

## 참조 블록의 특성에 기반한 선택적 1비트 변환 알고리즘

\*박미소 \*김재훈 \*김형도 \*정제창

\*한양대학교 전자컴퓨터통신공학과

\*misosmile1128@gmail.com, \*jh.kim7963@gmail.com, \*khd1207@gmail.com,  
\*jjeong@ece.hanyang.ac.kr

## Adaptive One-Bit Transform Using Characteristic of Reference Block

\*Miso Park \*Jaehun Kim \*Hyungdo Kim \*Jechang jeong

Dept. Electronics and Computer Eng. Hanyang University

## 요약

정확한 움직임 추정 기술은 원본과 가장 유사한 영상의 복원에 효과적이고 압축률에도 중요한 영향을 미친다. 하지만 기존의 전역 탐색 (Full Search) 알고리즘과 Sum of Absolute Difference (SAD)라는 정합 오차 기준은 연산량이 높고 하드웨어 구현시 비효율적이다. 이를 보완하기 위한 1비트 변환 알고리즘은 움직임 벡터의 변화량을 0과 1의 연산으로 나타내는데, 이 알고리즘은 움직임이 많아 픽셀 값의 변화가 심한 블록의 변화량도 0과 1로만 표현한다. 그렇기 때문에 정확한 움직임이 반영되지 않고 그로 인해 낮은 Peak Signal to Noise Ratio (PSNR)을 가져온다. 이 점을 개선하고자 본 논문에서는 참조 블록들의 움직임 벡터를 파악하고 분류하여 선택적으로 움직임의 변화량이 큰 영역은 전역 탐색 알고리즘을, 움직임이 작거나 없는 영역은 1비트 연산을 수행하도록 하여 기존의 알고리즘과 비교하여 Peak Signal to Noise Ratio (PSNR) 측면에서 우수한 성능을 확인할 수 있었다.

## 1. 서론

우리가 쉽게 전송하고 감상하는 영상 정보는 각 픽셀의 값부터 움직임 벡터, 깊이 정보 등등 굉장히 방대한 양의 데이터로 구성되어 있다. 이러한 방대한 데이터를 우리가 가진 한정된 대역폭으로 전송하기에는 역부족이다. 그렇기 때문에 데이터를 압축해야하고, 압축을 위한 여러 기법 중 필수 기법인 움직임 예측이 요긴하게 쓰인다. 움직임 기법에는 화면 내 예측과 화면 간 예측이 있는데 우리가 앞으로 살펴볼 1비트 변환 기법은 화면 간 예측에 속한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 기존의 알고리즘인 1비트 변환 알고리즘에 대해, 제 3장에서는 본 논문에서 제안하는 참조 블록의 특성에 기반한 선택적 1비트 변환 알고리즘에 대해 기술하고 있다. 제 4장에서는 실험 결과를 통해 제안한 알고리즘이 기존의 알고리즘에 비해 어떤 성능 향상이 있었는지 시술하고, 제 5장에서는 결론을 맺는다.

## 2. 기존 알고리즘

움직임 추정 기술 중 전역 탐색(Full Search)알고리즘

은 가장 정확하게 움직임 벡터를 찾아낸다. 이 알고리즘은 현재 프레임과 참조 프레임을 이용하여 참조 프레임 내의 탐색 영역 내에서 가장 작은 정합오차를 갖는 블록을 찾고 이 과정에서 Sum of Absolute Difference(SAD)라는 정합 오차 기준을 이용한다. SAD는 (1)과 같이 구할 수 있다.

$$SAD(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |f^t(i,j) - f^{t-1}(i+m,j+n)| \quad (1)$$

where  $-s < m, n < s-1$

여기서  $f^t$ 는 현재 프레임,  $f^{t-1}$ 는 참조 프레임,  $(i, j)$  화소의 위치,  $(m,n)$ 은 특정 블록까지의 거리,  $s$ 는 탐색 범위(Search range)를 나타낸다.

하지만, 이러한 전역 탐색 기법은 정확한 움직임 벡터를 찾아내지만, 하드웨어로 구현하는 경우 효율적이지 못하고 제한된 대역폭을 갖고 있는 무선 통신이나 인터넷 환경에서는 연산량이 높아 실시간 전송에 적합하지 못하다. 이는, 우리에게 전역 탐색 알고리즘의 문제를 해결하기 위해 다양한 고속 움직임 예측 알고리즘을 개발할 필요성을 제시해 준다[1]-[2].

고속 움직임 예측 알고리즘에는 1비트 변환(One-Bit Transform : 1BT)[1], 2비트 변환(Two-Bit Transform : 2BT)[3][5], 그리고 제한된 1비트 변환(Constrained One-Bit Transform : C1BT)[4]이 있다. 이러한 알고리즘들은 정합 오차 기준을 SAD 대신 Number of Nonmatching Points(NNMP)나 Constrained Number of Nonmatching Points(CNNMP)를 사용하는 이진 변환 움직임 예측 알고리즘이다. 이 알고리즘들은 8비트의 화소값으로 이루어진 각각의 프레임을 이진 변환하여 0과 1로 나타내어 줌으로써, 하드웨어 구현을 용이하게 하였고 필요한 대역폭 또한 감소시켜 속도 향상이라는 결과를 가져왔다.

