AN ALGORITHM FOR CENTROID-BASED TRACKING OF MOVING OBJECTS *

Jacinto C. Nascimento
IST
Lisboa, Portugal

Arnaldo J. Abrantes
ISEL
Lisboa, Portugal

Jorge S. Marques †
IST/ISR
Lisboa, Portugal

Abstract

This article addresses the problem of tracking moving objects using deformable models. A Kalman-based algorithm is presented, inspired on a new class of constrained clustering methods, recently proposed by Abrantes and Marques in the context of static shape estimation. A set of data centroids is tracked using intra-frame and inter-frame recursions. Centroids are computed as weighted sums of the edge points belonging to the object boundary. The use of centroids introduces competitive learning mechanisms in the tracking algorithm leading to improved robustness with respect to occlusion and contour sliding. Experimental results with traffic sequences are provided.

1. INTRODUCTION

Video segmentation is an instrumental operation for dynamic scene analysis. It is not easy to detect the objects in a scene and to estimate their motion, under general hypothesis. In some applications the problem is easier: the initial position of the object is known, e.g., defined by its external boundary, and the goal is to track the object motion in the next frames. This problem is interesting because the image of a moving object suffers time-varying deformations caused by the object motion and by occlusion. Besides, the object is often cluttered by a textured background.

Deformable models have been used to address this problem. Since the seminal work of Kass *et al.* [7], where the *snakes* were used for lip tracking, deformable models have been extensively used in many tracking applications, for instance biomedical image-analysis [8], extraction of facial features [13], or the analysis of traffic scenes [11]. A recent trend consists of addressing the problem in a probabilistic context. This allows to incorporate the available knowledge into separate models: the shape/motion and the sensor models.

It is often assumed that the object boundary is a curve belonging to a set of admissible shapes. Several classes of models have been proposed, e.g., point models [7], B-splines [9] and Fourier series [12]. The class of admissible shapes is often too general, being necessary to bound the number of shape variation modes, to reduce the degrees of freedom. This goal can be achieved in a number of ways, e.g., by restricting the shape to be a linear combination of templates defined by the user or estimated from the data (e.g., using eigen shapes [5]).

The motion model is a key feature in the tracker performance. It allows to predict the object position and velocity in future images and restricts the trajectories of the object boundary parameters: the evolution of motion and shape parameters must be constrained by assigning a high cost to unusual trajectories. This can be done by using stochastic difference equations. The estimation of shape and motion is then converted into a state estimation problem addressed by Kalman or non-linear filtering [13, 3].

The design of the sensor model is also a key feature to the sucess of a tracking algorithm. The shape model is attracted by the data features detected in the image (e.g., edge points). Two strategies are usually adopted in the literature to describe the data/model interaction: the assignment of model points to data features using a matching algorithm (explicit methods) or the use of a potencial function generating a force field (implicit methods). In this paper a third approach will be adopted based on a fuzzy classification of the data features using competitive learning. This is achieved by employing a unified framework recently proposed by Abrantes and Marques [1].

This paper extends the unified framework developed in the context of static shape analysis, to the problem of object tracking in dynamical scenes. The algorithm proposed in this paper exhibits good tracking capabilities and improved robustness with respect to incomplete data and outliers. The paper is organized as follows: section 2 describes the shape/motion dynamical models; section 3 addresses the observation (sensor) model; section 4 presents a tracking algorithm based on a Kalman filtering using the previous models; section 5 shows some experimental results, and section 6 concludes the paper.

2. SHAPE AND MOTION REPRESENTATION

Given a sequence of images $I_1, \ldots, I_t$, we wish to estimate the boundary of a moving object. In this paper, the object boundary will be approximated by a parametric curve defined as a weighted sum of basis functions $b_1, \ldots, b_N$, i.e.,

$$z(s) = \sum_{k=1}^{N} z_k b_k(s) \qquad z(s) \in \Re^2 \quad (1)$$

where s is a parameter defining the location of a point z on the curve and $z_1, \ldots, z_N$, $z_i \in \Re^2$ is a set of 2D vec-

