

H.26L 저전력 움직임 추정 구조

김태욱, 김재호
부산대학교 전자공학과

Low Power Motion Estimation Architecture for H.26L

Tae-wook Kim, Jae-ho Kim
Dept. of Electronics Eng. , Pusan National University
Email : terry2000@orgio.net

요약

본 논문은 영상압축 표준의 하나로 표준화가 진행중인 H.26L에 효율적인 저전력 움직임 추정 구조를 제안한다. 제안하는 방식은 움직임 추정에 사용하는 이전 프레임에서의 움직임 벡터 발생 빈도와 경향을 이용하여 계산량과 수행시간을 줄인다. 그리고 가변 블록 정합을 고려하여 먼저 최소 블록 크기 단위로 블록 SAD를 계산한 후 다른 모드 블록 SAD를 계산으로 생성한다. 제안하는 방식은 기존의 저전력 블록 정합 방식과 비교하여 최대 31% 전력 소모 감소가 이루어지며 완전 전역 탐색 블록 정합 방식에 비해 평균 75~90%의 계산량이 감소된다.

1. 서론

H.26x 나 MPEG 과 같은 동영상 압축 표준안에서는 움직임 보상 부호화 방식으로 블록 정합 알고리즘을 사용한다. 최근에 표준화가 진행중인 H.26L에서는 기존의 블록 정합 알고리즘과는 달리 블록 정합에 사용되는 블록의 크기가 다양하고, 이전 1 프레임만을 사용하던 것을 이전 5 프레임까지 사용한다. 그리고 1/2 화소, 1/4 화소, 1/8 화소까지 사용하여 보다 정확한 움직임을 추정하여 보상된 영상과 원 영상과의 차이를 줄여 압축의 효율을 증대 시킨다.^[1]

최근까지 계산량과 처리 속도를 개선한 다양한 고속 움직임 추정 알고리즘들이^{[2]-[4]} 나왔으나 이들 알고리즘은 실제 하드웨어 구현의 어려움과 화질의 저하, 그리고 H.26L 과 같이 다양한 블록 크기를 사용한 움직임 추정 방식에는 적용하기 어렵다. 최근에는 계산량과 전력 소비가 많은 움직임 추정을 저전력 구조로 설계하기 위한 많은 연구가 진행되고 있다.^{[5]-[9]}

II. 저전력 움직임 추정 구조

블록 정합 알고리즘의 탐색영역은 기준 블록이 $N \times N$ 의 화소로 구성되며 탐색범위가 $\pm p$ 일 때 $(2p+N)^2$ 크기를 가진다. 움직임 벡터는 탐색 영역의 $(2p+1)^2$ 개 후보 블록들 중 가장 작은 SAD(Sum of Absolute Difference)를 가지는 블록의 위치가 된다.

$$D(u,v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C(i,j) - P(u+i,v+j)| \quad (1)$$

$$(x,y) = (u,v) | \min D(u,v) \quad (2)$$

수식 (1)과 (2)은 SAD $D(u,v)$ 와 움직임 벡터 (x,y) 를 계산하기 위하여 사용한다. 여기서 $C(i,j)$ 는 현재 프레임의 기준 블록 화소값이며, $P(u+i,v+j)$ 는 이전 프레임의 탐색영역내의 후보 블록 화소값을 나타낸다.

저전력 움직임 추정 알고리즘의 SAD 계산식은 수식 (3)과 같이 변형된다.

$$D^i(u,v) = D^{i-1}(u,v) + D(i), \quad 0 \leq i \leq N-1 \quad (3)$$

$$D(i) = \begin{cases} \sum_{j=0}^{N-1} |C(i,j) - P(u+i, v+j)|, & \text{if } D^{i-1}(u,v) < D_{\min} \\ D(i-1), & \text{otherwise} \end{cases}$$

$$D(u,v) = D^{N-1}(u,v)$$

$$D_{\min} = \min\{D(u,v)\}$$

여기서 $D^{i-1}(u,v)$ 는 현재 블록에서 이미 계산된 부분 SAD 이고 $D(i)$ 는 AD(Absolute Difference)를 누적 크기까지 더한 값이다. D_{\min} 는 이전 후보 블록의 최소 SAD 이다. 부분 SAD $D^{i-1}(u,v)$ 가 이전 블록 최소 SAD D_{\min} 보다 적다면 누적 크기만큼 AD를 계산하여 $D(i)$ 를 생성한다. 수식(3)과 같이 이를 $D^{i-1}(u,v)$ 와 더하여 새로운 부분 SAD $D^i(u,v)$ 을 만든다. 만약 $D^{i-1}(u,v)$ 값이 현재 최소 SAD D_{\min} 보다 크다면 더 이상의 AD 계산을 중지한다. 따라서 i 에서 $N-1$ 까지의 SAD 계산을 하지 않는다. $N-1$ 까지 부분 SAD의 값이 이전 블록 최소 SAD보다 작다면 D_{\min} 는 새로운 최소 SAD 값으로 갱신이 된다.

