

Block-Based Method 를 이용한 Motion Estimation 에 관한 연구

A Study of a Motion Estimation with the Block-Based Method

°김 상기*, 이 원희*, 김 재영*, 변 재웅*, 이 범로*, 정 진현*

*광운대학교 제어계측공학과 (Tel: 940-5150; E-mail: sangki@ral.kwangwoon.ac.kr)

Abstracts: It is difficult that a non-translational motion in a block is estimated by the block matching algorithm (BMA). In this paper, a nodal-displacement-based deformation model is used for this reason. This model assumes that a selected number of control nodes move freely in a block and that displacement of any interior point can be interpolated from nodal displacements. As a special case with a single node this model is equivalent to a translational model. And this model can represent more complex deformation using more nodes. We used an iterative gradient based search algorithm to estimate nodal displacement. Each iteration involves the solution of a simple linear equation. This method is called the deformable block matching algorithm (DBMA).

Keywords: Motion, BMA, DBMA, Nodal Displacement, Shape Function

1. 서론

BMA 는 각각의 프레임을 독립적으로 정의하는 것 대신에 픽셀을 블록 단위로 나누고 한 블록에 대해서 다음 프레임을 검색하여 가장 큰 유사성을 가진 블록을 찾아 그 블록의 기준점의 이동을 모션벡터(motion vector)로 정의 하여 움직임을 추정하는 방법을 말한다. 주어진 프레임을 작은 겹치지 않는 정사각형의 블록으로 나누고 한 블록에서 모든 픽셀은 같은 평행이동을 갖는다고 가정한다. BMA 의 강점은 기본적인 motion model 을 매우 간단하고 효과적으로 결정하고 나타낼 수 있다는 점이다. 블록의 크기가 충분히 작을 때, 이 방법은 아주 좋은 움직임 예측이 가능하다.

우리가 일반적으로 BMA 를 이용하여 움직임 예측을 하는데 있어서 블록의 실제 움직임이 평행이동이 아닐 경우나 블록 내에서 혼합된 움직임이 존재할 경우에 오차가 발생할 수 있다. 그리고 화면의 갑작스러운 변화, 빛의 변화 또는 covered-uncovered 등의 문제에 의해 블록 내에서 픽셀 값이 변화할 때 역시 큰 오차로 작용한다. 이러한 오차 중에서 첫 번째 경우는 좀 더 복잡한 motion model 을 사용함으로 감소시킬 수 있고 두 번째 오차의 경우는 variable block size method 를 이용하면 감소시킬 수 있다. 세 번째 오차는 순방향 예측과 역방향예측을 사용함으로 오차를 줄일 수 있다. 본 논문에서는 nodal-displacement-based model 을 이용하여 블록의 움직임이 평행이동이 아닐 경우 그 오차를 줄이는 방법에 대해 연구하였다.

본 논문에서 제안된 nodal-displacement-based model 은 한 블록에서 선택된 제어 점의 수는 자유롭게 움직일 수 있다고 가정하고, 어떤 내부 점의 위치는 nodal displacement 로부터 유도될 수 있다. 이 모델은 단일 제어 점을 가지는 특별한 경우로 BMA 의 평행이동 모델을 포함하고 있고, 더 많은 노드를 사용함으로 좀 더 복잡한 변형의 모델을 표현할 수 있다. 수학적으로 이러한 모델은 bilinear deformation 함수와 같다. 본 논문에서 nodal displacement 를 결정하기 위해서 iterative gradient based

search algorithm 이 사용된다. 이러한 방법을 deformable block matching algorithm(DBMA)라고 한다.

실제로 translational model 은 image sequence 에서 많은 블록을 다루기에 충분하다. 일반적으로, 블록은 간단한 평행이동만 가지는 translation 블록과 단일의 non-translational deformation 을 가지는 non-translation 블록 그리고 이전의 프레임에서 영역을 찾을 수 없는 non-motion-compensable 블록이 있다. 따라서, 모든 블록에 DBMA 를 사용하는 것 보다는 먼저 BMA 를 이용하여 예측을 하고 예측 오차가 허용오차보다 작으면 translational model 로 간주하고 허용오차보다 크면 non-translational model 로 간주하여 다시 DBMA 를 처리하도록 하였다.

