A Robust Algorithm for Tracking Non-rigid Objects

Jong Ryul Kim, Hyun Tae Na, Young Shik Moon
Department of Computer Science and Engineering, Hanyang University,
1271 Sa-Dong, Ansan, Korea,
Tel. +82-31-407-8991, Fax.: +82-31-419-1162
e-mail : jrkim@cse.hanyang.ac.kr, htna@cse.hanyang.ac.kr, ysmoon@cse.hanyang.ac.kr

Abstract: In this paper, we propose a new object tracking algorithm using deformed template and Level-Set theory, which is robust against background variation, object flexibility and occlusion. The proposed tracking algorithm consists of two steps. The first step is an estimation of object shape and location, on the assumption that the transformation of object can be approximately modeled by the affine transform. The second step is a refinement of the object shape to fit into the real object accurately, by using the potential energy map and the modified Level Set speed function.

Experimental results show that the proposed algorithm can track non-rigid objects with large variation in the backgrounds.

1. Introduction

Object segmentation and tracking are basic and essential steps for many video processing applications and the advanced video compression standards, such as surveillance system, MPEG-4 and MPEG-7. Since some of these applications require more accurate object shape and location, many algorithms for tracking an object with accurate object shape have been proposed. However, some methods assume the hypothesis of rigid object in the image sequences. If the object is non-rigid or the object shape flexibly changes, accurate tracking of the object becomes more difficult. To solve this problem, several algorithms have been proposed including Active Contour Model(*Snake*)[1][2], Deformable Template[3][4], and Level Set Theory[5][6].

In this paper, we propose a new tracking algorithm for non-rigid objects, which is robust against background variation, object flexibility, and occlusion, using deformed template and Level-Set theory. The proposed tracking algorithm is based on the model-based approach and the edge-based approach among the well-known tracking models.

The organization of this paper is as follows. First, we present the overall scheme of the proposed algorithm. In section 2, we define a potential difference energy function used in the process of estimation and refinement and describe main steps for object estimation and refinement. In section 3, the performance of the proposed tracking algorithm is evaluated with experimental results.

2. The Proposed Algorithm

The proposed tracking algorithm is composed of two parts. The first step is to estimate the shape and location of template object using affine transform. The second step is the refinement to accurately fit to real object shape using the potential map and Level Set speed funtion.

The overall scheme of the proposed object tracking algorithm is shown in Fig . 1.

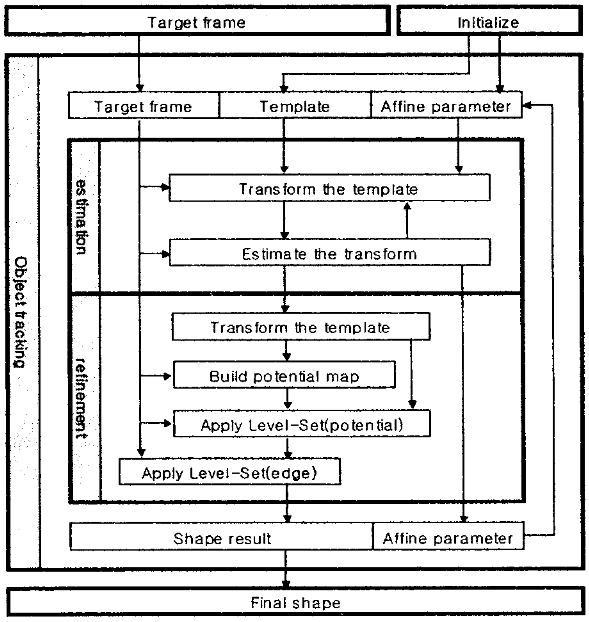


Fig. 1. Overall scheme of the proposed algorithm

2.1 Potential Difference Energy Function(PDEF)

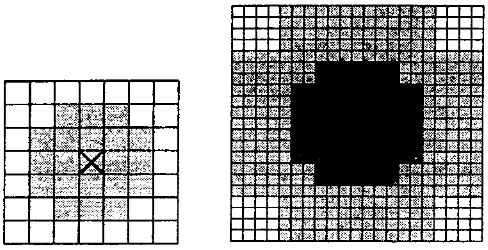
To estimate and refine the object shape in the next frame, we design a potential difference energy function(PDEF). This PDEF is used instead of pixel intensity in the process of object estimation and refinement. Therefore this energy function must be designed so as to have small values in case the object template is identical with the estimated object region and around edges. It is composed of two terms including inter-region distance and the edge values, which means the degree of difference between the initial template and the object shape estimated in the next frame. First, inter-region distance energy(E_{region}) is the summation of difference between each pixel intensity in the template and in the estimated object region on the target frame. According to whether the region belongs to the object or background, the weight $w(r)$ is differently applied as in equation (1).