저전력 움직임 추정 알고리즘의 전력 소모를 줄이는 방식은 AD 계산과 누적 크기마다 일어나는 비교기 사용(CMP)과 AD의 누적(ADD)을 막음으로써 이루어진다. 현재 최소 SAD와 부분 SAD와의 비교 횟수를 줄이기 위해서 누적 크기를 일정한 크기로 정하거나 가변적인 크기로 한다.^{[5]-[9]}

III. 제안하는 알고리즘

H.26L 움직임 추정 방식에는 7가지의 모드를 가지고 있으며 각 모드마다 움직임 추정에 사용되는 블록 크기가 다르다.^[1] 탐색범위가 $\pm p$ 일 때 모드 1의 경우에는 16×16 화소의 블록 크기를 가지며, 이전 프레임의 $(2p+16)^2$ 크기의 탐색 영역에서 $(2p+1)^2$ 개의 후보 블록 중 가장 작은 SAD를 가지는 블록을 찾는다. 모드 7의 경우 4×4 화소 크기의 16개의 블록으로 구성되어 각 블록이 서로 다른 탐색 영역을 가지며 각각 최소 SAD 블록을 찾는다. 각 모드의 탐색범위가 동일한 경우 그림 1에서 보는 바와 같이 모드 1의 16×16 블록

탐색영역과 모드 7의 16개의 4×4 블록 탐색영역이 동일하다. 이 점을 이용하여 최소 블록 크기를 가지는 4×4 블록의 SAD를 계산하여 다른 크기의 블록 SAD를 계산으로 생성한다.

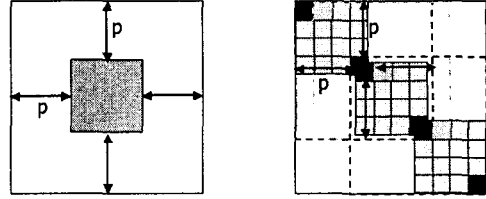


그림 1. 각 모드의 탐색 영역

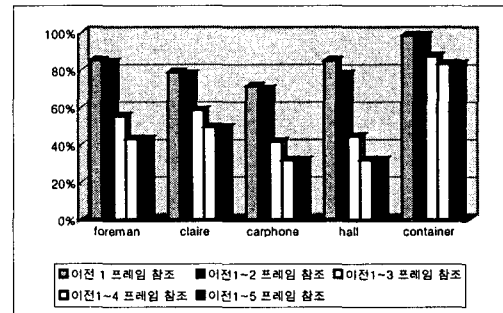


표 1. 움직임 벡터가 이전 프레임의 중심에 분포하는 정도

표 1은 움직임 벡터가 각 이전 프레임 중심($-1 \leq x, y \leq 1$)에 움직임 벡터가 얼마나 분포하는지를 실험한 결과이다. 이전 1 프레임만을 참조할 경우 움직임 벡터의 70%에서 98%까지 중심에 위치한다. 이전 3 프레임 이상을 참조할 경우 움직임 벡터가 각 프레임의 중심에 위치하는 확률이 낮아지게 된다. 이는 더 이전 프레임으로 갈수록 현재 프레임과의 상관관계가 떨어지므로 중심과 먼 위치에서 최소 SAD가 나온다. 움직임이 적은 시퀀스의 경우에는 움직임 벡터의 78%가 이전 1 프레임에서 발생하고 움직임이 많은 경우 40% 정도가 나타난다. 16×16 크기나 8×16 등의 비교적 큰 블록일수록 이전 1 프레임에 움직임 벡터가 나타나는 정도는 커진다.

움직임 추정시 가변 블록 크기와 이전 5 프레임을 사용할 경우 움직임 벡터의 발생 위치와 프레임의 분포를 이용하여 후보 블록을 탐색하는 순서와 SAD 누적 방식을 제안한다.

