

단일 문턱 값을 갖는 이진 블록 정합 움직임 예측

*정동진 **정제창

한양대학교 전자컴퓨터통신공학과

dlying@hanyang.ac.kr

Unithreshold Two-Bit Block Motion Estimation

*Jung, Dongjin **Jeong, Jechang

Electronics and Computer Eng. Dept., Hanyang University

요약

영상 압축은 멀티미디어 실시간 전송에 있어서 핵심적인 기술이다. 동영상 압축 기술 중 움직임 예측 부분은 가장 계산량이 많고 가장 복잡한 부분으로 실시간 전송을 위해서는 고속 알고리즘이 필요한 부분이다. 본 논문은 기존의 움직임 예측 알고리즘의 하나인 이진 블록 정합 움직임 예측 알고리즘 중 2bit 변환을 개선하여 더욱 빠른 알고리즘을 제안한다. 기존의 2bit 변환 알고리즘은 현재 프레임과 이전 프레임의 블록에 윈도우를 씌어 각 프레임의 윈도우에서 구한 평균과 표준편차를 가지고 각각 2bit 변환을 하였다. 그러나 본 논문은 현재 프레임과 이전 프레임의 블록에 윈도우를 씌우고 현재 프레임의 윈도우에서 구한 평균과 표준편차를 이전 프레임에 적용을 시켜 같은 평균과 표준편차를 이용하여 기존의 알고리즘을 개선한다. 제안하는 알고리즘은 계산량의 감소와 동시에 화질을 유지시킨다.

1. 서론

디지털 비디오 압축은 HDTV 나 SDTV 와 같은 방송장비를 포함해서 넓은 범위의 응용제품에서 비디오 데이터를 저장하거나 전송할 때에 대역폭 감소측면에서 매우 필수적인 부분이며 이를 효과적으로 줄이기 위해 MPEG-x,H.264와 같은 여러 동영상 압축 표준들이 제정되어 왔다.

움직임 예측과 보상은 비디오 압축 알고리즘에서 매우 중요한 역할을 하고 있다. 특히 움직임 예측은 전체 계산량에서 50%정도를 차지할 정도로 매우 중요한 부분이다.

움직임 예측에서 가장 널리 사용되는 알고리즘은 전역탐색 (full search)알고리즘이다. 전역 탐색 알고리즘은 주어진 탐색 영역 내의 모든 후보지점에서 최소의 정합 오차를 가지는 위치를 찾는 방법이다. 이 방법은 모든 범위를 탐색하기 때문에 정확한 최소 정합 오차 위치를 찾아 낼 수 있고, 하드웨어로 구현이 간편하다는 장점을 가지지만, 계산량이 막대하기 때문에 비디오의 실시간 부호화가 어렵다는 단점이 있다. 이러한 전역 탐색 알고리즘의 문제점을 해결하기 위해서 여러 알고리즘들이 연구 되어 왔다.

움직임 예측 알고리즘은 크게 세 종류로 분류 될 수 있다. 첫 번째로는 고속 전역 탐색 알고리즘인 Successive elimination algorithm (SEA) 계열이다.[1] 이 계열의 알고리즘은 탐색 영역 내의 모든 후보

지점의 정합 오차를 손실 없이 계산하기 때문에 무손실 알고리즘이다. 두 번째는 Partial distortion search, (PDS), Partial distortion elimination (PDE)계열이다.[2-3] 위 계열은 탐색 영역 내의 후보블록에서 후보 블록 내 일부분의 샘플을 채취해 비교하는 알고리즘으로 조기 중지 알고리즘이다. 세 번째는 정합 오차 기준을 바꾸는 알고리즘이다.[4-9] 이러한 알고리즘은 프로세서가 참조 프레임과 현재프레임 간의 정합 오차를 구할 때 병렬처리가 가능하게 한다. 본 논문에서는 [4]의 알고리즘의 계산 복잡도를 더욱 줄이는 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 알고리즘과 제안하는 알고리즘에 대해서 상세히 설명하고, 3장에서는 우리가 제안하는 알고리즘과 기존의 알고리즘(FS, 2BT)의 성능을 비교하기 위한 실험 결과를 보여준다. 마지막으로 4장에서는 결론을 맺는다.

2. 제안하는 알고리즘

2.1 전역 탐색 알고리즘

전역 탐색 알고리즘은 가장 흔히 쓰이는 움직임 예측방법이다. 식 (1)을 사용하여 최소정합블록을 찾는데, 현재 프레임의 블록과 이전프레임의 후보 블록에 화소의 차이를 모든 합한 값 중 최소가 되는 후보블록을 움직임 벡터로 설정하는 알고리즘이다. 이 알고리즘은 현

재 블록의 화소와 후보블록의 화소의 차이를 모두 계산하기 때문에 정확한 움직임 벡터를 구할 수 있지만, 계산량이 크다는 단점이 있다.

