

적응적 탐색 범위 조정을 고려한 1비트 변환 알고리즘

*김일승 **유호선 ***정제창

한양대학교 전자컴퓨터통신공학과

*ghanjang@gmail.com **hosungun@nate.com ***jjeong@ece.hanyang.ac.kr

Adaptive search range adjustment with one bit transform

*Kim Ilseung **Yoo, Ho-Sun ***Jeong, Je-Chang

Electronics and Computer Eng. Dept., Hanyang University

요약

본 논문은 1 비트 변환(1-bit transform: 1BT) 알고리즘에 적응적 탐색 지점 조정을 추가적으로 적용한 알고리즘이다. 1BT는 정합 오차 기준을 Sum of Absolute(SAD)를 대신 하여 Number of Non-matching Point(NNMP)를 사용하여 속도를 향상시키고, 하드웨어 구현을 용이하게 했다. 하지만 정합 오차 기준만 바뀌었을 뿐, 전역 탐색(FS: Full Search)알고리즘을 사용한다. 본 논문에서는 1BT방법에 적응적 탐색 지점 조정을 적용함으로써 추가적으로 속도 향상하는 알고리즘을 제안하였다. 실험 결과에서는 기존의 1BT와 제안하는 알고리즘을 비교하여 PSNR과 시간 측면에서의 뛰어난 성능을 보여준다.

1. 서론

동영상은 고효율로 압축할 수 있는 중요한 기술 중의 하나가 움직임 추정 기술이다. 움직임 추정 기술 중에서 가장 이상적이고 보편적으로 사용되는 방법은 전역 탐색(FS: Full search) 알고리즘과 Sum of Absolute Differences(SAD)이다. 하지만 이 방법은 과도하게 많은 연산을 요구하므로 이러한 문제점을 해결하기 위해 계산량을 줄이는 알고리즘들이 많이 제안되었다. 그 중 SAD 대신 다른 정합 오차 기준을 사용하여 움직임 추정기술의 계산량을 줄이면서 하드웨어 구현 시 많은 이점을 가지는 1비트 변환(One-bit Transform, 1BT) 알고리즘과 곱셈이 없는 1비트 변환(Multiplication free One-bit Transform, MF1BT) 알고리즘이 제안되었다[1], [2].

One-bit transform(1BT)은 원본영상을 이진영상으로 변환한 후 배타적 논리합 배타적 논리합(exclusive-OR)연산을 이용함으로써 계산량을 줄이는 방법이다[2]. 원본영상에 (2)의 17×17 대역 통과필터 커널을 사용해 필터가 적용된 영상을 얻는다.

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

필터가 적용된 영상을 얻은 후에 (3)의 비교 과정을 통해 이진영상을 만들어 낸다.

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

(1),(2)에서 K는 필터 커널, I는 원본영상, I_F 는 필터가 적용된 영상, B는 1비트 변환된 이진영상을 나타낸다. 이렇게 변환된 이진영상에 대해서 새로운 정합오차기준인 the Number of Non-Matching Points(NNMP)를 사용해서 움직임 벡터를 결정한다.

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

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

Multiplication Free One-bit transform(MF1BT)은 기존 1BT의 대역 통과필터 커널을 곱셈이 없는 커널로 대체하여 PSNR은 유지하면서 속도를 향상하고 하드웨어 구현을 더욱 용이하게 만들었다.

하지만 1BT와 MF1BT는 정합 오차 기준만 바뀌었을 뿐, 전역 탐색(FS: Full Search)알고리즘을 사용한다. 따라서 정합 오차를 NNMP를 사용하되 FS가 아닌 방법으로 추가적으로 계산량을 줄일 수 있다. 하지만 1BT와 MF1BT는 기존의 계산량을 줄이는 알고리즘 중에서 가장 심각한 PSNR 손실을 유발하므로 추가적으로 속도를 향상하는 방법은 더 이상의 PSNR 손실이 있으면 안 된다.

본 논문에서는 현재영상을 배경과 전경으로 나누어서 배경부분에서는 탐색 지점을 적게 할당하고, 전경부분에서는 일반적으로 탐색을 하는 적응적인 탐색 지점 조정 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 본 논문의 구성은 다음과 같다. 2장에서 제안하는 알고리즘에 대해 상세히 설명한다. 3장에서는 기존의 알고리즘과 우리가 제안하는 알고리즘의 성능을 비교하기 위한 실험결과를 보여준다. 마지막으로 4장에서 결론을 맺는다.

