

매크로블록 단위로 거리값을 선택해주는 제한된 1비트 변환 알고리즘

*김재훈 **김일승 ***윙텍셱 ****정제창

한양대학교 전자컴퓨터통신공학과

*hard_think@naver.com, **ghanjang@naver.com,

justsheng@yahoo.com *jjeong@ece.hanyang.ac.kr

Adaptive Distance Selection Algorithm

Using Constrained One-Bit Transform in Macroblock Unit

*Jaehun Kim **Ilseung Kim ***Ng Teck Sheng ****Jechang jeong

Dept. Electronics and Computer Eng. Hanyang University

요약

동영상을 고효율로 압축할 수 있는 중요한 기술 중의 하나가 움직임 추정 기술이다. 움직임 추정 기술에서 가장 널리 사용되는 탐색 방식과 정합오차를 판단하는 기준은 각각 전역 탐색(FS: Full Search)알고리즘과 Sum of Absolute Differences(SAD)이다. 하지만, FS와 SAD를 사용한 움직임 추정기술은 인코더 전체 계산량의 50%이상을 차지하게 된다. 무선 통신 및 인터넷 환경에서 사용되는 응용프로그램에 대해서는 많은 계산량이 실시간 전송을 어렵게 만든다. 이러한 이유로, 동영상 압축 기술에서 계산량을 줄이는 방법이 중요하게 되었다. 그 결과로, SAD대신 다른 정합 오차 기준을 사용하여 움직임 추정기술의 계산량을 줄이는 알고리즘들이 제안되었다. 본 논문에서는 계산량을 줄이면서 하드웨어 구현 시 많은 이점을 가진 1비트 변환 (One-bit Transform, 1BT)알고리즘과 1비트 변환의 성능을 향상시킨 제한된 1비트 변환 (Constrained one-bit transform, C-1BT)을 소개하고 C-1BT방법에 있어서 기존 방식의 빠른 인코딩속도를 유지하면서 PSNR을 증가시키는 알고리즘을 제안한다. 제안하는 알고리즘에서는 기존 C-1BT에서 고정해서 사용한 파라미터 D값을 대신에 현재블록과 동일한 위치에 있는 참조영역의 블록과 그 주변블록의 움직임벡터 정보를 이용해서 블록의 움직임 정도를 판단하고, 블록단위로 D값을 적응적으로 결정한다. 실험결과는 제안하는 알고리즘이 기존의 알고리즘과 비교하여 평균 0.11dB PSNR 증가를 보여 준다.

1. 서론

동영상을 고효율로 압축할 수 있는 중요한 기술 중의 하나가 움직임 추정 기술이다. 움직임 추정 기술은 이웃한 프레임들 간의 유사성을 이용해서 시간 축 상에 존재하는 데이터의 중복성을 제거함으로써 압축 효율을 높인다. 움직임 추정 기술에서 가장 널리 사용되는 탐색 방식과 정합오차를 판단하는 기준은 각각 전역 탐색(FS: Full Search)알고리즘과 Sum of Absolute Differences(SAD)이다. 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 현재프레임, f^{t-1} 는 참조프레임, (i,j) 는 화소의 위치, (m,n) 은 후보 블록까지의 거리, s 는 탐색 범위

를 나타낸다.

FS와 SAD를 사용한 움직임 추정기술은 인코더 전체 계산량의 50%이상을 차지하게 된다[1]. 이러한 문제점은 오프라인 환경에서 이루어지는 응용프로그램에서는 문제가 되지 않는다. 하지만, 무선 통신 및 인터넷 환경에서 사용되는 응용프로그램에 대해서는 많은 계산량이 실시간 전송을 어렵게 만든다. 이러한 이유로, 동영상 압축 기술에서 계산량을 줄이는 방법이 중요하게 되었다. 그 결과로, SAD대신 다른 정합 오차 기준을 사용하여 움직임 추정기술의 계산량을 줄이는 알고리즘들이 제안되었다[2]-[4].