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

여기서 f 는 화소의 세기를, (i,j) 는 화소의 위치를, (m,n) 은 후보 블록까지의 거리를, s 는 탐색 범위를 나타낸다. 최종적으로 선택되는 움직임 벡터는 최소 SAD를 가지는 후보 블록까지의 거리 (m,n) 가 된다.

2.2 이진 블록 정합 움직임 예측 알고리즘

이진 블록 정합 움직임 예측은 원래의 8bit로 표현된 영상을 적은 수의 bit로 표현하고 처리함으로써 계산 복잡도를 줄이기 위하여 개발되었다. 여러 비트를 한 번에 처리함으로써 계산 부담을 줄이는 효과가 있다. 그러나 8bit를 몇 개의 bit로 나타냄으로써 나타나는 부작용으로 정합오차의 값의 범위가 줄어들기 때문에 움직임 벡터가 부정확해지는 것을 감수해야한다. 이진 블록 정합 움직임 예측 알고리즘에는 크게 두 종류의 알고리즘이 있다. 그 중 첫 번째인 1bit 변환은 다음과 같은 17x17 크기의 대역 통과 필터커널을 필요로 한다[5].

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

현재 프레임과 이전 프레임의 오차를 구하기 이전에 두 프레임은 식(2)에 의해 필터링 되고 식(3)을 이용해 이진 영상을 만든다.

$$B(i,j) = \begin{cases} 1, & \text{if } F(i,j) \geq \hat{F}(i,j) \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

여기서 F 는 원본 프레임, \hat{F} 은 필터 K 가 적용된 후의 프레임을 나타낸다. 이처럼 한 프레임을 1bit의 이진표현으로 바꾸는 과정을 1bit 변환이라고 한다. 1bit 변환 후, 두 이진 프레임 간의 정합 오차는 NNMP (Number of Non-Matching Points)를 이용하여 구할 수 있다.

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

여기서 \oplus 는 배타적 논리합 (exclusive OR) 연산을 의미한다. 최종 움직임 벡터는 최소의 NNMP를 가지는 후보 블록까지의 벡터 (m,n) 가 된다. 1bit 변환은 8x8블록인 경우 NNMP가 최대 64까지 밖에 갖지 못하기 때문에 부정확한 움직임 벡터를 찾는 경향이 있을 수 있다. 이를 보완하기 위해서 2bit 변환이 개발되었다.

2bit 변환은 block-by-block로 각 8x8블록에 40x40 window를 씌우고 window내에서 평균과 표준편차를 구한다.

$$\begin{aligned} \mu &= E[I_{tw}] \\ \sigma^2 &= E[I_{tw}^2] - E^2[I_{tw}]. \end{aligned} \quad (5)$$

여기서 I_{tw} 는 window내에서 화소의 세기를 나타낸다. 그러나 표준편차를 구하는 계산량이 크기 때문에 다음과 같은 근사식을 이용한다.

$$\sigma_a = 15 + 0.0125\sigma^2 \quad (6)$$

여기서 σ_a 는 표준편차의 근사 값을 나타낸다. 각 windows에서 구한 표준편차와 평균을 가지고 식(7)을 이용해서 이진 변환을 한다.

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

여기서 $B_1(i,j)$ 와 $B_2(i,j)$ 는 변환된 2bit 평면을 나타낸다. 두 프레임간의 오차는 식(8)을 이용해 구할 수 있다.

$$NNMP(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} B_1^t(i,j) \oplus B_1^{t-1}(i+m,j+n) \oplus B_2^t(i,j) \oplus B_2^{t-1}(i+m,j+n), \quad -s \leq m,n \leq s-1 \quad (8)$$

2.3 단일 문턱 값을 갖는 이진 블록 정합 움직임 예측

2bit 이진 블록 정합 움직임 예측 알고리즘은 1bit 이진 블록보다 정합 오차를 가지는 범위가 늘어나서 보다 정확한 움직임 벡터를 구할 수 있다. 그러나 평균과 표준편차를 구하면서 생기는 계산 복잡도는 무시 할 수 없을 정도이다. 위 알고리즘을 통하여 계산 복잡도를 줄여 보다 효율적인 코딩이 가능 할 수 있다.

