사용된다. 블록 정합 알고리즘은 한 블록 내의

모든 화소들이 동일한 움직임을 갖는다는 가정으로 시작한다. 블록의 크기가 커지면 커질수록 더 적은 수의 움직임 벡터를 전송해야 하지만 예측 에러값은 그에 반해 커지게 된다.

블록 정합은 이전 프레임의 탐색 영역 안에서 현재 프레임의 현재 블록과 가장 유사한 후보 블록을 찾는 과정이다. 전역 탐색 블록 정합 알고리즘(full search block matching algorithm; FSBMA)은 탐색 영역 안의 모든 위치를 전부 검사하는 방식으로 최적의 성능을 나타낸다. 그러나 많은 계산적 부담은 three-step 탐색[2], Cross 탐색 알고리즘[3]과 같은 탐색속도 개선측면의 블록 정합 알고리즘들의 동기가 되어왔다. 그리고 블록의 크기가 고정일 때 영상

의 국부적인 변화에 따른 움직임 효과를 효과적으로 반영하기 어렵고 낮은 전송률에서 블록화 현상 등의 화질 저하를 초래하는 것을 개선하기 위해 가변 블록 크기(variable block size) 모델도 연구되어 왔다[4]. 이 방식은 전역 탐색 블록 정합 알고리즘 보다 더 적은 예측 에러값을 발생시키지만 계산적 부담이 따른다.

본 논문에서의 블록 정합 알고리즘은 블록 내의 평균값을 이용하여 8bit 화소값을 1bit로 변환한 후 움직임 예측을 수행하도록 한다. One-bit 변환을 통한 영상의 단순화는 움직임 추정의 계산적 부담을 감소시켜 빠른 탐색을 가능하게 한다. 그리고 블록 내의 움직임 정도를 미리 판별하고 이를 기반으로 한 적응적 탐색을 수행하여 불필요한 탐색을 제거하고 움직임이 큰 블록에서는 정합과정을 심화시켜 보다 정확한 움직임 예측을 수행하도록 한다.

본 논문의 구성은 2장, 3장에서는 제안된 알고리즘에 대해 설명하고, 4장에서는 성능을 평가하며, 5장에서 결론을 맺는다.

2. 블록 정합 알고리즘

블록 정합 알고리즘은 먼저 현재 프레임을 $N \times N$ 의 겹쳐지지 않는 블록으로 분할한 후 그림 1처럼 분할된 각각의 블록에 대해 이전 프레임에서의 그 블록과 가장 비슷한 블록을 찾는 방법이다. $(N+2w) \times (N+2w)$ 는 두 프레임사이의 블록이 움직일 수 있는 최대 변위가 w 인 탐색 영역의 크기를 의미한다. 각각의 탐색 지점에서 MAD(mean absolute difference)를 사용하여 식 (1)과 같은 두 블록 사이의 차분인 DBD(displaced block difference)를 계산한다.

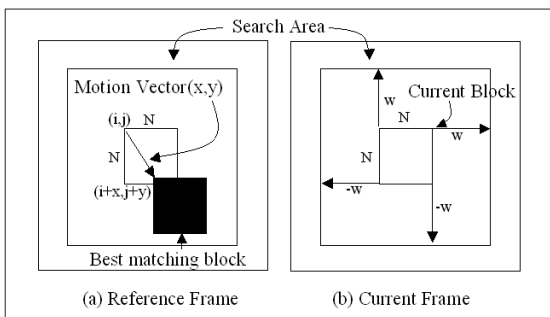


그림 1. 전역 탐색 방법

$$DBD(x,y) = \frac{1}{N \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |F_n(i,j) - F_{n-1}(i+x, j+y)| \quad (1)$$

$$x,y \in [-\omega, -\omega+1, \dots, \omega-1, \omega]$$

여기서 N 은 블록의 너비와 높이를, F_n 과 F_{n-1} 은 각각 현재 프레임과 이전 프레임을 나타낸다. (x,y) 는 오차를 최소화하는 움직임 벡터이다.

2.1 One-bit 변환

One-bit 변환은 정합과정에서의 계산량을 줄이기 위해 해당 블록과 탐색영역의 화소값을 1bit의 Bit-plane으로 바꾸는 것을 의미한다. 탐색 블록과 탐색 영역의 Bit-plane은 식 (2)를 통해서 얻어진다.

$$B(i,j) = \begin{cases} 1, & \text{if } x(i,j) \geq M \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

여기서 $x(i,j)$ 는 (i,j) 번째 화소값을 의미하고, M 은 그 블록의 평균 화소값을 의미한다. Bit-plane은 각각의 블록에 대해서 직관적인 블록의 정보를 나타낸다. 즉, 탐색 블록의 Bit-plane이 탐색 영역 안에 있는 블록의 Bit-plane과 매우 다르다면 그 두 블록은 유사한 이미지의 특징을 갖지 않고 있다는 것을 의미한다. 이를 기반으로 한 블록 정합은 FSBMA의 92~95%의 계산량 절감 효과가 있다[4]. 얻어진 Bit-plane을 통한 정합 과정은 다음과 같다.

