

## 적응적 탐색 영역을 이용하여 개선한 제한된 1비트 변환 알고리즘

\*장문석 \*\*정기석

한양대학교 융합전자공학부

\*lian6227@daum.net \*\* kiseokchung@gmail.com

## Improved Constrained One-Bit Transform Using Adaptive Search Range

\*Moon-Seok Jang \*\*Ki-Seok Chung

Department Of Electronic Engineering, Hanyang University

## 요약

본 논문에서 적응적 탐색 영역(Adaptive Search Range)을 이용하여 개선한 제한된 1비트 변환 알고리즘을 제안하였다. 이 변환은 전역 검색 알고리즘(Full Search Algorithm)을 사용한다. 그러나 이것은 매우 많은 연산량과 복잡도를 가진다. 제안된 알고리즘에서는 각 블록의 탐색범위를 결정하기 위한 움직임 벡터(Motion Vector)와 함께 제한된 1비트 변환 알고리즘의 제한된 마스크(Constrained Mask)를 사용한다. 실험결과를 통해 제안된 알고리즘은 움직임 예측의 정확도에 대한 성능을 비슷하게 유지하면서 평균적으로 Search Point의 수를 84% 줄일 수 있음을 보여준다.

## 1. 서론

영상 압축 시스템은 제한된 용량과 대역폭을 가진 멀티미디어 환경에서 필수적이다. 이러한 이유로 압축 시스템은 비디오 코딩 시스템에서 연속된 이미지 프레임 사이의 시간적 중복성을 줄이기 위해 움직임 예측(Motion Estimation : ME) 알고리즘을 사용한다. 가장 일반적이고 간단한 움직임 예측 방법은 블록 매칭 알고리즘(Block Matching Algorithm : BMA)이다[1-4]. BMA에서 이미지 프레임은 고정된 크기의 정사각형 블록으로 나누어진다. 현재 블록에 있어서, BMA는 탐색영역 내 이전 프레임에서 가장 유사한 블록의 위치를 찾는다. 이 후 해당 블록의 위치에 의해 현재 블록의 움직임 벡터가 결정된다. BMA의 최적의 탐색 방법은 탐색영역의 모든 지점에서 가장 적합한 블록을 찾는 전역 탐색 알고리즘(Full Search Algorithm : FSA)이다[5-7]. 그러나 이는 현재 블록과 탐색범위 내에 있는 모든 후보 블록들 사이에 SAD(Sum of Absolute Difference)를 계산하면서 매우 많은 연산량을 필요로 한다[8-11].

FSA의 문제를 해결하기 위해서 많은 고속 탐색 기법들이 제안되어 왔다[7-13]. 최근에는 이전 변환을 사용한 움직임 예측 알고리즘이 활발하게 개발되고 있다. 이러한 알고리즘은 이전 변환을 사용하기 때문에 오로지 Boolean 연산만 요구된다. 이는 하드웨어 구현을 매우 용이하게 한다. 원 영상 프레임의 픽셀값을 이전 변환하는 알고리즘 중 가장 먼저 제안된 알고리즘은 1비트 변환(One-Bit Transform : 1BT)이다. 그러나 1BT 과정이 완료된 후 원래의 화소 값의 손실로 인해 부정확한 움직임 벡터를 찾을 확률이 증가하였다. 이 점을 개선하고자 움직임 예측의 정확도를 향상시키기 위해 추가적인 비트평면을 사용하는 알고리즘이 제안되었다. 이 중에는 2비트 변환(Two-Bit Transform : 2BT)과 제한된 1비트 변환(Constrained One-Bit Transform)이 있다.

본 논문에서 우리는 적응적 탐색 영역을 이용함으로써 계산 복잡도를 줄인 CIBT를 제안한다. 이 방법은 각 블록에 대한 탐색 영역을 결정한다. 이전 비트평면을 사용하는 기존의 움직임 예측 알고리즘에 대한 설명은 2장에 설명되어 있다. 이후 제안된 알고리즘은 3장에서 소개한다. 마지막으로 실험 결과 및 결론은 4장과 5장에 있다.