2. Nodal Displacement Based Deformation Model

두개의 이미지 평면 사이의 deformation 함수는 식(1)과 같다.

$$d(p) = \sum_{0 \leq l, m \leq N_1, l+m \leq N_2} a_{l,m} x^l y^m \quad (1)$$

여기서, $p = [x, y]^T$ 은 원래 이미지에서의 픽셀의 좌표이고, $d(p) = [\xi(p), \eta(p)]^T$ 은 변형된 이미지의 픽셀 p 의 displacement 이고, $a_{l,m} = [c_{l,m}, d_{l,m}]^T$ 는 2 차원 다항식의 계수이다. 특히 bilinear mapping 은 식(2)와 같다.

$$d(p) = a_{00} + a_{10}x + a_{01}y + a_{11}xy \quad (2)$$

만약 주어진 영역 D 상에서 어떤 control point 나 node p_k , $k = 1, 2, \dots, K$ 의 displacement d_k 를 알고 있다면, 식(3)과 같이 d_k 로부터 $p \in D$ 인 어떤 점들의 displacement 도 유도가 가능하다.

$$d(p) = \sum_{k=1}^K \phi_k(p) d_k \quad (3)$$

여기서, $\phi_k(p)$ 는 p 에 대한 k 번째 control point의 기여도에 따라 좌우된다. 이러한 함수를 shape function 또는 interpolation kernels라고 한다. 만족스러운 interpolation 결과를 위해서 $0 \leq \phi_k(p) \leq 1$, $\phi_k(p_l) = \delta_{k,l}$ 와 $\sum_k \phi_k(p) = 1$ 의 조건을 만족해야 한다.

위의 표현은 매우 일반적인 표현이고 control point가 적절하게 선택되어 졌을 때는 임의적으로 형성된 두 요소 사이의 mapping을 표현할 수 있다. 사실상 이러한 모델은 단일 node가 사용될 때는 translational 모델로 변한다. 또 다른 특별한 경우는 영역 D 가 well posed rectangular block일 때와 control point가 4개의 점일 때이다. 이런 경우에 식(3)은 블록에서 임의의 4각형으로의 mapping 형식으로 설명된다. 블록의 좌표가 $(-1, 1)$ 로 표준화 되어 있을 때, 식(2)의 bilinear mapping으로 nodal displacement의 이동은 다음의 shape function을 산출한다.

$$\begin{aligned}\phi_1(p) &= (1+x)(1-y)/4 \\ \phi_2(p) &= (1+x)(1+y)/4 \\ \phi_3(p) &= (1-x)(1+y)/4 \\ \phi_4(p) &= (1-x)(1-y)/4\end{aligned}$$

3. Nodal Displacement Estimation

선택된 예측 오차 기준이 최소가 되도록 nodal displacement를 어떻게 결정하는가를 고려해 보자. BMA 처럼 하나의 위치 벡터나 2차원 공간의 매개변수를 찾는 것이 아니고, K 개의 위치 벡터나 $2K$ 의 매개변수를 찾아야 한다. $K \geq 4$ 인 경우 한정된 범위 안에서 완전 검색(exhaustive search)조차 숫자적으로 엄청난데, 때문에 gradient search method와 modified Newton method를 사용한다.

$\psi_i(p), i=1,2$ 는 비디오 sequence의 두 프레임을 표현하고, D 는 두 번째 프레임의 정사각형 블록을 표현한다고 하자. 두 번째 프레임에서 첫 번째 프레임까지 4개의 꼭지점의 displacement를 d_k 라 가정하자. 그러면, 내부의 점 $d(p)$ 의 displacement는 식(3)에 의해 nodal displacement와 연관 지어진다. 이 블록의 mean square error (MSE)는 식(4)와 같다.

$$E = \frac{1}{2} \sum_{p \in D} (\psi_1(p+d(p)) - \psi_2(p))^2 \quad (4)$$

계산이 t 번 반복되었다면, 예측된 nodal displacement는 d_k^t 이고 deformation function은 $d^t(p) = \sum_k \phi_k(p) d_k^t$ 이다. 그러면 nodal displacement $\delta_k = [\delta_{x,k}, \delta_{y,k}]^T$ 의 작은 움직임들은 $d^t(p)$ 는 $d^{t+1}(p) = d^t(p) + \sum_k \phi_k(p) \delta_k$ 로 변할 것이다. Taylor 급수로 표현하고 고차 항을 무시하면 다음과 같은 식(5)가 얻어진다.

$$\psi_1(p+d^{t+1}(p)) \approx \psi_1(p+d^t(p)) + \nabla^T \cdot \sum_k \phi_k(p) \delta_k \quad (5)$$

