

## Optical flow 를 이용한 motion estimation 에 관한 연구 A study of a motion estimation with an optical flow

Byun Jaeyoung, Kim Jaeyoung, Kim Sangki, Lee Wonhee, Lee Bumro, Chung Chinhyun  
Kwangwoon Univ. Control & Instrumentation Engineering

Chaeung Byun, Jaeyoung Kim, Sangki Kim, Wonhee Lee, Bumro Lee, Chinhyun Chung  
Control & Instrumentation Engineering, Kwangwoon Univ.

**Abstract:** The purpose of image sequence coding is to reduce the spatio-temporal redundancies. The transform coding such as DCT is used for the spatial redundancies. In this paper, the optical flow method is applied to solve the problem of temporal redundancies. So far, pixel intensity conservation has been used to solve the optical flow. We used the neighborhood information as well as pixel intensity conservation. And we compared the merits and demerits of the conventional method and the proposed method in this paper.

### 1. 서론

영상압축의 목적은 영상이 가지는 spatial redundancy 와 temporal redundancy 를 제거하여 작은 정보량으로 그 영상을 저장하거나 전송하는 것이다. Spatial redundancy 는 DCT 와 같은 transform 코딩방식이나 waveform 코딩방식을 이용하여 줄이고, temporal redundancy 는 motion compensation 을 이용하여 줄일 수 있다. 현재, 표준으로 제정되어있는 MPEG 도 DCT 와 motion compensation 을 이용하고 있으며, block matching 방법을 이용하여 motion compensation 을 한다. 그러나, block matching 방법은 추정된 움직임벡터에 대한 신뢰성이 떨어지며, 블록화현상과 같은 여러 가지 단점이 있다. 움직임 벡터를 예측하기 위한 방법은 gradient 를 이용한 방법, pel-recursive 방법, block matching 방법, frequency-domain 방법등 4 가지로 나눌수 있다. 본 논문에서 적용한 optical flow 는 gradient 를 이용한 방법으로 image sequence 분석과 같은 응용분야에 적용되어왔다. 이 방법은 움직임계측을 따라 image 의 세기(휘도성분)가 변하지 않는다는 가정에 근거하고 있다. Block matching 방법이 두 연속된 frame 사이에서 블록단위로 가장 예러가 작은 블록을 찾아 움직임 벡터를 구하는 것에 비해 optical flow 를 이용한 방법은 각각의 픽셀에 대해 optical flow 를 계산하기 때문에 각각의 픽셀에 대한 아주 세밀한 움직임 벡터가 구해진다. 따라서, block matching 방법에 비해 계산과정이 복잡하기 때문에 하드웨어로 적용하

기가 힘들다, 움직임 벡터를 더 정확하게 찾으며, 계산시간에 대한 부하가 줄어들어 따라 많은 범위에 적용될 것으로 보인다. 본 논문에서는 기존의 Horn-Schunck 방법과 Lucas-Kanade 방법과 각각의 움직임 벡터에 대한 주변정보를 이용하여 개선한 방법을 비교하였다.

### 2. Optical Flow

Image sequence  $S(X,t)$ 의 변화에 근거하여, image 좌표계  $X$ 의 시간  $t_1$ 에서  $t_2$ 에서의 displacement 를 correspondence vector 라 하며, 식(1)로 정의한다.

$$d(X,t) = [d_1(X,t), d_2(X,t)]^T \quad (1)$$

따라서, 시간  $t$ 에서의 복원될 image 는 식(2)와 같이 표현된다.

$$s(x+d_1(X,t), y+d_2(X,t), t+\Delta t) = s(x, y, t) \quad (2)$$

식(2)는 image 에 있어서 조명의 변화가 없이 세기의 변화가 단지 displacement( $d_1, d_2$ )에만 영향을 받는다는 가정을 두고 있다. Optical flow 벡터는 image 좌표계에서 임의의 픽셀( $X,t$ )에서의 순간적인 변화율로 정의한다. Optical flow 를 이용한 움직임추정은 주어진 image 에서의 속도성분벡터를 구하는 것이다. 따라서, 식(2)를 식(3)과 같이 쓸 수 있는데, 2 프레임에서 속도성분에