## 2. 기존의 알고리즘들

## 2.1 One-Bit Transform

1BT는 원 영상 프레임을 이전 비트평면으로 변환하기 위해 17x17 영역 통과 필터 커널을 사용한다. 이 커널은 다음과 같이 표현한다.

$$K(i,j) = \begin{cases} 1/25, & \text{if } i,j \in 0,4,8,12,16 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

이전 비트평면은 필터링된 프레임의 화소값과 기존 프레임의 화소값을 비교하여 만들어진다. 이전 비트평면을 구성되는 방법은 다음과 같다.

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

여기서  $I_F(i,j)$ 는 기존 프레임인  $I(i,j)$ 를 커널 K를 이용하여 필터링된 프레임을 나타낸다. 필터링을 거친 후 최적의 움직이 벡터를 찾기 위해 SAD 대신에 Number of Non-Matching Points (NNMP)를 사용하는 FSA가 사용된다. NNMP는 다음과 같이 정의된다.

$$NNMP(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} B^t(i,j) \oplus B^{t-1}(i+m,j+n) \quad (3)$$

식 (3)에서  $B^t(i,j)$ ,  $B^{t-1}(i,j)$ 는 각각 현재 프레임의 1비트 변환된 프레임, 이전 프레임의 1비트 변환된 프레임이다. 또  $N$ 은 정합블록 크기이고  $s$ 는 탐색 범위의 크기를 결정한다.  $(m,n)$ 은 탐색 범위 내 후보 정합블록들의 변위를 나타낸다.

## 2.2 Two-Bit Transform

2BT를 사용하는 움직임 예측에서는 이전 변환된 평면을 1개만 사용하여 원 영상 프레임의 정보를 많이 잃어버리면서 생기는 1BT의 부정확한 움직임 예측을 개선하기 위하여 제안된 움직임 예측 알고리즘이다. 2BT에서는 아래의 식과 같이 국부 주변 (local surrounding) 화소들의 평균, 표준편차와 근사치 표준편차(approximate standard) 값들이 사용된다.

$$\begin{aligned} \mu &= E[I_{nw}] \\ \sigma^2 &= E[I_{nw}^2] - E^2[I_{nw}] \\ \sigma_a &= 15 + 0.0125\sigma^2 \end{aligned} \quad (4)$$

$I_{nw}$ 는 2비트 변환되는 8x8 정합블록 주변의 40x40 크기의 문턱 윈도우 (Threshold Window) 안의 화소값을 의미한다. 이를 통해 구해진 평균과 근사치 표준편차는 원 영상 프레임을 이전변환하기 위해 사용된다. 2BT를 통해 이전 변환되는 두 비트평면은 원 영상 프레임  $I$ 의 화소와 국부 문턱 윈도우 (local threshold window) 안의 화소들의 평균 및 근사치 표준편차를 비교하여 얻어진다.

$$\begin{aligned} B_1(i,j) &= \begin{cases} 1, & \text{If } I(i,j) \geq \mu \\ 0, & \text{otherwise} \end{cases} \\ B_2(i,j) &= \begin{cases} 1, & \text{if } I(i,j) \geq \mu + \sigma_a \text{ or } I(i,j) \leq \mu - \sigma_a \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (7)$$

여기서  $B_n$ 의  $n$ 은 2BT의  $n$ 번째 비트평면을 나타낸다. 2BT에서는 정합오차 기준으로  $NNMP_{2BT}$ 를 사용한다.  $NNMP_{2BT}$ 는 다음과 같이 나타낸다.

$$\begin{aligned} NNMP_{2BT}(m,n) &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} A_1 \parallel A_2, \\ A_1 &= B_1^t(i,j) \oplus B_1^{t-1}(i+m,j+n), \\ A_2 &= B_2^t(i,j) \oplus B_2^{t-1}(i+m,j+n), \\ \text{where } -s &\leq m,n \leq s-1 \end{aligned} \quad (8)$$

## 2.3 Constrained One-Bit Transform

CIBT에서는 기존의 이미지 프레임  $I(i,j)$ 과 필터링을 거친 이미지의 프레임  $I_F(i,j)$ 를 비교하여 추가적인 비트평면인 제한된 마스크 (Constrained Mask : CM)를 이용한다. 제한된 마스크는 다음과 같이

정의된다.

$$CM(i,j) = \begin{cases} 1, & \text{if } |I(i,j) - I_F(i,j)| \geq D \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

