

## 적응적 탐색 범위를 이용한 2비트 변환 알고리즘

\*이상구 \*\*김용훈 \*\*\*정제창

한양대학교 전자컴퓨터통신공학과

\*esg803@hanyang.ac.kr, \*\*charismakyh@gamil.com, \*\*\*jjeong@hanyang.ac.kr

## Two-Bit Transform Algorithm using Adaptive Search Range

\*Lee, Sang-gu \*\*Kim, Yong-hoon \*\*\*Jeong, Je-chang

Department of Electronics and Computer Engineering, Hanyang University

## 요약

본 논문은 2비트 변환 (two-bit transform : 2BT) 알고리즘의 탐색 범위를 적응적으로 조정하는 알고리즘을 제안하였다. 2BT는 정합 오차 기준으로 SAD (sum of absolute differences)를 사용하지 않고 Number of Non-Matching Points (NNMP)를 사용하여 속도를 향상시키고, 하드웨어 구현을 용이하게 했다. 그러나 움직임 예측 시 기존의 방법인 전역 탐색 (FS : Full Search) 알고리즘을 사용하여 방대한 연산량을 요구한다. 이 점을 개선하기 위해 이 논문에서는 2BT의 탐색 범위를 매 블록마다 적응적으로 조정하면서 알고리즘의 계산량을 줄이고 속도를 향상시키는 알고리즘을 제안하였다. 기존의 2BT와 제안하는 알고리즘을 비교한 실험결과는 PSNR이 거의 동일하나 복잡도 측면에서 제안하는 알고리즘이 훨씬 우수한 성능을 보여준다.

## 1. 서론

제한된 용량과 대역폭을 가진 멀티미디어 환경에서 영상 압축 시스템은 필수적이다. 이를 위해 압축 시스템에서는 움직임 예측 알고리즘을 사용한다. 이 기법은 연속되는 영상 프레임 사이의 시간적 중복성을 없애 전송 데이터의 양을 획기적으로 줄여준다. 가장 보편적이고 간단한 움직임 예측 기법은 블록 매칭 알고리즘 (block matching algorithm)이다. 이는 각각의 영상 프레임을 정해진 크기의 정사각형 블록으로 분할하고 현재 프레임 내 각각의 블록과 가장 적합한 정합블록을 이전 프레임의 탐색 영역 내에서 찾아내는 방법이다. 이후 탐색된 후보 블록들 중 가장 유사한 블록의 위치로부터 움직임 벡터 (motion vector)가 결정된다. 이렇게 프레임 내 각 정합블록에 대한 움직임 벡터를 제공하는 것이 블록 매칭 알고리즘이다. 최적의 움직임 벡터를 찾는 여러 블록 매칭 알고리즘들 중에서도 가장 적합한 방법으로 알려져 있는 것은 전역 탐색 알고리즘 (full search algorithm)이다. 하지만 참조 프레임의 탐색 영역 내에서 모든 후보블록들을 탐색하여 움직임 벡터를 찾는 전역 탐색 알고리즘은 높은 복잡도와 방대한 연산량을 요구하여 실시간 동영상 부호화기의 구현에 활용하기 어려웠다[1][2].

전역 탐색 방법의 문제점을 해결하기 위하여 많은 고속 블록 정합 알고리즘들이 제안되고 있으며, 그 중 하나가 이진 블록 매칭 움직임 예측 알고리즘이다. 이 알고리즘은 영상을 이루고 있는 8비트 픽셀 (pixel)들을 낮은 비트수로 변환한 후 움직임을 예측한다. 이에 따라 최적의 움직임 벡터를 찾기 위해 불리언 (Boolean) 연산을 사용할 수 있게 되었고 하드웨어 구현 시 고속 움직임 예측이 가능하게 되었다. 이진 표현을 위한 알고리즘으로는 1비트 변환<sup>[1]</sup> (one-bit transform : 1BT) 알고리즘, 2비트 변환<sup>[2]</sup> (two-bit transform : 2BT) 알고리즘 그리고 제한된 1비트 변환<sup>[3]</sup> (constrained one-bit transform : C1BT) 알고리즘이 있다. 하지만 1BT의 경우 정합 오차가 가질 수 있는 범위가 좁아지기 때문에 정확하지 않은 움직임 벡터를 구할 확률이 증가하여 움직임 예측의 정확도가 전역 탐색 알고리즘보다 저하되는 단점을 가지고 있다. 그래서 2BT와 C1BT 등 개선된 이진 블록 매칭 움직임 예측 알고리즘들이 1BT의 정확하지 않은 정합 오차를 개선하기 위하여 제안되었다[2][3].