변화가 없이 일정하다는 가정에 근거한다. 즉,

$$s(x+v_1(X,t), y+v_2(X,t), t+\Delta t) = s(x, y, t) \quad (3)$$

$v(X, t)$ ,  $v(X, t)$ 는 각각  $\frac{dx}{dt}$ ,  $\frac{dy}{dt}$ 로서 image 좌표계에서의 좌표 각각의 속도벡터 성분을 나타낸다. 식(3)을 다시 정리하면, 시간의 변화에 따라 image  $s_c(x, x, t)$ 의 각각의 픽셀점은 움직임 궤적을 따라 일정하게 되는 점을 찾아가는데 식(4)와 같이 된다.

$$\frac{ds_c(x, y, t)}{dt} = 0 \quad (4)$$

식(4)는 시간의 변화에 따른 image-plane 좌표계의 변화율로서, 움직임 궤적에 따른 세기의 변화율을 나타낸다. 식(4)에 chain rule을 적용하면 식(5)와 같이 표현될 수 있다. 식(3)을 Taylor's expansion을 이용하여 고차항은 버리고, 나머지 초기 몇 개의 항만을 선택하여 식(5)를 유도할 수 있다.

$$\frac{\partial s_c(X; t)}{\partial x} v_1(X, t) + \frac{\partial s_c(X; t)}{\partial y} v_2(X, t) + \frac{\partial s_c(X; t)}{\partial t} = 0 \quad (5)$$

식(5)를 optical flow equation 또는 optical flow constraint라 한다.

식(5)를 gradient operator를 이용하여 다시 쓰면 식(6)과 같이 된다.

$$\nabla s_c(X, t) \cdot v(X, t) + \frac{\partial s_c(X, t)}{\partial t} = 0 \quad (6)$$

$\nabla s_c(X; t)$ 는  $x, y$ 에 대한 gradient를 나타낸다. Image-plane에서 식(6)을 이용하면, 임의의 한점에서 움직임으로 인한 세기의 변화에 따른 하나의 값을 구할 수 있다. 반면에, 그점에서의 움직임 벡터는 2 변수를 가지기 때문에 식(6)만으로는 움직임 벡터를 구할 수 없는데, aperture problem이라 한다. 즉, image-plane에서의 gradient 방향에 있는 움직임만을 알 수 있다. 따라서, 이러한 문제를 해결하기 위한 또다른 조건이 필요하게 되었다. Horn-Schunck는 smoothness 조건을 주어 optical flow gradient 제공값이 최소가 되는 값을 찾는 방식으로, 픽셀과 픽셀 사이의 속도 벡터의 변화에 대한 조건을 다음 식(7)과 같이 주

어서 optical flow를 계산한다.

$$\varepsilon_s(v(X, t))^2 = \left(\frac{\partial v_1}{\partial x}\right)^2 + \left(\frac{\partial v_1}{\partial y}\right)^2 + \left(\frac{\partial v_2}{\partial x}\right)^2 + \left(\frac{\partial v_2}{\partial y}\right)^2 \quad (7)$$

따라서, 속도 벡터가 더 smooth하면, 에러값은 더 작아지게 된다. 식(7)을 이용하여 optical flow equation은 다음 식(8)을 최소가 되게 함으로서 구할 수 있다.

$$\iint \left\{ \varepsilon_o(v(X, t))^2 + \alpha^2 \varepsilon_s(v(X, t))^2 \right\} dx dy \quad (8)$$

또 다른 방법으로는, 움직임 벡터가 특정한 블록내에서 일정하다는 것으로 Lucas-Kanade에 의해 제안되었다. 즉, 특정 블록 B에서 각각의 픽셀점에서의 속도성분은 모두 같은 방향의 속도성분을 가진다는 가정으로서, 속도성분은 다음 식(9)와 같이 표현될 수 있다.

$$v(X, t) = v(t) = [v_1(t) v_2(t)]^T \quad (9)$$

이 방법은 rotation과 같은 움직임은 해결할 수 없지만, 단지 translation만 있을 경우에 적용할 수 있는데, 픽셀에서의 세기 성분의 변화가 충분해야 된다.