CIBT와 2BT의 주요 차이점은 추가적인 비트평면을 복잡한 계산 없이 문턱값  $D$ 와 1BT에서 사용된 필터링된 프레임을 이용하여 쉽게 구성할 수 있는 것이다. 움직임 예측을 기반으로 하는 1BT에서는 원래 화소값은 서로 비슷하지만 이전 변환 후 서로 다른 값이 되면서 움직임 예측의 정확도를 떨어뜨린다. CIBT에서는 이러한 문제점을 제한된 마스크를 통해 해결한다. CIBT에서는 최적의 움직임 벡터를 사용하기 위해 Constrained NNMP (CNNMP)를 사용하는데 식은 다음과 같다.

$$\begin{aligned} CNNMP(m,n) &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} A_3 \parallel A_4, \\ A_3 &= CM^t(i,j) \oplus CM^{t-1}(i+m,j+n), \\ A_4 &= B^t(i,j) \oplus B^{t-1}(i+m,j+n), \\ \text{where } -s &\leq m,n \leq s-1. \end{aligned} \quad (10)$$

## 3. 제안하는 알고리즘

이전 변환 후 최적의 움직임 벡터를 찾기 위해 사용되는 FSA는 탐색 영역의 모든 위치 사이에서 가장 적절한 블록을 결정하는 일반적이고 유용한 방법이다. FSA에서는 고정된 탐색 범위 내에서 모든 후보 블록의 SAD들은 계산하여 이를 최소화하는 최적의 움직임 벡터를 찾기 위해 측정된다. 따라서 FSA는 많은 계산량을 필요로 한다. 본 논문에서는 모든 블록을 포함하는 고정된 탐색 범위 대신에 각 블록의 탐색 범위를 적응적으로 결정하는 알고리즘을 제안한다. 적응적 탐색 범위는 이전 블록의 제한된 마스크와 움직임 벡터를 통해 결정된다.

$$\begin{aligned} \delta &= \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [CM^t(i,j) \oplus CM^{t-1}(i+mx,j+my)]}{N \times N} \\ MV &= \text{Max}(mx, my) \end{aligned} \quad (11)$$

여기서  $mx$ 와  $my$ 는 이전블록의 최적의 움직임 벡터를 나타낸다. 현재블록의 탐색 영역(Search Range : SR)은 다음과 같이 나타낸다.

$$SR = MV \times (1 + \delta) + \alpha \times (1 + \beta \times \delta) \quad (12)$$

$MV$ 와  $\delta$ 는 식 (11)에서 주어지고  $\alpha, \beta$ 는 실험적으로 결정하는 문턱값이다.  $\alpha = 3, \beta = 6$ 일때 움직임 예측의 정확성과 계산량이 균형을 이룬다. SR은 움직임이 없을 때와  $\delta = 0$ 일 때 (적응적 탐색 범위를 사용한 FSA가 후보블록과 원래의 블록이 같을 때) 최소값을 가진다.

	CIBT	CIBT with SR( $\alpha = 2$ )		CIBT with SR( $\alpha = 3$ )		CIBT with SR( $\alpha = 4$ )		CIBT with SR( $\alpha = 5$ )	
	PSNR	PSNR	SPDR	PSNR	SPDR	PSNR	SPDR	PSNR	SPDR
컨테이너	37.96	37.73	97.49	37.82	94.78	37.85	90.98	37.89	86.23
꽃	25.68	25.64	90.59	25.68	83.54	25.69	74.74	25.69	64.30
모바일	24.72	24.70	92.43	24.71	85.18	24.71	75.68	24.71	50.23
축구	23.22	22.99	80.65	23.13	69.87	23.18	58.11	23.22	33.27
가수	36.76	36.77	97.22	36.80	94.46	36.82	90.77	36.81	80.70
평균	29.668	29.566	91.676	29.628	85.566	29.65	78.056	29.664	62.946

표 1 적응 탐색 영역 알고리즘을 사용한( $\beta = 6$ ) 테스트에서 PSNR과 SPDR의 결과 평균

	CIBT	CIBT with SR( $\beta = 4$ )		CIBT with SR( $\beta = 5$ )		CIBT with SR( $\beta = 6$ )		CIBT with SR( $\beta = 7$ )	
	PSNR	PSNR	SPDR	PSNR	SPDR	PSNR	SPDR	PSNR	SPDR
컨테이너	37.96	37.78	95.22	37.80	95.03	37.82	94.78	37.83	94.53
꽃	25.68	25.68	86.69	25.68	85.15	25.68	83.54	25.68	81.87
모바일	24.72	24.70	88.77	24.70	86.98	24.71	85.18	24.71	83.26
축구	23.22	23.09	75.73	23.11	72.83	23.13	69.87	23.15	66.90
가수	36.76	36.80	94.87	36.80	94.68	36.80	94.46	36.81	94.25
평균	29.668	29.61	88.256	29.618	86.934	29.628	85.566	29.636	84.162