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

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

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

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

(2),(3)에서 K는 필터 커널, I는 원본영상, I_F는 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.

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

Constrained one-bit transform(C-1BT)는 1BT의 성능저하 요인을 제거하기 위해 새로운 정합오차 기준인 Constrained NNMP(CNNMP)를 사용한다[3]. 그림1에서 b와 c 두 화소들과 같이 실제로 유사한 값을 가지는 화소들이 1BT에 의해서 문턱값(T)을 경계로 서로 다른 이진 영역으로 분류 된다. 이러한 화소들이 NNMP를 계산하는 과정에서 유사한 값을 가짐에도 불구하고 서로 정합하지 않는다고 판단되고, 이는 움직임 추정 성능을 저하시키는 요인 중에 하나다.

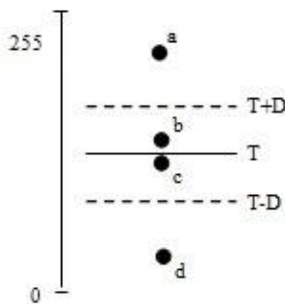


그림1. 1BT 문턱값과 거리에 영향을 받는 화소값

위에서 언급한 성능 저하요인을 제거하기 위해 거리(D) 파라미터를 추가하여 그림 1에서 T를 경계로 T-D와 T+D사이의 영역에 존재하는 화소들에 대해서는 NNMP에 의한 정합 판단기준에서 제외시킨다. D는 10으로 고정된 값을 사용하고, T-D와 T+D사이의 영역에 존재하는 화소를 판단하기 위한 새로운 이진 영상(CM)을 (4)에 의해 추가한다.

$$CM(i,j) = \begin{cases} 1, & \text{if } |I(i,j) - I_F(i,j)| \geq D \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

1BT의 이진영상과 CM영상에 대해 CNNMP를 사용해서 움직임 벡터를 결정한다.

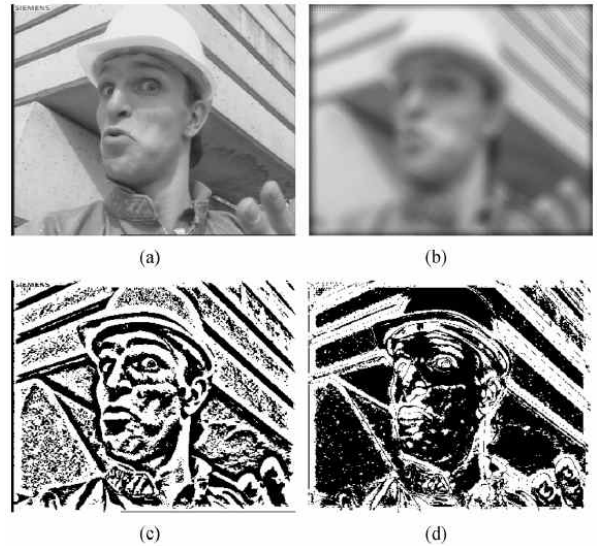


그림2. 1BT와 CM영상 생성의 예

- (a) 원래의 영상 (b) 1비트 필터링을 거친 영상
- (c) 1비트 변환후의 영상 (d) CM영상

$$CNNMP(i,j) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [CM^t(i,j) \parallel CM^{t-1}(i+m,j+n)] \odot [B^t(i,j) \oplus B^{t-1}(i+m,j+n)] \quad -s < m, n < s-1 \quad (5)$$

여기서 CM^t과 CM^{t-1}은 각각 현재영상에 대한 CM과 참조영상에 대한 CM을 나타내고 \parallel 는 논리합(or)연산, \odot 는 논리곱(and)연산을 나타낸다. 그림2는 C-1BT에 사용되는 이진영상의 생성 예를 보여준다. C-1BT에서 D값에 따라 PSNR이 영향을 크게 받는다. 움직임이 많거나 복잡한(high-motion)영상은 D값이 클수록 PSNR이 증가하고 반면에, 움직임 정도가 보통(medium-motion)인 영상은 D값이 작을수록 PSNR이 증가한다[4]. 우리는 D에 따른 PSNR의 민감한 변화로 인해 성능이 저하되는 요인을 보완하기 위해 10으로 고정해서 사용했던 D값을 참조영상의 움직임 특성을 분석해서 프레임 단위로 적응적으로 D값을 변경해주는 방법을 제안한 바 있다[6].