### 3. Neighborhood Information

일반적으로, 한 중심 픽셀에서의 움직임 벡터는 픽셀 주위의 움직임 벡터로부터 정보를 구할 수 있다. 각각의 주변 픽셀의 움직임 벡터는 중심 픽셀로부터의 거리에 따라 영향을 미치는 정도가 다르기 때문에, 본 논문에서는 spatial image gradient가 일정하다는 가정을 두어 식(4)로부터 각각의 움직임 벡터를 구한다. 그리고, 각각의 좌표값에 대한 gradient는 weighting matrix를 이용하여  $3 \times 3$  주변 픽셀정보를 이용하였다. 그리고, 프레임사이에서의 움직임 벡터는 smooth하게 변한다는 가정을 두어 현재 t 프레임의 gradient를 이전 t-1 프레임과 이후 t+1 프레임의 gradient를 이용하여 위의 weighting matrix를 이용하여 계산하였다. 다음 식(10)은 주변 움직임 정보를 이용하여 계산한 gradient이다.

$$E(G(X, t)) = \frac{1}{2} \{W[G(X, t-1)] + W[G(X, t+1)]\} \quad (10)$$

위 식(10)에서 이용한 Weighting function 은 2-D triangular window 를 이용하였다.

#### 4. Gradient Estimation

좌표성분에 대한 gradient 는 다음과 같이 2 가지 방식을 이용하여 구하였다. 첫 번째 방식은 finite differences 를 이용한 것으로 중심픽셀  $\pm 1$  주위의 픽셀과 다음 프레임에서의 같은 픽셀의 값들의 합을 평균하여 근사화된 값을 사용하였다. 두 번째 방식은 polynomial fitting 을 이용한 방식으로 image  $S_c(x, y, t)$ 를 다음 식(10)과 같이 다항식으로 근사화하여 계산한다.

$$s_c(x, y, t) \approx \sum_{i=0}^{N-1} a_i \phi_i(x, y, t) \quad (11)$$

$a$  는 각각의 다항식에 대한 계수값,  $\phi$  는 기본다항식,  $N$  은 다항식 개수를 나타낸다. 위 근사치를 이용한 optical flow 계산은 원 픽셀값과 근사화된 값의 차이가 최소가 되는 계수값을 찾아냄으로써 구해지는데, 다음 식(11)과 같이 least square error 를 이용하여 계수값  $a$  를 계산한다.

$$e^2 = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \sum_{k=0}^{N-1} (s(n_1, n_2, k) - \sum_{i=0}^{N-1} a_i \phi_i(x, y, t))^2 \quad (12)$$

위 식(11)을 다시 계수  $a$  에 대해 계산하면 각각의 좌표값에 대한 gradient 를 계산할 수 있다.

#### 5. 모의 실험

실험에서는 시간상의 연속된 2 프레임 을 이용하여 Lucas-Kanade 방법, Horn-Schunck 방법과 제안한 방법에 대해 움직임 벡터를 구하였다. 픽셀사이의 정보에 대한 정확성을 높이기 위해 Raw image sequence 에 gaussian smoothing 을 이용하여 전처리 를 한후에 움직임 벡터를 구하였다. 복원된 영상에 대해서는 프레임사이의 에러 신호에 대한 보상을 하지 않았다. Horn-Schunck 방법에서는  $\alpha^2$  값을 100 으로 하였으며, Gauss-Seidel 방식을 이용하여 15 번 반복하여 각각의 좌표값에 대한 속도성분을 구하였다. Lucas-Kanade 방법에서는 블록을  $5 \times 5$  로 하여 계산을 하였다. 제안한 방법은 weighting matrix 를  $3 \times 3$  으로 하여 움직임 벡터를 구하였다. 또, 각각의 기본항에 대한 계수값  $a$  를 구하였다. 그리고, 각각의 반복계산을 수행할 때, 속도성