$$w(r) = \begin{cases} 1-\gamma & r \text{ belongs to object region} \\ 0 & r \text{ belongs to shape region} \\ \gamma & r \text{ belongs to background region} \end{cases} \quad (1)$$

Inter-region distance energy(E_{region}) is calculated using equation (2).

$$E_{l,region}(X_s, X_c) = \sum_{dX \in region} w(R_{l,t}(X_s + dX)) \cdot \left(I_{l,t}(X_s + dX) - I_t(X_c + dX)\right)^2 \quad (2)$$

where l is multi-resolution level, t is template, X_s, X_c is a pixel on template and the target frame, and I, I_t is a pixel intensity on template and the next frame, respectively. $R_{l,t}(X)$ decides whether a pixel X belongs to object or background.



(a) 3×3 mask (b) mask for coarse-to-fine strategy

Fig. 2. Mask of weights

Fig. 2 shows the 3×3 mask of weights and mask of weights for coarse-to-fine strategy.
The final inter-region distance function is expressed as equation(3) by the summation for each multi-resolution level.

$$E_{region}(X_s, X_c) = \sum_{l \in level} E_{l,region}(X_s, X_c) \quad (3)$$

In order to express the energy of edge, we define it as the inverse derivative edge filter with a small value as in equation(4). Therefore, this energy has a small value around edges.

$$E_{edge}(X_c) = \frac{1}{|\nabla I_t(X_c)| + \xi}, \quad (4)$$

where $|\nabla I_t(X_c)|$ is the derivative edge detector and ξ is a small value. Finally, we define the potential difference energy function by multiplying two terms as in equation(5).

$$E_{potential}(X_s, X_c) = E_{region}(X_s, X_c) \cdot E_{edge}(X_c) \quad (5)$$

This potential difference energy function(PDEF) is used to estimate the affine motion parameters in the next process, instead of pixel intensity.

2.2 Estimation of object boundary

Affine model is widely used to get the parametric motion in many applications[6]. We also use the affine transform to obtain the motion parameters between the object template in the current frame and target object in the next frame, based on the assumption that the object transform follows the affine transform.

In this paper, to reduce the error caused by occlusion and background variation, we use the Tucky-Weight function as in equation(6).

$$w_{tw}(x, C) = \begin{cases} -(C^2 - x^2)^3 / C^6 & \text{if } |x| < C \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Also, in order to overcome the time consumption and local minima problem, we use the coarse-to-fine approach.

We firstly define the object PDEF ($E_{trans}(A)$) transformed to an arbitrary pixel X_s on the object template with affine transform matrix A as shown in equation (7).

$$E_{trans}(A) = \sum_{X_s \in shape} E_{potential}(X_s, trans(X_s, A)), \quad (7)$$

where $trans(X_s, A)$ is the location of point X_s moved by affine transform matrix A.

Since we consider the class of 2D polynomial motion models, these models can be expressed in a general way as in equation(8)

$$V_A(X_i) = B(X_i)A, \quad (8)$$

where $X_i = (x_i, y_i)$ denotes the spatial image position of a point and B is a matrix depending only on the point coordinates. Equation(8) is linear with respect to the n motion parameters($A = \delta$ as shown in equation (9)).

$$\delta = \left(\delta_a \quad \delta_b \quad \delta_c \quad \delta_d \quad \delta_e \quad \delta_f\right)^T \quad (9)$$

We can state the problem of estimating the parameter δ as the minimization of the cost function $E_{trans}(A)$ using iterative method and coarse-to-find strategy as in equation(10)-(12). We finally obtain an estimation of δ as in equation (13).

$$\hat{\delta}_n = \left(\hat{\delta}_a \quad \hat{\delta}_b \quad \hat{\delta}_c \quad \hat{\delta}_d \quad \hat{\delta}_e \quad \hat{\delta}_f\right)^T$$
$$A_n = \begin{pmatrix} \hat{\delta}_a & \hat{\delta}_b & \hat{\delta}_c \\ \hat{\delta}_d & \hat{\delta}_e & \hat{\delta}_f \end{pmatrix} \quad (10)$$

$$\begin{aligned} & E_{trans}(A_n) \\ &= E_{trans}(A_{n-1}) + \Delta E_{trans}(A_{n-1}) \\ &= E_{trans}(A_{n-1}) + \frac{d}{dA_{n-1}} E_{trans}(A_{n-1}) \cdot \hat{\delta} \\ &= \sum_{s \in shape} E(X_s, trans(X_s, A_{n-1})) + \sum_{X_s \in shape} \frac{d}{dA_{n-1}} E(X_s, trans(s, A_{n-1})) \cdot \hat{\delta} \\ &= \sum \left[E(X_s, trans(X_s, A_{n-1}))\right] + \sum \left(\frac{dE}{dx_s}, \frac{dE}{dy_s}\right) \cdot \left(\frac{dx_s}{dA_{n-1}}, \frac{dy_s}{dA_{n-1}}\right) \cdot \hat{\delta} \\ &= \sum \left[E(X_s, trans(X_s, A_{n-1}))\right] + \sum \left(\frac{dE}{dx_s}, \frac{dE}{dy_s}\right) \cdot \begin{pmatrix} x_s & y_s & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_s & y_s & 1 \end{pmatrix} \cdot \hat{\delta} \end{aligned}$$
(11)