여기서 주어진 변수를 고려한 부분적인 도함수를 나타내는 $\nabla = [\psi'_{1,x}(p+d^t(p)), \psi'_{1,y}(p+d^t(p))]^T$ 이다. 식(4)의 $\psi_1(p+d(p))$ 를 위의 근사식으로 대체시키고 δ_k 를 고려하여 최소값을 찾기 위해 미분을 하고 식을 0으로 맞추면, 식(6)이 얻어진다.

$$\sum_{l=1}^K \left(\sum_{p \in D} \phi_k(p) \phi_l(p) \nabla \nabla^T \right) \delta_l = \sum_{p \in D} e(p) \phi_k(p) \nabla \quad k=1,2,\dots,K \quad (6)$$

여기서 $e(p) = \psi_2(p) - \psi_1(p+d^t(p))$ 은 계산을 t 번 반복했을 경우 예측 오차이다. 위의 식 (6)은 다음 과 같이 행렬의 형태로 좀더 간단히 표현할 수 있다.

$$\begin{aligned}g_{x,k} &= \sum_p e(p) \phi_k(p) \psi'_{1,x}(p+d^t(p)) \\ g_{y,k} &= \sum_p e(p) \phi_k(p) \psi'_{1,y}(p+d^t(p)) \\ H_{xx,kl} &= \sum_p \phi_k(p) \phi_l(p) \left(\psi'_{1,x}(p+d^t(p)) \right)^2 \\ H_{yy,kl} &= \sum_p \phi_k(p) \phi_l(p) \left(\psi'_{1,y}(p+d^t(p)) \right)^2 \\ H_{xy,kl} &= \sum_p \phi_k(p) \phi_l(p) \psi'_{1,x}(p+d^t(p)) \psi'_{1,y}(p+d^t(p))\end{aligned}$$

$$\delta_x = \begin{bmatrix} \delta_{x,1} \\ \dots \\ \delta_{x,K} \end{bmatrix}, \quad \delta_y = \begin{bmatrix} \delta_{y,1} \\ \dots \\ \delta_{y,K} \end{bmatrix}$$

$$g_x = \begin{bmatrix} g_{x,1} \\ \dots \\ g_{x,K} \end{bmatrix}, \quad g_y = \begin{bmatrix} g_{y,1} \\ \dots \\ g_{y,K} \end{bmatrix}$$

$$H_{xx} = \begin{bmatrix} H_{xx,11} & \dots & H_{xx,1K} \\ \dots & \dots & \dots \\ H_{xx,K1} & \dots & H_{xx,KK} \end{bmatrix}$$

$$H_{yy} = \begin{bmatrix} H_{yy,11} & \dots & H_{yy,1K} \\ \dots & \dots & \dots \\ H_{yy,K1} & \dots & H_{yy,KK} \end{bmatrix}$$

$$H_{xy} = \begin{bmatrix} H_{xy,11} & \dots & H_{xy,1K} \\ \dots & \dots & \dots \\ H_{xy,K1} & \dots & H_{xy,KK} \end{bmatrix}$$

$$\delta = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix}; \quad g = \begin{bmatrix} g_x \\ g_y \end{bmatrix}; \quad H = \begin{bmatrix} H_{xx} & H_{xy} \\ H_{xy} & H_{yy} \end{bmatrix}$$

그러면 식(6)은 식(7)로 표기되어 진다.

$$H\delta = g \quad (7)$$

여기에서 $\frac{\partial \psi}{\partial x} \cdot \frac{\partial \psi}{\partial y} \equiv 0$ 이므로 계산을 간단히 하기 위해서 x 방향과 y 방향의 displacement 따로 계산한다.

$$H_{xx}\delta_x = g_x \cdot H_{yy}\delta_y = g_y$$

이때, 두 개의 $K \times K$ 행렬의 역행렬이 필요하다. 본 논문에서는 H 는 8×8 행렬이고, H_{xx}, H_{xy}, H_{yy} 는 4×4 행렬이다.

만약 초기 값이 실제 최소값과의 차이가 크다면, modified Newton algorithm은 잘못된 최소값에 접근해 갈 수 있다. 프레임 간의 움직임이 크다면, nodal displacement의 초기값의 설정은 중요한 문제가 된다. 이러한 이유 때문에 BMA를 이용하여 translational vector를 찾은 후에 이 벡터 값을 nodal displacement의 초기값으로 사용한다. 위 식의 반복적인 계산은 오차 감소 비율이 미리 정한 허용치보다 적어질 때 계산을 멈춘다.