3.1 후보 블록 탐색 순서

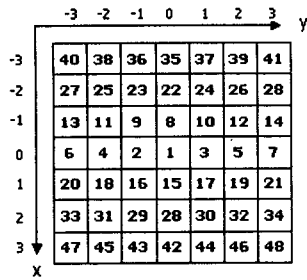


그림 2. 이전 1 프레임의 탐색 순서

그림 2는 탐색 범위를 ± 3 으로 했을 경우 그림과 같은 순서로 먼저 이전 1 프레임의 움직임 벡터 x 좌표 값이 0인 후보 블록을 차례로 탐색한다. 그런 다음 $(-1,0)$ 에서 시작하여 움직임 벡터 x 좌표 값이 -1인 후보 블록을 계산한다. 다음 이전 2~5 프레임의 탐색 방식은 이전 2 프레임의 $(0,0)$ 에서 탐색을 시작한다. 연속적으로 이전 2~5 프레임의 x 좌표 움직임 벡터가 0인 영역을 탐색하고 탐색이 모두 끝나면 다시 이전 2 프레임의 $(-1,0)$ 후보 블록에서 탐색을 시작한다. 이전 2~5 프레임까지의 x 좌표가 -1인 영역을 계속 탐색한다. 각 이전 프레임 내에서의 순서는 이전 1 프레임과 동일하다.

3.2 SAD 누적 방식

제안하는 SAD 계산방식은 그림 3에 표시된 1번 화소 AD 그룹을 누적 시킨 후 2번 그룹 순서로 누적 시킨다. 기존의 순차 누적 방식보다 빨리 탈락 후보 블록을 찾아 계산량이 감소된다. 이 방식을 7가지 모드의 모든 블록 SAD를 계산할 때 적용한다.

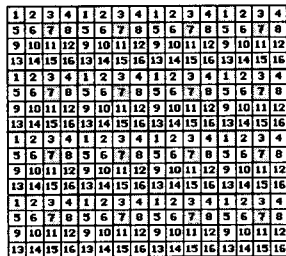


그림 3. SAD 누적 방식

3.3 구조

제안하는 구조는 16×16 PE(Processing Element)를 가지고 있고 이것은 16개의 4×4 PE 어레이 그룹으로 나누어져 독립적으로 AD를 계산하게 된다. 총 41개의 모드 블록별 비교기가 있어서 각 블록에 대한 최소 SAD를 계산한다. 각 모드 블록에 해당하는 PE 어레이 그룹의 AD값만을 받아서 합산기를 통하여 누적 크기까지 더한다. 그런 후 각 이전 블록의 최소 SAD와 비교하여 현재 블록의 부분 SAD가 같거나 크면 해당 블록 AD를 누적 시키는 것을 중단하고 stop 신호가 set된다. 모든 블록의 비교기에서 stop 신호가 set되면 PE 어레이에서 AD계산을 중지한다. 마지막까지 한 모드 블록이라도 stop 신호가 set이 되지 않으면 계속 AD의 계산을 수행하게 된다. 비교기에서는 누적 크기마다 비교를 하며 누적 크기가 작을수록 비교 횟수는 늘고 AD와 ADD 횟수는 줄어든다.

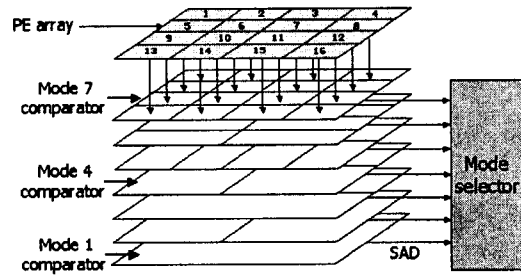


그림 4. 제안하는 저전력 움직임 추정 구조

IV. 실험 결과

본 논문에서는 H.26x에서 사용하는 QCIF 포맷 5종류 시퀀스를 사용하여 기존 저전력 알고리즘^[5]과 결과를 비교하였다. 사용한 프레임 수는 100 프레임이고 탐색 범위를 ± 12 로 했을 경우 누적 크기에 따라 결과를 비교하였다. 전력소모 비교를 위해 ADD의 전력 소모를 1S로 할 때 AD는 2S, CMP는 1S로 정의한다.^{[5][7]}

표 2는 foreman 시퀀스를 사용하여 1 프레임 움직임 벡터를 찾는 데 사용한 CMP와 ADD, AD의 횟수와 전력 소모량을 누적 크기에 따라 기존의 저전력 움직임 추정 알고리즘과 비교하였다.

sequence	Proposed method				Conventional method			
	accumulation size				accumulation size			
	1	4	8	16	1	4	8	16
foreman	60156713.2	14746438.4	7082947.4	1502755.6	79542171.7	16883076.8	7107334.9	2552931.2
CMP	60156713.2	75945436.5	98291445.6	178783400.1	79542171.7	98313303.5	12732358.7	194751880.7
ADD	35104742.1	38180726.0	45882368.0	64577280.0	54620408.1	551335368	588009528	64577280.0
Power	190522910.6	167053326.9	197099129	309450715.7	268325159.6	225463453.9	250432837.2	326459371.9