표 2 적응 탐색 영역 알고리즘을 사용한( $\alpha = 3$ ) 테스트에서 PSNR과 SPDR의 결과 평균

#### 4. 실험 결과

실험은 서로 다른 움직임 특성을 가진 5개의 CIF(352x288) sequences 로 수행 했다. 그 5개는 각각 Container(300 frames), Flower(250frames), Mobile(300frames), Football(150 frames), Singer(300 frames)이다. 블록 크기와 탐색 범위의 최대값은 각각 16 과 16이다. 원 영상 프레임은 이전과 현재 프레임 사이의 움직임 벡터와 이전 프레임의 후보 블록을 사용하여 복원한다. 움직임 예측의 성능은 PSNR로 측정된다. 계산 복잡도를 비교하기 위해 SPDR을 다음과 같이 정의한다.

$$SPDR(\%) = \left\{ 1 - \frac{\text{search point of CIBT with SR}}{\text{search point of CIBT}} \right\} \times 100 \quad (13)$$

SPDR은 제안된 알고리즘을 기반으로 한 CIBT와 기존의 CIBT에서의 Search point의 총 합 간의 비교 결과를 만든다.

표 1은  $\beta = 6$ 으로 고정시키고  $\alpha$ 를 조절하여 제안된 알고리즘의 성능을 측정한 것이고 표 2는  $\alpha = 3$ 으로 고정하고  $\beta$ 를 조절하면서 성능을 측정한 것이다. 그림 1에서 PSNR은 2-3사이에서 증가가 관찰된 것과 동일한 간격에서 SPDR에 비해 가파르게 증가한다. 2-4사이에서 SPDR은 거의 일정하게 감소한다. 하지만 SPDR과 달리 PSNR은 2-3사이에서는 급격하게 증가하고 3-4사이에서는 보다 완만하게 증가한다. 이러한 이유 때문에  $\alpha = 3$ 으로 정해졌다. 이 결과는 Search point의 수가 감소함을 보여주지만 움직임 예측의 성능은 비례하게 감소되지 않았다. 이와 같은 이유로 [그림 2]에서  $\beta = 6$ 으로 정했다.  $\alpha = 3, \beta = 6$ 으로 설정한 경우, 평균 움직임 예측의 평균 성능 손실은 0.025dB정도 이다. 이와 반대로 SPDR은 70%에서 95%로 달성되었다.