2. 제안하는 알고리즘

기본적으로 움직임 추정은 병진 운동 영상만을 다룬다는 가정이 있다. 배경에 대한 움직임은 카메라의 이동 속도와 상당히 밀접한 관계가 있고, 따라서 이러한 가정 하에 한 프레임 내에 있는 움직임 벡터는 항상 일정하다고 볼 수 있다. 실제로 카메라의 움직임이 병진 운동 영상이 아니라고 하더라도 배경의 움직임 벡터는 상당히 비슷할 것이고 예측 가능할 것이다. 따라서 프레임 단위로 움직임 벡터를 배경과 전경으로 나눠서 탐색 지점을 적응적으로 조정하는 것이 이 논문에 핵심이다.

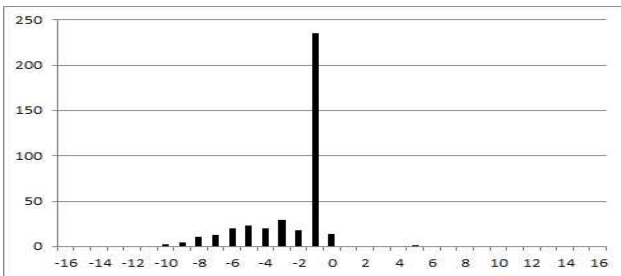


그림1. 한 프레임의 움직임 벡터에 대한 히스토그램

먼저 배경과 전경을 나누기 위해서 그림1과 같이 이전 프레임의 움직임 벡터에 대한 히스토그램을 이용하여 전역 움직임 벡터(GMV: global motion vector)를 구한다. 영상은 병진 운동(translated motion)으로만 이루어져 있고, 화면 중 배경이 차지하는 비중이 가장 크다. 이러한 가정 하에 이전 프레임의 움직임 벡터 중에서 가장 많은 비중을 차지하는 것을 GMV로 결정하는 것은 합리적이다. 이를 위해, 이전 프레임의 움직임 벡터에 대한 히스토그램을 얻고 이 중 가장 비중이 큰 부분을 GMV로 설정한다. 이를 배경의 속도로 생각하고, 나머지 부분을 전경으로 가정한다.

이렇게 분리된 배경에 해당되는 움직임 벡터는 카메라의 이동속도와 관련되므로 상당히 일정하고, 예측가능하다. 하지만 보다 정확한 예측을 위해서 그 속도에 정확도를 위한 요소인 TH1을 첨가해서 탐색 지점의 범위를 결정한다. 만약 카메라의 속도 자체가 엄청 빠른 경우에 설정된 탐색 지점의 범위가 전역 탐색 범위(Full search range)를 벗어날 가능성도 있기 때문에 범위를 0 ~ Full Search Range로 설정해준다. 결과적으로 탐색 지점의 범위를 식 (4)와 같이 설정한다.

$$D = CLIP(GMV + TH1, 0, Full\ Search\ Range) \quad (4)$$

이 때, D는 유동적인 탐색 범위를 의미하고, CLIP(x,p,q)는 x를 p와 q사이의 값으로 설정한다는 의미이다. 또한 TH1은 실험적으로 2로 설정하였다.

지금까지 전개해온 탐색 지점 조정 방법은 모두 이전 프레임의 움직임 벡터를 활용한다. 만약 이전 프레임과 현재 프레임의 움직임 벡터가 현격하게 차이가 난다면 이전 프레임의 움직임 벡터만을 가지고 예측이 불가능하다. 따라서 프레임 간 신뢰도를 확인하고 프레임 간의 관계가 예측 가능한 경우에만 적응적인 탐색 지점 조정을 적용한다.