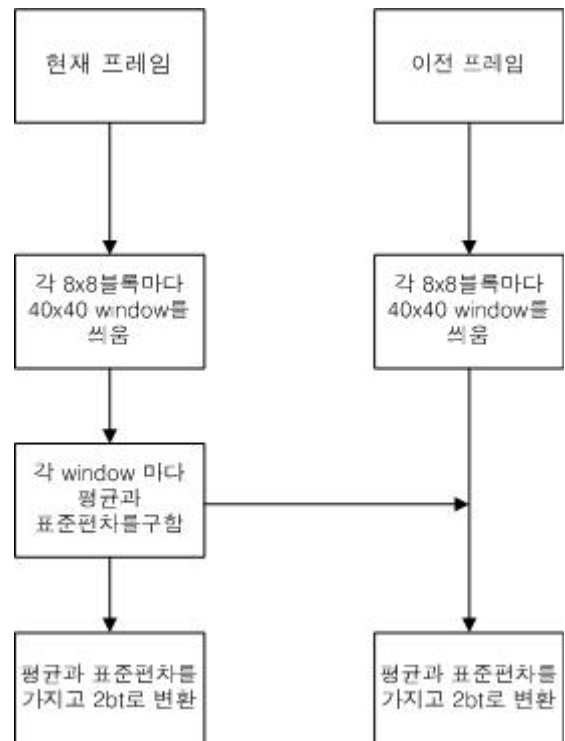


그림 1. 단일 문턱 값을 갖는 이진 블록 정합 움직임 예측 순서도

제안하는 알고리즘의 순서는 다음과 같다. 2bit 변환과 같이 현재 프레임에서 각 8x8 블록에 40x40 windows를 씌우고 평균과 표준편차를 구한다. 현재 프레임에서 구한 각 블록의 평균과 표준편차를 가지고 이전 프레임에 같은 평균과 표준편차를 적용하여서 2bit변환을 한다.

3. 실험 결과 및 분석

위 실험에서는 성능 평가를 위해서 CIF(352x288)를 가진 시퀀스들을 이용하였다. 탐색 범위는 8로 설정 하였고 각각 299프레임에 대해 실험을 하였다.

표 1 PSNR(dB) 성능 비교

PSNR	FS	2BT	Proposed
COASTGUARD	31.70	30.5	30.51
FOREMAN	33.07	30.31	30.32
MOBILE	25.87	24.96	24.98
STEFAN	25.48	24.33	24.33
TABLE	34.08	32.48	32.55
CONTAINER	38.43	37.72	37.76
HALL	36.01	34.02	34.09

표 1을 보면 제안 하는 알고리즘은 기존의 2bit 알고리즘과 비교해서 약간의 PSNR이득이 있거나 같을 것을 확인 할 수 있다.

표 2는 각 알고리즘을 실행하는 데에 필요한 계산 복잡도를 나타낸 표이다. 표 2를 보면 2bit 알고리즘에 비해서 제안하는 알고리즘은 덧셈이 약13% 감소 했고, 곱셈은 50% 감소 했으며, 비교는 10% 감소 했다. 이 결과는 2bit 알고리즘이 현재 프레임과 이전프레임에 대해서 각각 평균과 표준편차를 구하는 것과 달리, 제안하는 알고리즘은 현재 프레임의 평균과 표준편차만 구하기 때문에 계산량 이득이 있었다.

표 2 매크로 블록당 계산 복잡도

	FS	2BT	Proposed
Add	36992	24900	21698
Multiplication	0	3208	1604
Comparision	289	673	609
Abs	18496	0	0
Bit_operation	0	55616	55616

4. 결론 및 향후 연구 방향

본 논문에서는 이진 블록 정합 움직임 예측에서 보다 적은 계산량을 쓰는 알고리즘을 제안하고 실험을 통하여 이를 검증하였다. 이 실험을 통해 제안하는 알고리즘이 비슷한 화질을 유지하거나 보다 나은 화질을 얻으면서도 낮은 계산 복잡도를 가지는 것을 확인 하였다.

향후 연구 방향은 제안하는 알고리즘이 화질이 2bit변환과 비교해서 더 좋은 장면이 있는데 이것을 분석하여 화질 개선 알고리즘을 개발할

계획이다.

5. 참고 문헌

- [1] Y. Wang and G. Tu, "Successive elimination algorithm for binary block matching motion estimation," *Electronics Letters*, vol. 36, pp. 2007-2008, Nov. 2003.
- [2] Chok-kwan Cheung and Lai-man Po, "Normalized partial distortion search algorithm for block motion estimation," *IEEE Trans. CSVT*, vol. 10, pp. 417-422, Apr. 2000.
- [3] Chun-ho Cheung and Lai-man Po, "Adjustable partial distortion search algorithm for fast block motion estimation," *IEEE Trans. CSVT*, vol. 13, pp. 100-110, Jan. 2003.
- [4] A. Erturk and S.Erturk, "Two-Bit Transform for Binary Block Motion Estimation," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 15, no. 7, pp. 938-946, July. 2005.
- [5] B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low-Complexity Block-based Motion Estimation via One-Bit Transforms," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 7, no. 5, pp. 702-706, Aug. 1997.
- [6] Sarp Erturk, "Multiplication-Free One-Bit Transform for Low-Complexity Block-Based Motion Estimation," *IEEE Signal Processing Letters*, vol. 14, no. 2, 109-112, Feb. 2007.
- [7] B. Demir and S. Erturk, "Block motion estimation using modified two-bit transform," A. Levi et al. (Eds.) *ISCIS2006*, LNCS, pp. 522-531, 2006.
- [8] H. Lee, S. Jin, and J. Jeong, "Early termination scheme for 2BT block motion estimation", *Electronics Letters*, vol. 45, Apr. 2009.
- [9] O. Urhan and Sarp Erturk, "Constrained one-bit transform for low complexity block motion estimation," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 17, pp. 478-482, Apr. 2007.