$$B_{n-1} = \sum \left[w_{tw}(X_s, A_{n-1}) \cdot E_{potential}(s, trans(X_s, A_{n-1})) \right]$$

$$A_{n-1} = \sum \left[w_{tw}(X_s, A_{n-1}) \cdot \left(\frac{dE_{potential}}{dx_s}, \frac{dE_{potential}}{dy_s} \right) \cdot \begin{pmatrix} x_s & y_s & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_s & y_s & 1 \end{pmatrix} \right] \quad (12)$$

$$E_{trans}(\hat{A}_n) = B_{n-1} + A_{n-1} \cdot \hat{\delta}_n$$
$$\hat{\delta}_n = \left(A_{n-1}{}^T \cdot A_{n-1} \right)^{-1} \left(A_{n-1}{}^T \cdot -B_{n-1} \right) \quad (13)$$

The details of this estimation process are shown in the following.

```
function A = estimate(A0)
  iteration ← 0
  A ← A0
  while ( iteration < max_iteration ) {
    iteration ← iteration + 1
    compute A, B
    dÂ ← (A^T · A)^-1 (A^T · -B)
    weight = 1
    while ( weght < min_weight ) {
        If ( E_trans(A+αdÂ) < E_trans(A) ) {
            A ← A + αdÂ
            α ← α/2
        }
    }
    if (weight < α) return A
}
return A
end function
```

2.3 Refinement of object shape

In the previous process, we estimate the object position, but since it can't cover the change of object boundary itself, we add the refinement process for object boundary to handle the object flexibility. To exactly fit into the object boundary, we use the probability map of object shape and the modified Level Set speed function. PDEF defined in the previous section may be modified to be applied to Level Set theory by transforming the energy function to the probability function of object boundary.

The probability map is generally expressed as the inverse of potential difference energy function(PDEF). But, it needs to be normalized since each pixel on the boundary doesn't have a consistent probability value. We normalize the energy function(PDEF) for a pixel on the estimated boundary using only the higher ranked m values as in equation(14).

$$LP_{shape} = normalize(E_{potential}(X_s, X))$$
$$= \begin{cases} \dfrac{\min_m(E_{potential}) - E_{potential}(X_s, X)}{\min_m(E_{potential}) - \min_1(E_{potential})} & \text{if } E(X_s, X) \le \min_m(E_{potential}) \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

where $\min_m(E)$ is the m-th small value among pixels on the object boundary, and $\min_1(E)$ means the smallest value. The final probability map is defined by selecting the maximum value among the normalized probability for a pixel on the object boundary as shown in equation (15).

$$P_{shape}(X) = \max_{X_s \in shape} LP_{shape}(X_s, X) \quad (15)$$

This probability model is very similar to the probability map used in the maximum likelihood method.

Finally, to refine the exact object shape, we utilize the Level set speed function proposed by N. Paragios and R. Deriche[2], as shown in equation (16).

$$\phi_t = \left(g(I,\sigma) \cdot \kappa + \nabla g(I,\sigma) \cdot \frac{\nabla \phi}{|\nabla \phi|} \right) |\nabla \phi|, \quad (16)$$

where ϕ is the signed distance function, κ is a curvature for function ϕ at the given position, and $g(I,\sigma)$ is the gradient for edge image I.

The probability map may be directly applied to Level Set speed function. But, to overcome the problem of occlusion and execution time, we modify the original speed function by adding the term $\alpha{\cdot}sign(\Phi)$ as in equation (17).

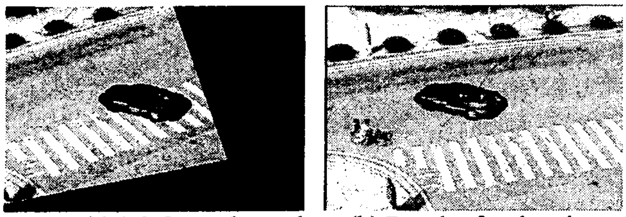
$$\phi_t = \left(\left(g(I,\sigma) + \alpha \cdot sign(\Phi_{t=0}) \right) \kappa + \nabla g(I,\sigma) \cdot \frac{\nabla \phi}{|\nabla \phi|} \right) |\nabla \phi| \quad (17)$$
$$sign(\Phi_{t=0}) = \begin{cases} +1 & \text{object region} \\ -1 & \text{background region} \end{cases}$$

The term $\alpha{\cdot}sign(\Phi)$ controls to keep the interface not getting out of the estimated position. That is, by adding the characterisic that the interface moves to the estimated position in the occluded regions, we solve the occlusion problem to some extent. If the value of α is large, the interface is easy to be fitted toward the initial gradient image. Otherwise, it is likely to be fitted toward the given probability map.