본 논문의 구성은 다음과 같다. 제 2장에서는 기존의 알고리즘들인 1BT, 2BT, C1BT에 대해 상세히 기술하고 제 3장에서는 본 논문에서 제안하는 탐색 범위를 적응적으로 조정하는 2BT 알고리즘에 대해 소개한다. 제 4장에서는 실험결과를 통해 기존의 알고리즘과 제안하는 알고리즘의 복잡도 및 화질을 비교하고 제 5장에서는 결론을 맺는다.

## 2. 기존의 알고리즘들

## 2.1 1비트 변환(1BT)을 사용하는 움직임 예측

1BT 움직임 예측 알고리즘은 먼저 원 영상 프레임을 식 (1)에서 정의된  $17 \times 17$  대역 통과 필터 커널  $K$ 를 적용하여 새로운 영상 프레임 을 만들어낸다.

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

이 후 필터가 적용된 프레임  $I_F$ 과 원본 영상 프레임  $I$ 의 각 화소를 비교하여 식 (2)와 같이 원본 영상 프레임을 1비트 변환한다.

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

식 (2)에서  $I(i, j)$ 는 현재 프레임 내  $(i, j)$  위치의 화소값을 나타내고  $I_F(i, j)$ 는 필터가 적용된 프레임 내  $(i, j)$  위치의 화소값을 나타낸다. 그 후 기존의 움직임 예측 기법인 전역 탐색 알고리즘을 진행하여 최적의 움직임 벡터를 찾게 된다. 여기서 움직임 벡터를 찾을 때 전역 탐색 알고리즘에서 사용하는 SAD 기법 대신에 불리언 연산자를 사용하는 NNMP (Number of Non-Matching Points)를 이용하며 식 (3)과 같이

정의 한다.

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

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

### 2.2 2비트 변환(2BT)을 사용하는 움직임 예측

2BT를 사용하는 움직임 예측은 1BT의 움직임 예측의 부정확성을 개선하기 위하여 제안된 움직임 예측 기법이다. 식 (4)와 같이 2BT에서는 국부 주변 (local surrounding) 화소들의 평균, 표준편차와 근사치(approximate) 표준편차 값들이 사용된다.

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

식 (4)에서  $I_{tw}$ 는 2비트로 변환되는 8x8 크기 변환 블록 주변의 40x40 크기의 국부 문턱 윈도우 (local threshold window) 안의 화소 값을 의미한다. 2BT에 이용되는 두 비트평면 중 첫 번째 비트 평면은 식 (5)와 같이 원 영상 프레임  $I$ 의 화소가 국부 문턱 윈도우 안의 화소들의 평균과 비교되어 얻어지며, 두 번째 비트 평면은 원 영상 프레임  $I$ 의 화소가 국부 문턱 윈도우 안의 화소들의 평균 및 표준편차와 비교되어 얻어진다.

$$\begin{aligned} B_1(i,j) &= \begin{cases} 1, & \text{If } I(i,j) \geq \mu \\ 0, & \text{otherwise} \end{cases} \\ B_2(i,j) &= \begin{cases} 1, & \text{if } I(i,j) \geq \mu + \sigma_a \text{ or } I(i,j) \leq \mu - \sigma_a \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (5)$$

$B_1(i,j)$ 는 2비트 변환된 프레임의 첫 번째 비트 평면,  $B_2(i,j)$ 는 2비트 변환된 프레임의 두 번째 비트 평면을 의미한다. 2BT에서 사용되는 오차 기준인  $NNMP_{2BT}$ 는 식 (6)과 같이 정의된다.

$$NNMP_{2BT}(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \{B_1^t(i,j) \oplus B_1^{t-1}(i+m,j+n)\} \oplus \{B_2^t(i,j) \oplus B_2^{t-1}(i+m,j+n)\} \quad -s \leq m, n \leq s-1 \quad (6)$$

식에서  $B_n^t(i,j)$ 는 2비트 변환된 현재 프레임의  $n$ 번째 비트 평면이고  $B_n^{t-1}(i,j)$ 는 2비트 변환된 이전 프레임의  $n$ 번째 비트 평면이다. 2BT 역시 1BT와 같이 전역 탐색 알고리즘보다 움직임 예측의 정확도가 떨어지나 1BT에 비해 비트 평면을 한 개 더 이용하므로 정확도가 개선되었다.