본 논문에서는 C-1BT에서 현재영상의 기준블록과 동일위치에 있는 참조 영상의 블록과 그 이웃블록의 움직임 벡터를 분석해서 매크로블록(MB)단위로 적응적으로 D값을 선택하는 알고리즘을 제안한다.

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

C-1BT에서 영상의 움직임정도에 따라 최적의 D값이 다르다는 것을 확인할 수 있다. 움직임 추정을 수행해야

하는 현재블록에 대해서 참조영상의 동일한 위치에 있는 블록의 움직임벡터와 그 주변의 움직임벡터 정보를 이용하여 주변의 움직임특성을 고려하게 되고, 이에 따라 블록 단위로 D값을 적응적으로 결정한다.

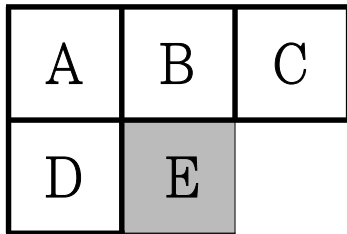


그림3. 움직임 특성 분석에 사용되는 이웃블록

그림3은 움직임 특성을 분석하기 위해 참조영상에서 사용되는 이웃블록을 보여준다. 음영이 있는 블록은 현재블록과 동일한 위치에 있는 참조영상의 블록을 나타낸다. 프레임의 가장자리에 위치한 매크로블록의 경우에는 그림2의 이웃블록을 모두 사용하지 못하는 경우가 생기고, 이러한 경우에는 사용할 수 있는 이웃블록만의 움직임 벡터를 가지고 움직임 특성을 판단한다. 움직임 특성을 판단하는 과정은 다음 단계를 따른다.

단계 1: 사용가능한 이웃블록의 움직임 벡터에 대해서 x 방향, y방향의 평균을 구한다. (6)은 모든 이웃블록이 이용 가능한 경우를 나타낸다.

$$mv_x = average(A_x, B_x, C_x, D_x, E_x)$$

$$mv_y = average(A_y, B_y, C_y, D_y, E_y) \tag{6}$$

여기서 A_x, \dots, E_x 는 각각 블록의 x방향 움직임벡터의 절대값을 나타내고, A_y, \dots, E_y 는 각각 블록의 y방향 움직임 벡터의 절대값을 나타낸다.

단계 2: 움직임 특성을 판단하기 위한 기준값(Th)을 결정한다. (7)로부터 구해진다.

$$Th = Max(mv_x, mv_y) \tag{7}$$

단계 3: Th 값으로부터 움직임 특성을 판단하고 D값을 결정한다. 본 논문에서 사용한 조건은 표1에 나타낸다.

표 1. 기준값에 따른 움직임특성과 D값

기준값 조건	움직임 특성 정보	D값
$Th = 0$	정지	6
$Th = 1$	느린 움직임	9
$Th = 2$	보통 움직임	10
$Th = 3$	빠른 움직임	11
$Th \geq 4$	매우 빠른 움직임	12

3. 실험 결과

제안하는 알고리즘의 성능을 살펴보기 위해 기존 알고리즘들과의 PSNR과 시간을 비교하였다. 실험에 사용된 영상은 표2에 나타낸다.