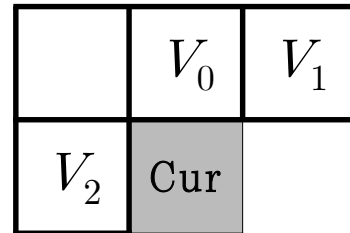


그림2. 프레임 간 신뢰도 분석에 사용되는 이웃블록

프레임 간의 신뢰도를 확인하기 위해서 현재 프레임에서 복원된 움직임 벡터와 참조 프레임에서 현재 프레임과 같은 위치에 있는 블록의 움직임 벡터의 차이를 식 (5)로 확립 함으로써 신뢰도를 확인한다. 그림2는 프레임 간 신뢰도 분석에 사용되는 이웃 블록을 보여준다.

$$R = \sum_{i=0}^2 |MV_{ref, V_i} - MV_{cur, V_i}| \quad (5)$$

만약 R 값이 TH2보다 작을 때는 프레임 간의 신뢰도가 형성된다고 보고 탐색 범위를 조정하게 되고, 클 경우에는 전역 탐색을 하게 된다. 이때, TH2는 실험적으로 3으로 설정하였다.

3. 실험 결과

제안하는 알고리즘의 성능을 살펴보기 위해 기존 알고리즘들과의 PSNR과 시간을 비교하였다. 실험에 사용된 영상은 표1에 나타낸다.

표 1. 테스트 영상

CIF(352×288)	프레임 수
Akiyo	300
Coastguard	300
Container	300
Foreman	300
Hall-monitor	300
Mothre & daughter	300

탐색범위는 ± 16 , 움직임 추정 블록크기는 16×16 으로 설정하였다.

표 2. 기존 알고리즘과의 상대적인 PSNR[Δ dB] 비교

	1BT	MF-1BT
Akiyo	0.11	0.07
Coastguard	0.05	0.03
Container	0.12	0.12
Foreman	0.03	0.02
Hall-monitor	0.57	0.54
Mothre & daughter	0.11	0.16
평균	0.16	0.16

표 2는 기존의 알고리즘과 제안하는 알고리즘의 PSNR [Δ dB]를 나타낸 결과이다. 이때 Δ dB는 제안하는 알고리즘의 PSNR과 기존 알고리즘의 PSNR의 차이를 의미한다. 표 2에서 볼 수 있듯이 제안하는 알고리즘은 1BT와 MF1BT에서 모두 0.16dB 정도의 PSNR 이득이 있음을 알 수 있다.

표 3. 기존 알고리즘과의 상대적인 시간 비율[Δ 배]

	1BT	MF-1BT
Akiyo	5.46	3.52
Coastguard	2.02	2.02
Container	3.99	4.03
Foreman	1.44	1.44
Hall-monitor	2.01	2.04
Mothre & daughter	1.73	1.66
평균	2.78	2.45

표 3은 기존의 알고리즘과 제안하는 알고리즘의 상대적인 시간을 나타낸 결과이다. 이때 Δ 배는 기존의 알고리즘에 비해 제안하는 알고리즘이 몇 배나 빨라졌는지를 의미한다. 표 3에서 보듯이 제안하는 알고리즘이 기존의 알고리즘에 비해 평균적으로 2.5배 정도 빠른 것을 알

수 있다.

4. 결론

1BT와 MF1BT는 NNMP라는 SAD와는 다른 정합 오차 기준을 이용하여 속도를 향상시키고, 하드웨어 구현을 용이하게 만들었다. 그러나 정합 오차 기준만 바뀌었을 뿐 지속적으로 FS를 이용하기 때문에 추가적으로 계산량을 줄일 여지가 있었다. 본 논문에서는 영상을 배경과 전경으로 나누어서 적응적으로 탐색 범위를 조정함으로써 PSNR을 유지하거나 향상시키면서 추가적으로 계산량을 줄이는 알고리즘을 제안하였다.

배경과 전경을 나누기 위해서는 참조 프레임의 움직임 벡터들의 히스토그램을 이용하여 GMV를 구했고, 이를 배경의 속도로 가정하였다. 이를 이용하여 배경에 해당하는 부분의 속도를 예측하여 적응적으로 탐색 범위를 조정하였다. 하지만 탐색 범위를 조정하는 데 참조 프레임의 움직임 벡터 정보만을 이용했으므로 신뢰도를 확인하여 탐색 범위에 신뢰성을 부여했다.

믿을만한 구역에서 탐색 범위를 적응적으로 줄임으로써 실험결과에서 보이듯이 PSNR을 기존의 알고리즘보다 0.16dB 정도 올라갔고, 시간도 평균적으로 2.5배 정도 올라가는 것을 확인할 수 있었다.

5. 감사의 글

"본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 육성지원 사업의 연구결과로 수행되었음" (NIPA-2011-C1090-1100-0010)

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.