*this work was partially supported by PRAXIS XXI

†contact author: IST, Av. Rovisco Pais, Torre Norte 1096 Lisboa, Portugal

tors which control the model shape. The basis functions are chosen by the user. Equation (1) can be written in a compact way as

$$z(s) = B(s)Z \tag{2}$$

where $B(s) = [b_1(s), \ldots, b_N(s)]$ is a $1 \times N$ row vector and $Z = (X, Y)$ is a $N \times 2$ matrix of coefficients. The object boundary dedined in (1) belongs to a vector space with finite dimension $2N$. Since the object is moving, the matrix Z is allowed to vary. This dependency becomes more explicit rewritting (2) as $z(t, s) = B(s)Z(t)$. A dynamical model is adopted in the sequel to represent the evolution of Z. We will assume that the columns X, Y of Z are independent random processes. Furthermore it is considered that $X(t)$ is defined by a stochastic difference equations

$$x(t) = Ax(t-1) + w(t) \tag{3}$$

where $x(t) = (X^T(t), \dot{X}^T(t))^T$ is a state vector containing X and its derivative , A is a dynamic matrix and $w(t)$ is a white noise processes with Gaussian distribution $\mathcal{N}(0, Q)$. Equation (3) defines a stochastic model for the evolution of the object boundary through time. A similar equation is used for the second coordinate $Y(t)$.

3. OBSERVATION MODEL

In order to track a moving object, a set of visual features are extracted from the video sequence. A common approach consists of sampling the model contour (e.g., a *B-spline*), and for each sample point seeking for the highest image gradient point lying inside a search window (e.g., using a strip band orthogonal to the model contour [4]). This is a low-complexity method, well suited for real-time tracking applications. Unfortunately this measurement has two major drawbacks: it is very sensitive to false-alarm detection and it usually produces significant contour sliding due to the aperture problem [6].

A different approach to compute the visual features was proposed in [1], based on an unified data clustering framework containing several well known methods as special cases: snakes, c-means, fuzzy c-means, elastic nets and Kohonen maps. A method belonging to this unified framework, instead of explicity associating an image feature to each model point, selects a large set of candidate features from the image, (e.g, all the edges points) and attempts to associate them (in a fuzzy way) to the model points, obtained by sampling the contour model. The fuzzy partition of the feature space depends on the method being used. Denoting by $\vartheta_n(p)$ the weight degree of membership of edge point $p \in \Re^2$ to the k-th model sample, $z(s_n)$, it becomes natural to associate z_n with a visual feature, defined as the centroid of the k-th fuzzy region, i.e.,

$$\xi_n = \frac{\sum_p p\vartheta_n(p)}{\mu_n} \tag{4}$$

where

$$\mu_n = \sum_p \vartheta_n(p) \tag{5}$$

measures the amount of data in the k-th region. The external force applied on each model point is defined by

$$f_{ext}(z_n) = \mu_n(\xi_n - z(s_n)) \tag{6}$$

which can be interpreted as a force produced by a zero-length spring with stiffness μ_n, coupling the model sample, $z(s_n)$, to the corresponding centroid, ξ_n, (see figure 1).

The choice of the weighting functions $\vartheta_n(p)$ is a key issue in the performance of the algorithm. They are obtained by the minimization of a cost function often used in clustering algorithms (see [1] for details). It should be stressed that in shape analysis the scope of the weighting functions must be bounded to avoid long range forces namely, the attraction of the contour points by edges detected inside the object or in the background. These are undesirable forces which have to be cancelled. To achieve this goal, $\vartheta_n(p)$ is multiplied by an appropriate window, going to zero when $|p - z_n| \to \infty$. A Gaussian function will be used for this purpose, leading to modified weights

$$\tilde{\vartheta}_n(p) = \vartheta_n(p) \exp\left(-\frac{|z_n - p|^2}{2\sigma_\epsilon^2} \right) \tag{7}$$

where σ_ϵ is a scale parameter. Alternative methods using a noise model can be found in [2].

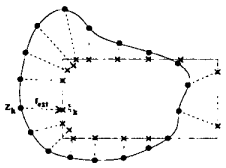

Figure 1: Spring forces applied on each point model.

It is important to observe that μ_n and ξ_k are iteratively estimated and in general they depend on the whole set of model samples. This is a consequence of the competitive learning embeded in the computation of ϑ_n. The potencial energy of the stretched spring is given by

$$P_n = \frac{1}{2}\mu_n \left| \xi_n - B(s_n)Z \right|^2 \tag{8}$$

Adopting a Gibbsian approach, the quadratic energy (8) leads to a Gaussian sensor distribution.

$$p(\xi_n|Z) = \mathcal{N}\Big(B(s_n)Z, \mu_n^{-1}I\Big) \tag{9}$$

Therefore, the centroids are represented by

$$\xi(s_n, t) = C_n Z(t) + v_n(t) \tag{10}$$

where $C_n = [B(s_n)\ 0]$ and $v_n(t)$ is a white noise with Gaussian distribution $\mathcal{N}(0, \mu_n^{-1}I)$.