- 단계 1 : 현재 블록과 그 블록에 대한 탐색영역의 Bit-plane을 생성.
- 단계 2 : 현재 블록의 Bit-plane과 탐색영역내의 각각의 후보 블록들의 Bit-plane과의 비교.
- 단계 3 : 가장 많은 수의 Bit가 일치된 상위 m 개의 후보 블록들의 움직임 벡터를 저장.
- 단계 4 : 저장된 후보 블록들에 대한 DBD를 계산하여 최소의 DBD값을 가진 블록을 정합된 블록으로 결정.

여기서 m 값의 선택에 따라 계산량과 예측 에러의 발생에 영향을 미친다. 그림 2에서는 Bit-plane 정합의 예를 보여주고 있다.

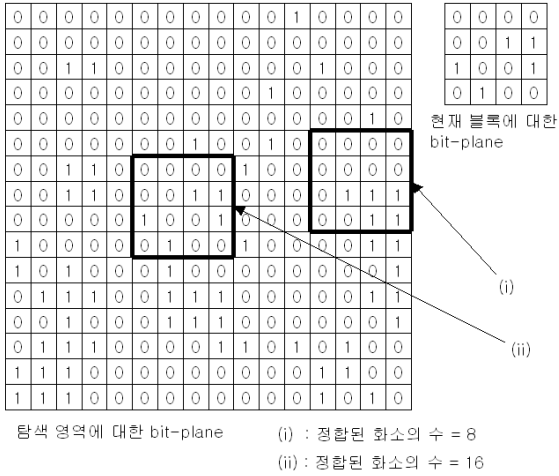


그림 2. Bit-plane 정합의 예

3. 블록내의 움직임 정보에 따른 적응적 정합

각각의 블록에 대해 일관된 움직임 예측을 하는 것은 정합의 속도면에서나 예측의 정확도 면에서나 효율적이지 못한 방식이다. 빠른 시간 내에 정확한 움직임 추정을 하기 위해서는 블록내의 움직임 정보를 이용한 적응적 예측 방식이 요구된다. 본 논문에서는 정합 과정의 계산 속도 측면의 효율성과 예측 에러 감소 측면의 효율성을 고려해 블록 내의 움직임 정보에 따라 여러 가지 적응적 탐색 기법을 적용한다.

3.1 움직임이 없는 블록 결정

식 (3)은 현재의 블록과 그 위치에 해당하는 과거 프레임에서의 블록과의 DBD를 계산하여 특정 문턱값보다 작으면 움직임이 없는 블록으로 간주하여 움직임 벡터를 (0,0)으로 주고 정합과정을 제외시킨다 [6]. 이는 움직임이 없는 블록에 대한 불필요한 정합 과정의 계산적 부담을 줄일 수 있다. 움직임이 있는 블록에 대해서는 움직임 추정을 통해서 정합된 블록과의 DBD를 계산하여 블록 분할 및 적응적 탐색이 이루어진다.

$$DBD(0,0) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |F_n(i,j) - F_{n-1}(i,j)|$$

if, $DBD(0,0) \leq TH_{no-motion}$ (3)

then, No motion searching

3.2 가변 블록 크기 움직임 추정

정합된 블록과의 DBD값이 특정 문턱값 보다 클 경우 식 (4)와 같이 블록의 분할이 발생한다. 분할된 블록의 탐색 후 DBD값이 다시 문턱값 보다 클 때는 재분할이 발생한다 ($B_{S0}, B_{S1}, B_{S2}, B_{S3}$). 이러한 분할 방식은 상당한 예측에러의 감소를 가져온다.

$$\text{if } DBD(i,j) \geq TH_{split} \text{ Splitted} \\ \text{otherwise, } \text{Unsplitted} \quad (4)$$

그림 3은 블록 내의 움직임 정보에 따라 블록이 나뉘어진 모습을 보여주고 있다.

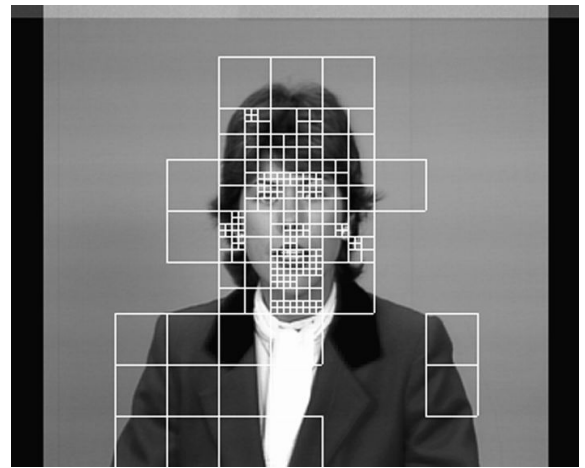


그림 3. Claire 4번 프레임에 대한 블록 분할

3.3 화소 subsampling, 탐색 영역 크기, 탐색 시작 위치 설정

화소의 subsampling 비율과 탐색 영역 설정도 블록내의 움직임 정도에 따라 설정한다. 화소 subsampling 비율은 계산량 절감에 직접적인 역할을 한다. 블록의 크기가 작으면 작아질 수록 움직임의 정도는 크기 때문에 그리고 bit 패턴의 단순화를 막기 위해서 식 (5), (6)에서와 같이 화소 subsampling 율은 낮추고 탐색영역의 크기($w_{S0}, w_{S1}, w_{S2}, w_{S3}$)는 크게 설정한다. 반대로, 움직임이 작은 큰 블록의 경우는 그 반대이다.