분의 초기치는 0 으로 하여 계산하였으며, 2 번째 반복부터는 주위 8-direction 의 평균을 계산하여 현재 좌표값으로 취하였다. 각 방법에 대한 성능비교를 위해 PSNR 값을 이용하였다. 다음 그림 1, 2 는 각방법들에 대해서, table tennis sequence, mobile calendar sequence 에 대해 PSNR 값과 entropy 값을 비교한 것이다. 그림 3, 4 에는 각각의 sequence 에 대한 움직임 벡터를 나타내었는데, 그림에서 알 수 있는바와 같이 table tennis sequence 에 대해서는 estimation 을 잘하는 것을 알 수 있다. 그러나, mobile calendar sequence 에 대해서는 estimation 을 잘하지 못하는데, 중심픽셀 주위의 픽셀값들을 전처리 과정과 부가적인 개선된 알고리즘을 통하여 방향성에 대해 어느정도 정확한 정보를 주어야 될 것 같다. 각각의 좌표값에 대한 미분값은 polynomial fitting 방식을 이용한 Horn-Schunck 방법이 상당히 많은 계산시간이 걸렸는데, 약 22 분정도 걸렸다. 다음 그림 1, 2 에서  $\Delta$  는 제안한 방법,  $\square$  는 Horn-Schunck 방법,  $\diamond$  는 Lucas-Kande 방법이다.

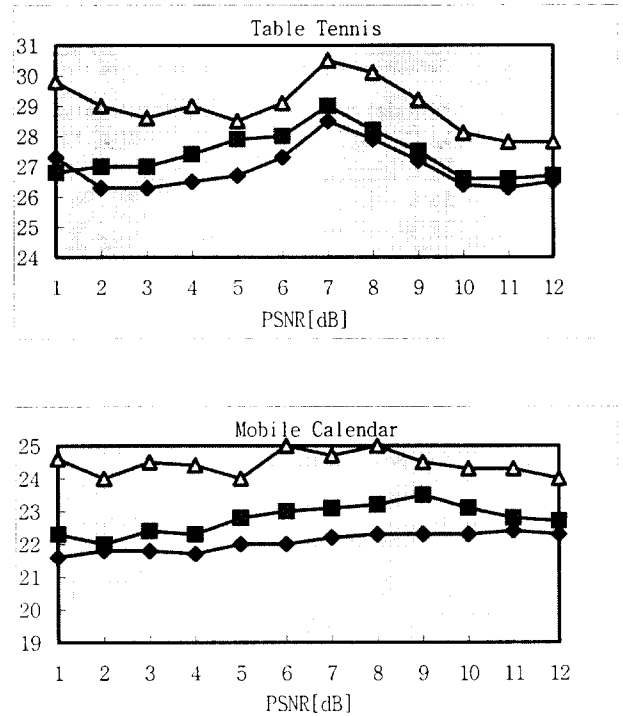


그림 1. 복원영상의 PSNR 값; (a) Table Tennis, (b) Mobile Calendar

Figure 1. The PSNR value of the reconstructed image frame;

(a) Table Tennis, (b) Mobile Calendar

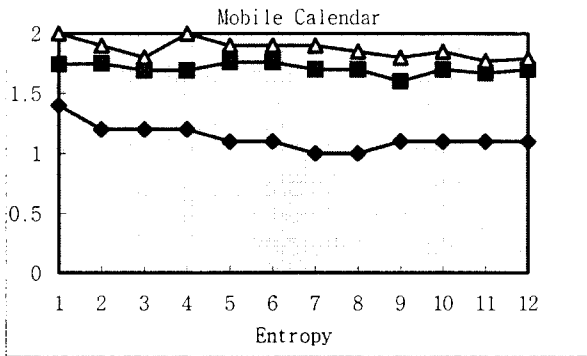
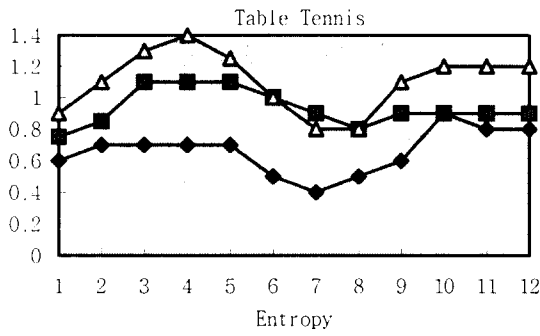