4. 결과

그림 1은 본 논문에서 사용한 이미지이다. (a)와 (b)는 table-tennis 이미지이고, (c)와 (d)는 garden 이미지이다. 실제 이미지의 크기는 360×240 의 크기를 가지지만 본 논문에서는 이미지 중에서 실제 움직임이 가장 많이 존재하는 부분을 160×160 크기로 잘라 사용하였다. 프레임은 30 frame/sec로 만들어진 두 개의 연속된 프레임이다. 본 논문에서는 BMA를 이용하여 먼저 motion vector를 찾는다. 미리 정한 임계 값과 예측한 오차를 비교하여 오차가 임계 값보다 작으면 translational model로 간주하고, 오차가 임계 값보다 크면 non-translational model로 간주하여 DBMA를 이용하여 계산을 하였다. 여기서 BMA에서 찾은 motion vector를 DBMA의 초기값으로 사용한다. BMA에서 사용한 블록의 크기는 8×8 의 블록으로 하였고, 검색 영역은 ± 16 으로 하였다. 그림 2는 BMA를 이용하여 구한 motion vector이다. 실제로 이 모션 벡터만으로도 거의 실제에 가까운 프레임을 만들어 낼 수 있다. MPEG의 경우 BMA를 움직임 추정에 사용하는데 순방향과 역방향 예측을 함으로 오차를 줄이고 있다. 그림 3은 BMA를 이용하여 찾은 motion vector를 초기값으로 translation이 발생한 블록은 이동을 시키고, non-translation이 발생한 블록은 흰 색으로 표시한 그림이다. 다시 말하면, 흰 블록은 DBMA도 거쳐서 이동할 블록이고 나머지 블록은 BMA만으로 이동할 블록이다. 그림에서 보면 블록이 물체의 경계를 포함하고 있을 때 non-translation이 발생한다는 것을 알 수 있다. 다음의 그림 4는 DBMA의 과정을 거쳐 복원된 이미지이고 (a)table-tennis의 PSNR 값은 31.02dB이고 (b)garden 이미지의 PSNR 값은 25.69dB이다.

5. 추후 과제

본 논문에서는 두가지 model에 대한 움직임 예측을 하였다. 그러나 움직임에는 translation과 non-translation의 model 이외에 non-motion-compensable model이 있다. 본 논문에서는 이 모델을 고려하지 않고 DBMA 처리만을 하였는데, tree-mode hybrid method를 사용하여 처리하면 더 좋은 움직임 예측이 가능하다. 이 방법은 non-motion-compensable model에 대해 오차 보정을 가지는 BMA를 이용하여 움직임을 추정하게 된다.

이 외에도 nodal-displacement를 이용한 방법은 active-mesh, snakes, adaptive-mesh 등이 있다. 모두 nodal point의 개념을 기본으로 하여, non-motion-compensable model에 대한 BMA의 단점을 보완한 방법들이다.

참고 문헌

- [1] Frederic Dufaux and Fabrice Moscheni, "Motion Estimation Techniques for Digital TV: A Review and a New Contribution". Proc. IEEE, Vol. 83, No. 6, JUNE, 1995.
- [2] A. Murat Tekalp, "Digital Video Processing", Prentice-Hall, 1995.
- [3] Y. Wang and O. Lee, "Active Mesh - A Feature Seeking and Tracking Image Sequence Representation Scheme", IEEE Trans. Image Processing, Vol. 3, No. 5, September, 1994.
- [4] Y. Wang and O. Lee, "Motion Compensated Prediction Using Nodal Based Deformable Block Matching", Polytechnic University, Brooklyn, NY 11201.
- [5] O.C.Zienkewicz and R.L.Taylor, "Finite Element Method", Vol. 1, Prentice-Hall, 4-th ed., 1989.
- [6] 이 범로, "이미지 코딩을 위한 수정 퍼지 신경 회로망에 관한 연구", 광운대학교, 1994.
- [7] 이 병철, "동화상 압축을 위한 퍼지 움직임 추정에 관한 연구", 광운대학교, 1995.
- [8] 정 제창, "그림으로 보는 최신 MPEG", 교보 문고, 1995.