$$B_{S0} - 4 : 1 \text{ pixel-subsampling} \\ B_{S1} - 2 : 1 \text{ pixel-subsampling} \quad (5)$$

$$B_{S2}, B_{S3} - \text{No pixel-subsampling}$$

$$w_{S0} \leq w_{S1} \leq w_{S2} \leq w_{S3} \quad (6)$$

탐색영역의 시작위치는 가장 상위 단계의 블록인 경우 영상의 공간적 상관관계를 이용한 인접 블록들의 움직임 벡터를 이용하고, 하위 단계의 블록인 경우 상위 단계에서 얻어진 움직임 벡터를 이용한다.

4. 실험 결과 및 고찰

본 논문에서 사용한 실험 영상은 CIF(352×288)의 Claire 영상 1번부터 100번 프레임을 사용하였다. 제안된 알고리즘의 기본 블록 크기는 32×32로 하였고, 4×4 크기까지 분할이 가능하도록 하였다. 탐색 영역의 크기 w 는 블록의 분할이 진행될수록 움직임이 많다는 것을 의미하므로 작은 블록으로 갈수록 탐색영역의 크기 ($w_{S0}=1, w_{S1}=2, w_{S2}=3, w_{S3}=4$)를 증가시켰다. 따라서 가장 큰 움직임 벡터의 크기

표 2. Claire 영상에 대한 성능비교

	FSBMA	Proposed
PSNR	41.22	41.73
MSE/pixel	5.24	4.55
블럭수/Frame	396	213

는 ± 10 까지 나올 수 있도록 하였다.

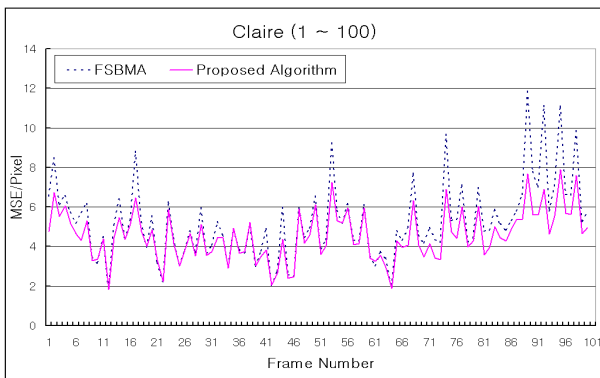


그림 4. 프레임별 평균 MSE의 비교

표 1과 그림 4는 제안 알고리즘과 블록의 크기가 16×16 , 탐색 영역의 크기가 ± 7 인 전역 탐색 알고리즘(FSBMA)과 비교를 나타낸다. Bit-plane의 도입으로 고속으로 수행할 수 있는 제안 알고리즘은 FSBMA에 비해 적은 개수의 블록을 가지고 약

0.5dB 정도의 PSNR 개선을 볼 수 있다.

5. 결론

본 논문에서는 Bit-plane, 가변 블록 크기 및 적응적 정합 방식을 이용하여 한 프레임당 적은 블록수를 가지고 적은 예측 에러를 발생시킬 수 있는 새로운 고속의 움직임 추정 방법을 제안하였다. 즉, Bit-plane을 기반으로 블록 내의 움직임 정보에 따라 적응적으로 탐색하여 효율적인 정합을 수행할 수 있다.

6. 참고문헌

- [1] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. Communication, vol. COM-29, no. 12, pp. 1799-1808, December 1981
- [2] S. Kappagantula, and K. R. Rao, "Motion compensated interframe image prediction," IEEE Trans. Communication, vol. COM-33, pp. 1011-1015, 1985
- [3] H. Gharavi, and M. Mills, "Block matching motion estimation algorithms - new results," IEEE Trans. Circuits Syst., pp. 649-651, 1990
- [4] J. Zhang, M. O. Ahmad, and M. N. S. swamy, "A New Variable Size Block Motion Compensation," Proceedings of the IEEE Int. Conf. on Image Processing, vol. 2, pp. 164-167, 1997
- [5] J. Feng, K.-T. Mehrpour, and A.E. F. Karbowski, "Adaptive block matching algorithm for video compression," IEEE Proc. Vision, Image and Signal Processing, vol. 145 no. 3, pp. 173-178, 1998
- [6] 김진한외, "복잡한 움직임 영역의 예측을 통한 가변 블록 크기 움직임 추정", 신호처리 합동학술대회 논문집, 제12권 제 1호, pp 65-68, 1999. 10. 2.