그림 2. 각각의 프레임에 대한 Entropy 값.

Figure 2. The entropy value of the reconstructed image frame.

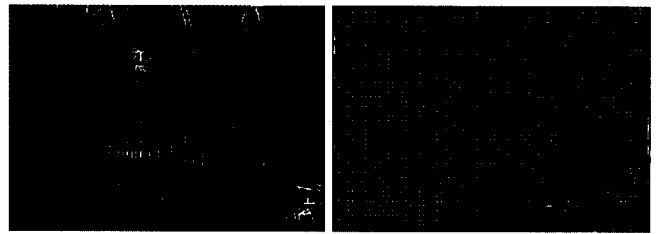
## 6. 결론

그림 3, 4에서 알 수 있는바와 같이 table tennis sequence에 대해서는 움직임 벡터를 아주 정확하게 잘 추정함을 알 수 있다. 그러나, mobile calendar sequence에 대해서는 움직임 벡터를 그다지 정확하게 추정하지 않았는데, 상당히 움직임과 픽셀세기의 분포가 복잡하기 때문에 각각의 좌표값에 대한 gradient를 부가적인 정보를 더 주어야 함을 알 수 있다. 그림 1, 2에서 알 수 있는 바와같이 제안한 방법이 다른 방법에 비해 PSNR 값이 높게 나온 반면 entropy 값이 높음을 알 수 있다. 그리고, 에러신호에 대한 보상을 하지 않았기 때문에 약간 PSNR 값이 낮게 나왔다.



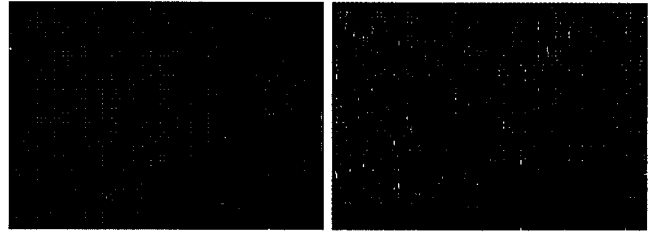
(a)

(b)



(c)

(d)



(e)

(f)

그림 3. 각각의 방식에 대한 움직임 벡터: (a) (d) 제안한 방법, (b) (e) Lucas-Kanade 방법, (c) (f) Horn-Schunck 방법.

Figure 3. The motion vector obtained from first and second frame ; (a) (d) proposed, (b) (e) Lucas-Kanade, (c) (f) Horn-Schunck.

## 참고 문헌

- [1] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," Proc. DARPA Image Understanding Workshop, pp. 121-130, 1981
- [2] B. K. P. Horn and B. G. Schunck, "Determining optical flow," Artif. Intell., vol. 17, pp. 185-203, 1981.
- [3] J. K. Aggrawal and N. Nandhakumar, "On the Computation of Motion from Sequences of Images-A Review," Proc. IEEE, vol. 76, pp. 917-935, Aug. 1988.
- [4] F. Dufaux and F. Moscheni. "Motion Estimation Techniques for Digital TV: A Review and a New Contribution," Proc. IEEE, vol. 83, pp. 858-876, June 1995.
- [5] A. murat tekalp, "Digital video processing," Prentice Hall Signal Processing Series, 1995.
- [6] A. K. Jain. "Fundamentals of digital image processing." Prentice Hall, 1989.