이진 변환 기법을 사용하는 움직임 예측 알고리즘들 중에서 1비트 변환은 원본 영상을 이진 영상으로 변환한 후 특정 필터를 씌우고, 배타적 논리합(exclusive-OR)을 사용하여 계산량을 줄이는 방법이다. 이때 사용한 필터는 (2)와 같은  $17 \times 17$  다중 대역 통과 필터 커널K로 아래와 같이 사용한다.

$$K(i,j) = \begin{cases} 1/25, & \text{if } i,j \in [0,4,8,12,16] \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

이 필터는 현재 프레임의 각 위치에 해당하는 화소 커널을 씌워 화소값과 곱하고, 그 곱한 값들의 합을 이용하여 필터링 된 프레임인  $I_f(i,j)$ 를 구한다. 이를 통해 하나의 필터링 된 영상을 만들고, 원본 프레임  $I_f(i,j)$ 과 동일한 위치에 있는 각 화소를 아래의 (3)과 같은 비교를 통해 이진 영상을 만들어 낸다.

$$B(i,j) = \begin{cases} 1, & I(i,j) \geq I_f(i,j) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

이렇게 1비트 변환된 이진영상 B는 전역 탐색과 비슷하게 정해진 탐색 범위 내에서 최적의 움직임 벡터를 찾게 된다. 전역 탐색의 경우 SAD를 사용하였지만, 1비트 변환의 경우 NNMP의 배타적 논리합을 이용하게 된다.

$$NNMP(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} B^t(i,j) \oplus B^{t-1}(i+m,j+n) \quad -s < m,n < s-1 \quad (4)$$

식 (4)에서  $B^t(i,j)$ 는 1비트 변환된 현재 프레임의 (i,j)에 해당하는 위치의 값이고,  $B^{t-1}(i,j)$ 는 1비트 변환된 참조 프레임의 (i,j)에 해당하는 위치의 값이다. 또한 N은 위의 과정을 수행하는 블록 단위이고, s는 탐색 범위를 나타낸다.

### 3. 제안하는 알고리즘

전역 탐색을 이용하여 움직임 벡터를 추출해 보면, 움직임이 거의 없거나 정지된 배경과 같은 부분의 움직임 벡터는 0에 가까운 굉장히 작은 경우가 많다. 이러한 경우에 대해서는 1비트 변환을 수행해 보아도 전역 탐색을 수행한 경우와 큰 차이가 없다. 하지만 움직임이 많거나 복잡한 블록을 탐색하는 경우, 1비트 변환은 전역 탐색에 비해 움직임 예측의 정확도가 많이 떨어지기 때문에 움직임 벡터를 잘못 찾아내서 PSNR이 상대적으로 많이 떨어지게 된다. 본 논문에서는 움직임 벡터가 잘못 찾아지는 영역에 대해 선택적으로 1비트 변환 대신 전역 탐색을 수행하기 위해 참조 프레임의 동일한 위치에 있는 블록의 움직임 벡터와 그 주변의 움직임 벡터들의 특징을 이용한다.

B	C	D
A	E	

그림 1. 움직임 벡터를 추출할 때 사용하는 참조 블록

그림 3은 현재 프레임의 특정 위치에 있는 현재 블록의 움직임 특성을 추출하기 위해 사용하는 참조 프레임의 동일 위치에 해당하는 블록과 그 주변 블록들을 보여준다. 그림 3의 회색 블록(E)는 현재 프레임과 동일 위치에 있는 참조 프레임의 블록이다. 참조 블록이 모두 존재하지 않는 프레임의 가장 왼쪽, 오른쪽, 위쪽의 블록의 경우, 움직임 벡터가 1비트 연산을 수행하여 처리하기도 충분하다고 판단하였다. 아래는 움직임 벡터를 추출하고 그 특징에 따라 1비트 연산과 전역 탐색 구간을 나누는 방법을 기술하고 있다.