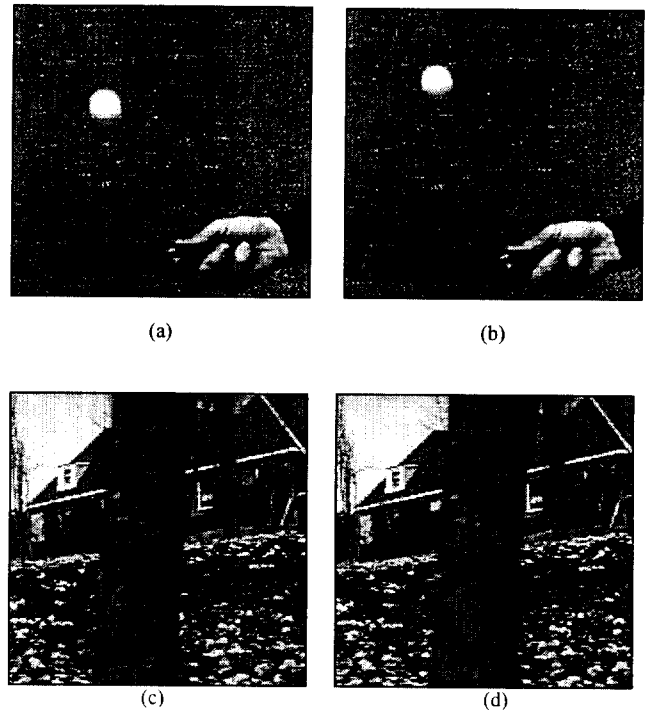


그림 1. 프레임 1 과 프레임 2 ; (a)table-tennis 1, (b) table-tennis 2, (c)garden 1, (d)garden 2
Fig 1. Frame1 and Frame2 ; (a)table-tennis 1, (b) table-tennis 2, (c)garden 1, (d)garden 2

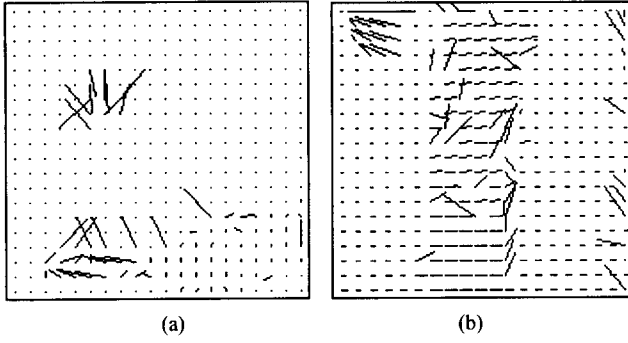


그림 2. BMA 를 이용한 모션벡터; (a)table-tennis (b)garden
 Fig 2. A motion vector using BMA; (a)table-tennis (b)garden

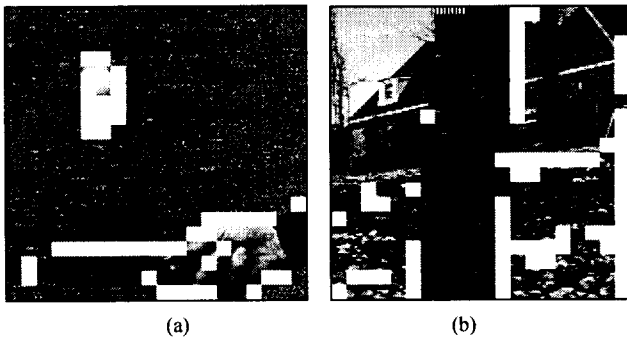


그림 3. Non-translational model (흰 블록); (a)table-tennis (b)garden
 Fig 3. Non-translational model (White block);
 (a)table-tennis (b)garden

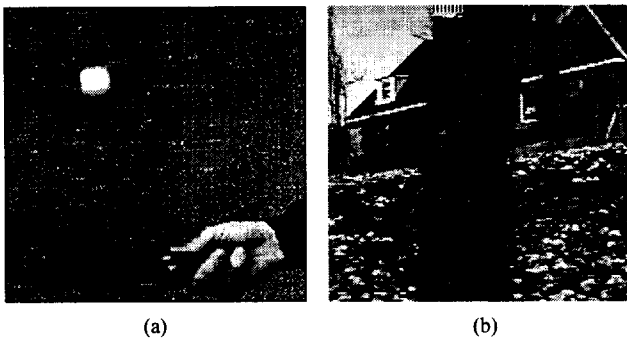


그림 4. 복원된 이미지; (a)table-tennis (b)garden
 Fig 4. Predicted images ; (a)table-tennis (b)garden