### 2.3 제한된 1비트 변환(C1BT)을 사용하는 움직임 예측

C1BT를 사용하는 움직임 예측도 1BT의 단점인 부정확한 움직임 예측을 개선하기 위해 제안된 움직임 예측 기법이다. 2BT와 같이 두 비트평면을 사용하나 평균, 표준편차와 근사치 표준편차를 이용하는 2BT와 달리 C1BT는 식 (7)과 같이 문턱 값  $D$ 를 이용한다.

$$CM(i,j) = \begin{cases} 1, & \text{if } |I(i,j) - I_F(i,j)| \geq D \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

식 (7)에서 중요한 점은 다른 정합조건을 이용하는 2BT와 달리 특별한 연산과정 없이 1BT에서 사용하는 필터를 적용한 프레임  $I_F$ 와 문턱 값  $D$ 를 이용하여 쉽게 또 다른 비트 평면인 제한된 마스크 (constraint mask)를 얻는다는 점이다. 제한된 마스크를 사용하는 C1BT는 프레임 내 두 화소값이 서로 비슷하나 1비트 변환되면서 두 화소가 각각 다른 값으로 치환되면서 생기는 움직임 예측의 부정확성을 개선한다. C1BT에서 사용되는 오차 기준인 CNNMP (constrained NNMP)는 식 (8)과 같이 정의된다.

$$CNNMP(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \{CM^t(i,j) \oplus CM^{t-1}(i+m,j+n)\} \cdot [B^t(i,j) \oplus B^{t-1}(i+m,j+n)] \quad -s \leq m, n \leq s-1 \quad (8)$$

여기서  $CM^t(i,j)$ 는 제한된 1비트 변환된 현재 프레임의 제한된 마스크,  $CM^{t-1}(i,j)$ 는 제한된 1비트 변환된 이전 프레임의 제한된 마스크를 의미한다. C1BT 역시 전역 탐색 알고리즘보다 움직임 예측의 정확도가 떨어지나 1BT의 문제점을 개선하며 1BT에 비해 정확도가 향상되었다.

### 3. 제안하는 알고리즘

이번 장에서는 2BT의 연산량을 줄이기 위해 본 논문에서 제안하는 알고리즘에 대해 기술하고자 한다. 기존의 2BT 알고리즘은 움직임 예측 시 전역 탐색 알고리즘을 사용하면서 고정된 탐색 범위 안에서 최적의 움직임 벡터를 찾는다. 하지만 이 경우 프레임 내 각 블록의 움직임 벡터를 찾을 때 매번 고정된 탐색 범위 내 모든 주변 블록들과 정합 오차를 구해 높은 복잡도와 방대한 연산량을 요구한다. 하지만 제안하는 알고리즘에서는 전역 탐색 알고리즘을 실행할 때 고정된 탐색 범위를 이용하지 않고 매 블록마다 적응적으로 가변적인 탐색 범위를 이용하여 연산량을 크게 줄인다. 가변적인 탐색 범위 SR (Search Range)을 정의하는 식은 다음과 같다.

$$SR = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [B_2^t(i,j) \oplus B_2^{t-1}(i+mx,j+my)]}{\alpha} + \beta \quad (9)$$

식 (9)에서  $B_2^t(i,j)$ 는 2비트 변환된 현재 프레임의 2번째 비트 평면이고  $B_2^{t-1}(i,j)$ 는 2비트 변환된 이전 프레임의 2번째 비트 평면이다.  $mx$ 와  $my$ 는 전역탐색 알고리즘을 통해 얻어진 최적의 움직임벡터이다. 식을 통해 구해진 SR은 다음 블록의 전역탐색 알고리즘의 탐색 범위가 된다.

### 4. 실험결과

실험은 움직임 특성이 서로 다른 6개의 CIF(352 x 288) 영상을 사용하였으며 움직임 예측을 위한 알고리즘이므로 휘도 성분만을 고려하였다. 실험조건으로 최대 탐색 범위와 움직임 블록 크기는 ±16과 16x16, 탐색 순서는 나선형(spiral) 기법을 사용하였다. 또한 제안하는 알고리즘과 기존 알고리즘의 성능 비교를 위해 화질 비교에 자주 사용되는 복원 영상의 PSNR (Peak signal-to-noise ratio) 과 연산량 비교를 위해 SPDR (search point drop rate)을 이용하였다. 적응적 탐색 범위 알고리즘을 통해 줄어든 연산량을 비교하기 위한 방법으로 탐색 범위의 총합을 비교하는 SPDR의 식은 다음과 같다.

$$SPDR = \frac{\text{search point of proposed algorithm}}{\text{search point of 2BT}} * 100 \quad (10)$$

제안하는 알고리즘에서는 PSNR이 거의 차이가 없으면서 연산량

을 크게 줄일 수 있는 문턱 값을 실험적으로 결정된 최적의 값인  $\alpha=12$ ,  $\beta=3$ 으로 결정하였다. 표 1, 2는 최적의 문턱 값을 결정하는 과정에서  $\alpha$ ,  $\beta$  중 하나의 문턱 값을 고정하고 다른 하나의 문턱 값을 변경하며 실험한 결과를 보여준다. 제한한 알고리즘의 성능 측정 결과 중 6개의 실험 영상들에 대한 평균 결과를 살펴보면  $\alpha=12$ ,  $\beta=3$ 일 때 PSNR이 0.03 dB 감소하였으나 연산량이 92% 감소하였음을 알 수 있다. 특히 상대적으로 움직임이 적은 영상에서는 연산량이 대폭 감소하면서도 PSNR이 올라가는 점을 확인할 수 있었는데 이는 2BT의 탐색범위와 움직임 예측도가 비례하지 않다는 점을 시사한다.