Several methods can be used to define the weight $\vartheta_n(p)$ inspired in constrained clustering algorithms. Most of them introduce competitive learning mechanisms since the model points compete to represent each image feature. An exception is the snake algorithms which can also be expressed in terms of centroids. In general, the use of centroids computed with competitive learning increases the tolerance to outliers and reduces the sliding effect. Figure 2a,b shows the centroids obtained with and without competitive learning using the fuzzy c-means and the snakes. As we can

see the fuzzy c-means algorithm leads to better motion estimates. This improvement becomes more clear when the model is iteratively updated during the measurement process as discussed in section 4.

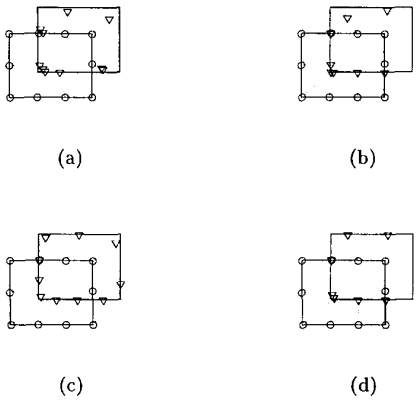

Figure 2: Data centroids after the 1st (a, b) and the 5th (c, d) iterations using (a, c) fuzzy c-means and (b, d) snake sensor models. ○ - model units, ▽ - centroids.

In this paper, we shall adopt the weights of the fuzzy c-means algorithm. In this case,

$$\vartheta_n(p) = \left(\sum_j \left(\frac{|z(s_n) - p|^2}{|z(s_j) - p|^2} \right)^{\frac{1}{f-1}} \right)^{-f} \qquad (11)$$

where f is a fuzziness parameter.

4. TRACKING ALGORITHM

Given the state model with dynamic equation (3) and observation equation (10), the state vector can be estimated from the observed images using two independent Kalman filters to estimate $x(t)$, $y(t)$, respectively. Kalman filtering is based on two steps:

- a prediction step which predicts the state vector and error covariance at time t, knowing the observations until the instant $t-1$.
- a filtering step which updates the predicted values based on the observation at instant t.

The evaluation of the observation (data centroids) is not trivial since it depends on the shape estimates: better shape estimates provide more accurate centroids. This suggests a recursive measurement process. At each instant of time, we recursively compute a set of centroids associated with curve samples $\hat{z}(s_1), \ldots, \hat{z}(s_N)$ obtained from the current shape estimate. These estimates updated during the measurement process until all the centroids converge to steady locations. Convergence is usually achieved after a small number of iterations (typically less than 5). Figure 2c,d shows the centroids obtained after 5 iterations using the weighting functions of snakes and fuzzy c-means algorithms. Comparing with the results obtained in figure 2, after the first iteration, a significant improvement is observed in the case of fuzzy c-means method. No motion model is used during the measurement iteration since this is an intra-frame recursion.

To enhance the robustness of the tracker it is often convenient to restrict the class of admissible shapes. A simple way to achieve this goal is by considering the object as an affine transform of a reference shape, defined by the user. A method to incorporate this information in the estimation process is the *persistent template* algorithm described in [4].

The tracking algorithm proposed in this paper is detailed in table 1; $\hat{x}(t)$, $P(t)$ denote the state estimate and the covariance matrix of the estimation error at the t-th image, and x_{aux}, P_{aux} are auxiliary variables used for centroid refinement in the measurement loop.

Kalman Prediction:
$$\hat{x}^-(t) = A\hat{x}(t-1)$$
$$P^-(t) = AP(t-1)A^T + Q$$
Loop1: measurement loop
$x_{aux} \leftarrow \hat{x}^-(t)$
$P_{aux} \leftarrow P^-(t)$
Loop2: for all sampling points $x(s_n)$
Compute $\xi(s_n) = (\xi_x(s_n), \xi_y(s_n))$ by (4)
Filtering step
$K \leftarrow P_{aux}C_n^T \left[\; C_nP_{aux}C_n^T + \mu_n^{-1} \; \right]^{-1}$
$x_{aux} \leftarrow x_{aux} + K \left(\; \xi_x(s_n) \; - \; C_nx_{aux} \; \right)$
$P_{aux} \leftarrow \left[\; I - KC_n \; \right] P_{aux}$
EndLoop2
State updating using persistent template [4]
EndLoop1
$\hat{x}(t) = x_{aux}$
$P(t) = P_{aux}$

Table 1: Tracking Algorithm.