표 2. 기존 방식과의 계산량 비교

sequence	Proposed method				Conventional method			
	accumulation size				accumulation size			
	1	4	8	16	1	4	8	16
foreman	190522910.6	190522910.6	190522910.6	190522910.6	268325159.6	225463453.9	250432837.2	326459371.9
carphone	167053326.9	167053326.9	167053326.9	167053326.9	237721531.8	208921890.2	239124070.8	320122640.9
claire	225463453.9	225463453.9	225463453.9	225463453.9	260516484	123480901.3	247733883.8	324639008.6
hall	197099129	197099129	197099129	197099129	264295693	223297922	250317253.9	327236469.6
container	250432837.2	250432837.2	250432837.2	250432837.2	249383771.8	214762045.8	24280786.5	322318056.2

표 3. 누적 크기에 따른 전력 소모 비교

sequence	Percentage of power saved			
	1	4	8	16
foreman	29.00%	25.90%	21.30%	5.21%
carphone	33.50%	28.10%	22.51%	5.25%
claire	29.80%	24.11%	21.59%	5.44%
hall	24.66%	21.24%	16.24%	3.97%
container	37.99%	30.21%	21.87%	6.70%
average	30.99%	25.91%	20.78%	5.31%

표 4. 누적 크기에 따른 전력 소모 감소

표 3, 4 는 제안 방식의 누적 크기에 따른 전력 소모와 전력 소모 감소량이다. 누적 크기를 1 로 하였을 경우는 기존의 방식에 비해 평균 31% 정도의 전력 소모가 감소한다. 누적 크기가 4 인 경우에는 약 26% 정도의 전력 소모 감소를, 누적 크기가 16 인 경우 AD 의 계산량 감소는 없고 비교하는 횟수가 전체 비교 계산량의 1/10 정도로 줄었다. 따라서 전력 소모는 기존의 방식에 비해 약 5%정도 감소된다. 좀 더 객관적인 비교를 위하여 완전 전역 블럭 정합과 비교하면 평균 75%에서 90%정도의 계산량을 줄일 수 있다.

V. 결 론

H.26L 은 기존의 MPEG2, MPEG4, H.263 보다 더 좋은 성능을 가진다. H.26L 움직임 추정시 가변 블럭 크기와 이전 5 프레임을 사용하므로 계산량이 기존 동영상 압축에 비해 많다. 본 논문에서는 움직임 벡터의 발생하는 위치와 프레임의 분포를 이용하여 후보 블럭을 탐색하는 순서와 SAD 누적 방식을 제안한다. 제안한 알

고리즘과 구조를 사용할 경우 기존의 저전력 움직임 추정 방식에 비해 최대 31%정도의 전력 소모를 감소시킬 수 있으며 완전 전역 블럭 정합 계산량의 평균 75%~90%를 줄일 수 있다.

참고문헌

- [1] ITU STUDY GROUP 16 VCEG "H.26L Test Model Long Term Number 8 draft"
- [2] R. Li, B. Zeng and M. L. Liou "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits and Systems for Video Tech.*, Page(s): 438-442, Aug. 1994.
- [3] L. M. Po and W. C. Ma "A novel four-step search algorithm for fast block motion estimation" *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 6, Page(s): 313-317, June 1996.
- [4] Costa, A., et al "A VLSI architecture for hierarchical motion estimation" *Consumer Electronics, IEEE Transactions on*, Volume: 41 May 1995 Page(s): 248 -257
- [5] Sousa, L. and Roma, N. "Low-power array architectures for motion estimation" *Multimedia Signal Processing, 1999 IEEE 3rd Workshop on*, 1999 Page(s): 679 -684
- [6] Wujian Zhang and Runde Zhou "Low-power motion-estimation architecture based on a novel early-jump-out technique" *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, Volume: 5, 2001 Page(s): 187 -190
- [7] Do, V.L. and Yun, K.Y. "A low-power VLSI architecture for full-search block-matching motion estimation" *Circuits and Systems for Video Technology, IEEE Transactions on*, Volume: 8 Issue: 4, Aug. 1998 Page(s): 393 -398
- [8] Kin-Hung Lam and Chi-Ying Tsui "Low power 2-D array VLSI architecture for block matching motion estimation using computation suspension" *Signal Processing Systems, 2000. SiPS 2000. 2000 IEEE Workshop* Page(s): 60 -69
- [9] Berns, J.P. and Noll, T.G. "A flexible motion estimation chip for variable size block matching" *Application Specific Systems, Architectures and Processors, 1996. Proceedings of International Conference on*, 1996 Page(s): 112 -121