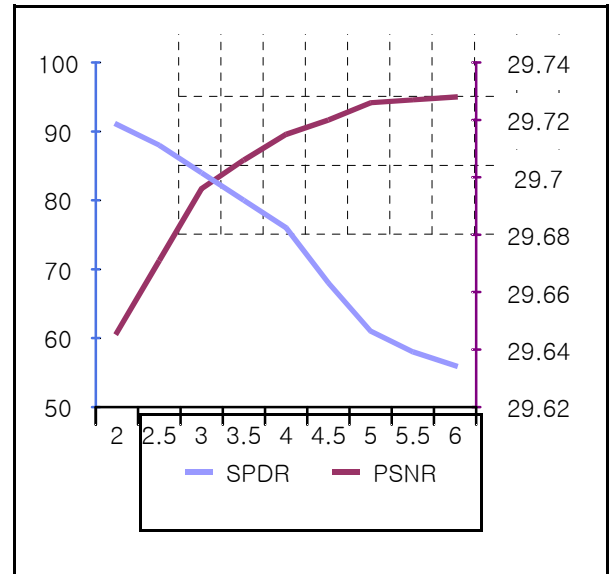


그림 1.  $\beta = 6$ 일 때 PSNR과 SPDR의 관계

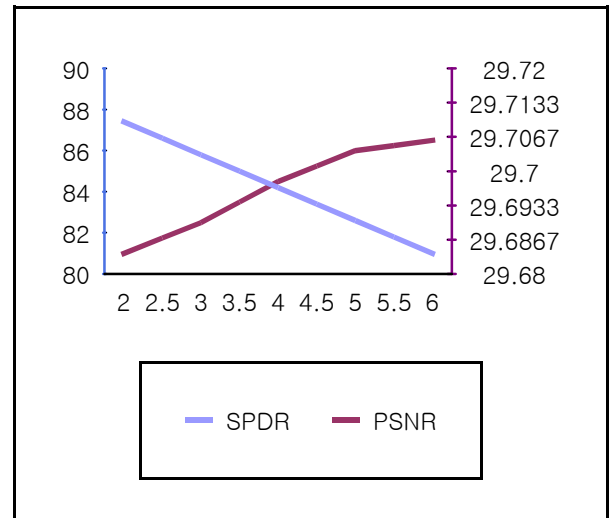


그림 2.  $\alpha = 3$ 일 때 PSNR과 SPDR의 관계

## 5. 결론

본 논문에서는 적응적 탐색 범위 알고리즘을 포함한 C1BT를 제안하였다. 각 블록의 탐색 범위를 결정하기 위해 이전 블록의 움직임 벡터와 C1BT의 제한된 마스크를 사용한다. SPDR은 계산 복잡도를 비교하기 위해 사용했다. 실험을 통해  $\alpha$ 와  $\beta$ 값을 각각 3과 6으로 결정했다. 그러면 움직임 예측의 정확도와 PSNR의 손실이 균형을 이룬다. 즉, 이 알고리즘은 움직임 추정을 기반으로 한 C1BT의 성능은 비슷하게 하면서 Search Point를 많이 감소시킬 수 있다. 또한 제안된 알고리즘은 상당한 계산량 감소에 따라 디지털 영상의 실시간 구현에 적합하다.

## 참고문헌

- [1] B. Natarajan, V. Bhaskaran, & K. Konstantinides, Low-complexity block-based motion estimation via one-bit transforms, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.7, no.5, 1997, pp.702-706.
- [2] P. H. W. Wong & O. C. Au, Modified one-bit transform for motion estimation, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.9, no.7, 1999, pp.1020-1024.
- [3] A. Erturk & S. Erturk, Two-bit transform for binary block motion estimation, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.15, no.7, 2005, pp.938-946.
- [4] O. Akbulut, O. Urhan & S. Erturk, Fast sub-pixel motion estimation by means of one-bit transform, *Lecture Notes in Computer Science*, 4263, 2006, pp.503-510.
- [5] R. Demir & S. Erturk, Block motion estimation using modified two bit transform, *Lecture Notes in Computer Science*, 4263, 2006, pp.522-531.
- [6] O. Urhan & S. Erturk, Constrained one-bit transform for low complexity block motion estimation, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.17, no.4, 2007, pp.478-482.
- [7] R. Demir & S. Erturk, Block motion estimation using adaptive modified two-bit transform, *IET Image Processing*, vol.1, no.2, 2007, pp.215-222.
- [8] H.-Y. Oh, D.-H. Kim, O. Urhan & T.-G. Chang, Modified constrained one-bit transform based fast block motion estimation, *IEEE Transactions on Consumer Electronics*, vol.5, no.3, 2007, pp.1093-1097.
- [9] O. Urhan, Constrained one-bit transform based motion estimation using predictive hexagonal pattern, *Journal of Electron. Imaging*, vol.61, no.3, 2007, Article ID:033019.
- [10] H. Lee, S. Iin & I. Jeon, Early termination scheme for binary block motion estimation, *IEEE Transactions on Consumer Electronics*, vol.53, no.4, 2007, pp.1682-1686.
- [11] H. Lee, S. Iin & I. Jeon, Early termination algorithm for 2BT block motion estimation, *Electronics Letters*, vol.45, no.8, 2009, pp.403-405.
- [12] N. I. Kim, S. Erturk & H. Y. Lee, Two-bit transform based block motion estimation using second derivatives, *IEEE Transactions on Consumer Electronics*, vol.55, no.2, 2009, pp.902-910.
- [13] O. Urhan, Constrained one-bit transform based fast block motion estimation using adaptive search range, *IEEE Transactions on Consumer Electronics*, vol.56, no.3, 2010, pp.1868-1871.