5. EXPERIMENTAL RESULTS

The proposed algorithm was evaluated with real images and succeed to follow moving objects in cluttered backgrounds. Figure 3 shows a segment of a traffic sequence exhibiting velocity changes and partial occlusions caused by the trees. The difficulty of the problem can be observed in the edge images displayed in figure 5. It is stressed that edge points are the only features used for tracking in this method. The tracking results (see the black curves in figure 3) show the ability of the proposed algorithm to deal with incomplete data and significant pose and velocity changes which occur during the manoeuvre of the car.

When the centroids are evaluated without competitive learning, the control points tend to cluster in regions with higher density of data, due to the sliding effect. This leeds

to poorer representations of the object shape as can be observed in the Figure 4.

Figure 3: Tracking results using fuzzy c-means algorithm (frames 13, 19, 23, 39).

Figure 4: Tracking results using snakes algorithm (frames 13, 19, 23, 39).

6. CONCLUSIONS

This paper studies the estimation of non rigid shapes in image sequences using a Kalman-based tracking algorithm with a novel type of observations. These observations are computed using the Fuzzy c-means algorithm with isotropic limitation of attraction regions. Although the fuzzy c-means algorithm has been recently used by other authors for tracking [10], the algorithm proposed in this paper is derive from an unified framework described in [1] which allows an easy extension to other types of methods (e.g., Kohonen maps, elastic nets). Image features (edges points) are converted

Figure 5: Sequence of edges used in tracking.

into a set of centroids computed with appropriate weighting functions. Therefore, the user may change the performance of the algorithm by simply replacing the set of weighting functions by alternative ones. Experimental results with traffic sequences allow to conclude that the proposed algorithm is able to deal with object tracking in cluttered backgrounds as well as partial occlusion of the object boundaries, exhibiting good tracking capabilities. Further improvement is expected by using more efficient representations of the trajectories of the motion and shape parameters and by restricting these parameters to apropriate manifolds. The choice of a vector space of affine shapes adopted in this paper is just a first step towards this purpose.

7. REFERENCES

[1] A. J. Abrantes, J. S. Marques, "A Class of Constrained Clustering Algorithms for Object Boundary Extraction" *IEEE Trans. Image Processing*, Outubro 1996.

[2] A. J. Abrantes, J. S. Marques, "Pattern Recognition Methods for Object Boundary Detection". In *British Machine Vision Conference*, 1998.

[3] A. Blake, M. Isard. "Contour Tracking by Stochastic Propagation of Conditional Density". In *Proc. European Conference on Computer Vision*, 342-356, 1996.

[4] A. Blake, R. Curwen, A. Zisserman, "A Framework for Spatio-Temporal Control Tracking of Visual Contours", in *Real-Time Computer Vision*, C. Brown, D. Terzopoulos, eds., Cambridge University Press, 1994.

[5] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, "Active Shape Models - Their Training and Aplication", CVIU Vol 61., No 1, 38-59, 1995.

[6] B. Klaus and P. Horn, "Robot Vision". MIT Press, 1986.

[7] M. Kass, A. Witkin, e D. Terzopoulos, "Snakes: Active Contour Models", in *Proceedings, First Int. Conf. on Computer Vision*, 259-268, 1987.

[8] T. McInerney, D. Terzopoulos, "Deformable Models in Medical Image Analysis: A Survey", in *Medical Image Analysis*, 1(2), 1996.

[9] S. Menet, P. Saint-Marc, and G. Medioni. "B-snakes: Implementations and Applications to Stereo". In *Proc. DARPA Image Understanding Workshop*, pages 720-726, 1990.

[10] J. Ohm, P. Ma. Feature-based cluster segmentation of image sequences. Proc. ICIP, 178-181, 1997.

[11] N. Paragios, R. Deriche. "Detecting Multiple Moving Targets using Deformable Contours". In *Proc. IEEE Int. Conf. on Image Processing*, 183-186, Santa Barbara, 1997.

[12] G. L. Scott,"The Alternative Snake-And Other Animals", in *Proceedings, 3rd Alvey Vision Conference, Cambridge*, 341-347, 1987.

[13] D. Terzopoulos, R. Szeliski, "Tracking with Kalman Snakes", in *Active Vision*, A. Blake, A. Yuille, eds., MIT Press, 1992.