표 2. 테스트 영상

CIF (352×288)	프레임 수
container	300
dancer	250
flower	250
foreman	300
football	150
stefan	300

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

표 3. PSNR[dB] 비교

	FS	1BT	MF-1BT	C-1BT	Proposed
container	38.15	37.01	36.79	37.96	37.98
Dancer	30.70	28.26	28.03	28.49	28.82
Flower	25.88	25.63	25.59	25.63	25.69
foreman	31.67	30.10	30.03	30.38	30.43
Football	24.03	22.83	22.79	23.20	23.27
Stefan	23.86	22.94	22.86	23.08	23.17
평균	29.05	27.80	27.68	28.12	28.23

표 3은 기존의 알고리즘과 제안하는 알고리즘의 PSNR[dB]을 비교한 결과를 나타낸다. 표 3에서 볼 수 있듯이 제안하는 알고리즘은 D값을 고정된 C-1BT에 비해 평균 0.11dB PSNR증가를 보인다.

표 4. 시간[sec] 비교

	FS	1BT	MF-1BT	C-1BT	Proposed
container	150.73	16.10	15.75	25.57	25.76
Dancer	111.77	13.42	13.17	20.95	21.23
Flower	126.39	13.38	12.97	21.16	21.66
foreman	129.67	17.12	15.63	24.32	24.60
Football	72.46	8.06	7.91	12.64	12.97
Stefan	137.50	16.07	15.57	24.01	24.91

표 4는 기존의 알고리즘과 제안하는 알고리즘의 시간 [sec]을 비교한 결과를 나타낸다. 제안하는 방법은 기존의 C-1BT보다 평균 0.4초 정도의 시간 증가를 보인다. 이는 참조영역의 이웃블록을 이용해 움직임 특성을 판단하는 부분이 추가되었기 때문이지만 기존의 빠른 인코딩 속도를 유지한다.

4. 결론

C-1BT가 영상별로 D값에 따라 PSNR의 변화가 민감하다는 문제점이 있었다. 본 논문에서는 이런 문제점을 해결하기 위해 고정된 D값을 사용하지 않고 현재블록과 동일한 위치에 있는 참조영상의 블록과 그 주변블록의 움직임벡터 정보를 이용해서 블록단위로 적응적으로 D값을 결정하는 알고리즘을 제안하였다. 주변블록의 움직임벡터의 x방향과 y방향에 대한 평균값을 가지고 움직임 특성을 판단하는 기준값을 결정한다. 결정된 기준값에 따라 움직임특성에 맞는 D값을 결정한다. 기존의 알고리즘보다 움직임 특성을 판단하는 부분이 추가되어 계산량이 조금 늘어났고, 실험결과에서 평균 0.4초 정도의 시간 증가를 나타내지만 기존의 빠른 인코딩속도를 유지한다. 마지막으로, 실험에 사용한 영상들에 대해서 제안하는 방법은 기존의 C-1BT보다 평균 0.11dB PSNR 향상을 가져왔고 실험 결과에서 확인 하였다.

5. 감사의 글

“이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (2011-0011312)”

참고문헌

- [1] Z.-L. HE, C.-Y. Tsui, K-K.Chan, and M.L.Liou, "Low-power VLSI design for motion estimation using adaptive pixel truncation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no.5, pp.669-678, Aug. 2000.
- [2] 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.
- [3] 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.
- [4] O. Urhan and S. Ertürk, "Constrained one-bit transform for low-complexity block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 17, no. 4, pp. 478-482, Apr. 2007.
- [5] X. Yi and N. Ling, "Rapid block-matching motion estimation using modified diamond search algorithm," in *Proc. IEEE ISCAS 2005*, May 2005, vol. 6, pp.5489-5492.
- [6] 김재훈, 길일승, 김기백, 정제창, "제약있는 이진 블록 정합 움직임 추정에서 프레임 단위로 적응적 거리 선택 알고리즘," 2011 신호처리합동학술대회, Sep.2011.
- [1] Z.-L. HE, C.-Y. Tsui, K-K.Chan, and