3. Experimental Results

We carried out expriments using various image sqeünces with object flexibility, background variation, and occlusion. Fig. 3 shows the procedures for tracking a car. Fig. 3 (a) is the initial deformed tempate. Fig. 3 (b) is the result of estimating the object position using the energy function and affine transform. Fig. 3 (c) is the probability map used to refine the exact object shape in the refinement process. Finally, Fig. 3 (d) is the final result of object tracking.

We have also analyzed the performace of the proposed algorithm compared with the other existing algorithms. The performance have been compared in terms of accuracy, background variation, object flexibility, occlusion, and error accumulation.



(a) Initial deformed template (b) Result of estimation

(c) Probability map for object refinement (d) Final result of tracking

Fig. 3. Procedures for tracking a car

Fig. 4 shows car sequence with occlusion. As shown in Fig. 4, the tracking result is robust against various occlusions.

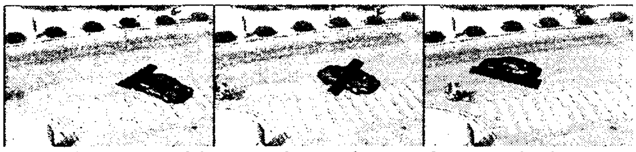
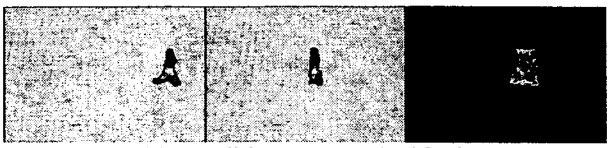


Fig. 4. Tracking results against various occlusions

We test the proposed tracking algorithm on various sequences as shown in Fig 5(a)~(c). In Fig. 5 (a), the sequence has a complex background variation, but the proposed algorithm shows a good tracking performace.

(a) air wolf helicopter with background variation

(b) a car turning left at the crossing corners



(c) a walking man on the block

Fig. 3. Tracking results for various sequences

Table 1 summarizes the performance comparison in term of accuracy, background variation, object flexibility, occlusion, and error accumulation.

Table 1. Performance comparison

	The proposed method	N. Paragios's method	Deformable template
Accuracy	Good	Very good	Poor
Tracking with background variation	Succeed	Fail	Succeed
Tracking with object flexibility	Succeed	Succeed	Partially succeed
Tracking with occlusion	Succeed	Fail	Succeed
Error accumulation	Relatively more	None	Relatively less

4. Conclusion

In this paper, we propose a tracking algorithm which is robust against background variation, object flexibility, and occlusion. We design a potential difference energy function used in the estimation and refinement of object boundary. We also use the affine transform and Level Set theory to improve the peformance.

Experimental results show that the proposed tracking algorithm is more effective than other existing methods in terms of accuracy, background variation, object flexibility, occlusion, and error accumulation.

Future research includes the improvement of execution time and solving the problem of template re-initialization.

Acknowledgement

This work was supported by grant No. R01-2000-00281 from the Basic Research Program of the Korea Science & Engineering Foundation.

References

[1] M. Kass, A. Witkin, D. Terzopoulos, "Snakes : Active Contour Models," *International Journal of Computer Vision*, Vol. 1, pp.321-332, 1987.

[2] N. Paragios, R. Deriche, "Geodesic Active Contour and Level Set for the Detection and Tracking of Moving Objects," *IEEE Transaction on Pattern Analysis and Machine Intellegence*, Vol. 22, no. 3, pp.296-280, 2000.

[3] Y. Zhong, A. K. Jain, M. -P. Dubuisson-Jolly, "Tracking Using Defomable Template," *IEEE Transaction on Pattern Analysis and Machine Intellegence*, Vol. 22, no. 5, pp.554-549, 2000.

[4] M. –P, Dubuisson-Jolly, A. Gupta, "Tracking Deformable Templates Using a Shortest Path Algorithm," *Computer Vsion and Image Understanding*, Vol. 81, no. 1, pp.26-45, 2001.

[5] D. K. Park, H. S. Yoon, C. S. Won, "Fast Obect Tracking in Digital Video," *IEEE Transaction on Comsumer Electronics*, Vol. 46, no. 3, pp.785-790, 2000.

[6] J. M. Odobez, P. Bouthemy, "Robust Multiresolution Estimation of Parametric Motion Models," *Jounrnal of Visual Communication and Image Representation*, Vol. 6, no. 4, pp.348-365, 1995.