움직임 벡터의 특성을 나누기 위해 참조 가능한 영역에 해당하는 블록에 대해 참조 블록 A~D의 움직임 벡터들을 추출하여 각 움직임 벡터의 x성분과 y성분들의 절대값의 평균을 구한다.

$$\begin{aligned} mv_x &= \text{average}(A_x, B_x, C_x, D_x) \\ mv_y &= \text{average}(A_y, B_y, C_y, D_y) \end{aligned} \quad (5)$$

여기서  $A_x, B_x, C_x, D_x$ 는 각 해당 블록의 움직임 벡터의 x성분들의 절대값으로 양수, 음수에 상관없이 얼마나 떨어져있는지 그 정도를 판단하기 위해 사용하였다. 이와 마찬가지로  $A_y, B_y, C_y, D_y$ 는 각 해당 블록의 움직임 벡터의 y성분의 절대값을 나타낸다. 이렇게 구한 x성분과 y성분의 평균을 움직임 벡터의 특성을 판단하기 위한 기준

값(Threshold :TH)으로 정한다.

$$TH = \text{average}(mv_x, mv_y) \quad (6)$$

위의 기준값을 이용하여 움직임 벡터의 특징을 판단하고 탐색 기법을 선택한다.

#### 4. 실험 결과

이번 실험의 탐색 범위는  $\pm 16$ 이고 움직임 추정을 위한 블록의 크기는  $16 \times 16$ , 영상은  $352 \times 288$ 의 CIF영상을 사용하였다. 또한 제안하는 알고리즘의 효과에 대해 정확히 판단하기 위해 기존의 전역 탐색 기법을 포함한 고속 움직임 예측 알고리즘들의 PSNR과 시간을 비교하였다. 표 1에서는 이번 실험에 사용하는 샘플 영상들을 보여준다.

표 1. 테스트 영상

CIF(352X288)	프레임수
tempeste	260
table	300
news	300
mobile	300
football	150
container	300
coastguard	300
children	300
bus	150
akiyo	300

표 2. PSNR[dB] 비교

	FS	1BT	MF1BT	C1BT	2BT	Proposed
tempeste	27.165	26.319	26.346	26.626	26.607	26.513
table	31.456	29.876	29.932	30.44	30.239	29.995
news	34.997	32.772	32.771	33.486	33.542	32.928
mobile	25.036	24.623	24.67	24.705	24.724	24.63
football	24.08	22.839	22.792	23.222	23.158	23.101
container	38.171	37.04	36.786	37.949	37.781	37.313
coastguard	30.432	29.888	29.88	30.023	29.896	29.891
children	29.178	28.211	28.138	28.505	28.185	28.334
bus	25.591	24.46	24.675	24.926	24.907	24.508
akiyo	42.276	41.495	41.531	41.861	41.819	41.496
Average	30.8382	29.7523	29.7521	30.1743	30.0858	29.8709

표 3. 시간[sec] 비교

	FS	1BT	MF1BT	C1BT	2BT	Proposed
tempeete	59.114	5.935	5.824	16.522	11.897	9.907
table	80.454	6.851	6.642	18.726	13.63	8.024
news	68.116	6.796	6.578	18.645	13.489	7.603
mobile	80.425	6.752	6.759	18.864	13.491	7.265
football	37.223	3.42	3.306	9.247	6.847	5.627
container	75.112	6.78	6.55	18.736	13.499	8.478
coastguard	68.187	6.785	6.633	18.423	13.614	7.021
children	54.386	6.766	6.731	18.587	13.476	7.822
bus	37.099	3.405	3.296	9.186	6.829	3.972
akiyo	31.82	3.385	3.275	9.417	6.711	3.577
Average	59.1936	5.6875	5.5594	15.6353	11.3483	6.9296

표 2는 기존 알고리즘과 제안하는 알고리즘의 PSNR을 비교한 결과이고, 표 3은 기존의 알고리즘과 제안하는 알고리즘의 시간(초)을 비교한 결과를 나타낸다. 표 3에서 볼 수 있듯이 기존의 1비트 연산과 비교하여 평균 0.1186dB의 PSNR 증가를 얻을 수 있었다. 또한 표 4에서 볼 수 있듯이 시간은 평균 1.2421sec가 증가하였다.

## 5. 결론

본 논문에서는 1비트 변환의 움직임 예측의 정확도를 개선하기 위해 참조 프레임의 참조 블록의 움직임 벡터를 기준값에 따라 분류하고 전역 탐색을 영역에 따라 선택적으로 적용하는 알고리즘을 제안하였다. 제안한 알고리즘으로 실험한 결과 기존의 1비트 변환 알고리즘과 비교해 평균 0.1186dB의 PSNR을 올릴 수 있었다.

## 6. 감사의 글

"본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학 IT연구센터 육성지원 사업의 연구결과로 수행되었음" (NIPA-2013-H0301-13-1011)

## 7. 참고문헌

- [1] 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.4, pp. 702-206, Aug. 1997.
- [2] S. Ertürk, "Multiplication-free one-bit transform for low-complexity block-based motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol.49, no. 12, pp. 3032-3044, Dec. 2001.

- [3] A.Erturk and S.Erturk, "Two-Bit Transform for Binary Block Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol.15, no.7, pp938-946, July 2005.

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

- [5] B. Demir and S.Erturk, "Block motion estimation using adaptive modified two-bit transform," *IET Image Processing.*, vol.1, no.2, pp.215-222, Jun.2007