	2BT	2BT with SR ( $\beta=2$ )		2BT with SR ( $\beta=3$ )	
		SPDR (%)	PSNR (dB)	SPDR (%)	PSNR (dB)
akiyo	41.89	97.19	41.94	94.84	41.94
coastguard	29.9	93.05	29.62	89.69	29.75
container	37.78	97.23	37.66	94.88	37.68
flower	25.53	93.33	25.4	90.15	25.51
mobile	24.73	91.7	24.72	88.01	24.73
singer	36.26	97.24	36.21	94.91	36.27
Average	32.682	94.957	32.592	92.080	32.647

  

	2BT	2BT with SR ( $\beta=4$ )		2BT with SR ( $\beta=5$ )	
		SPDR (%)	PSNR (dB)	SPDR (%)	PSNR (dB)
akiyo	41.89	91.75	41.94	87.92	41.93
coastguard	29.9	85.57	29.82	80.72	29.86
container	37.78	91.8	37.7	87.98	37.7
flower	25.53	86.22	25.56	81.53	25.57
mobile	24.73	83.59	24.73	78.43	24.73
singer	36.26	91.84	36.28	88.03	36.28
Average	32.682	88.462	32.672	84.102	32.678

표 1. 기존 2BT와 PSNR 및 SPDR 비교( $\alpha=12$ )

	2BT	2BT with SR ( $\alpha=11$ )		2BT with SR ( $\alpha=12$ )	
		SPDR (%)	PSNR (dB)	SPDR (%)	PSNR (dB)
akiyo	41.89	94.8	41.94	94.84	41.94
coastguard	29.9	89.19	29.75	89.69	29.75
container	37.78	94.86	37.68	94.88	37.68
flower	25.53	89.65	25.51	90.15	25.51
mobile	24.73	87.3	24.73	88.01	24.73
singer	36.26	85.13	36.27	94.91	36.27
Average	32.682	90.155	32.647	92.08	32.647

	2BT	2BT with SR ( $\alpha=13$ )		2BT with SR ( $\alpha=14$ )	
		SPDR (%)	PSNR (dB)	SPDR (%)	PSNR (dB)
akiyo	41.89	94.94	41.94	94.96	41.94
coastguard	29.9	90.38	29.73	90.69	29.73
container	37.78	95.01	37.66	95.02	37.66
flower	25.53	90.78	25.5	91.1	25.5
mobile	24.73	88.89	24.73	89.36	24.73
singer	36.26	94.99	36.27	95.01	36.26
Average	32.682	92.498	32.638	92.69	32.637

표 2. 기존 2BT와 PSNR 및 SPDR 비교( $\beta=3$ )

### 5. 결론

본 논문에서는 기존의 알고리즘과 비교하여 거의 비슷한 화질을 보이지만 고정적인 탐색 범위를 적응적으로 변경하면서 움직임 예측을 고속으로 수행하는 알고리즘을 제안하였다.

실험결과에서 알 수 있듯이 제안하는 알고리즘은 기존의 알고리즘과 거의 비슷한 PSNR을 유지하면서 연산량을 크게 줄였다. 따라서 고화질 디지털 영상의 실시간 구현에 적합한 알고리즘이 될 수 있겠다.

### 6. 감사의 글

“본 연구는 지식경제부 및 산업전략기술개발사업의 연구결과로 수행 되었음”(10039886, 하드웨어 기반 변복조기 결합형 실시간 멀티코덱 트랜스코더)

### 참고문헌

[1] B. Natarajan, V. Bhaskaran, and K. Konstantinides, “Low-complexity Block-based Motion Estimation via One-Bit Transforms,” IEEE Trans. Circuits Syst. Video Technol., vol.7,no.5,pp.702-706, Aug. 1997

[2] A. Erturk and S. Erturk, “Two-Bit Transform for Binary Block Motion Estimation,” IEEE Trans. Circuits Syst. Video Technol., vol.15, no.7, pp.938-946, July 2005.

[3] O. Urhan and S. Erturk, “Constrained one-bit transform for low complexity block motion estimation,” IEEE Tans. Circuits and Syst. Video Technol., vol.17, no.4, pp.478-482, Apr. 2007.

[4] O. Urhan “Constrained one-bit transform based fast block motion estimation using adaptive search range,” IEEE Tans. Consumer Electron, vol.56, no.3, pp.1868-1871, Aug. 2010.