A New Energy Term Combining Kalman–Filter and Active Contour Models for Object Tracking

J. Denzler, H. Niemann
Universität Erlangen–Nürnberg
Lehrstuhl für Mustererkennung (Informatik 5)

Abstract. In the past years active contour models have been applied in the field of object tracking. For object tracking a prediction step is necessary, especially when tracking in natural scenes with an imhomogeneous background or for fast moving objects. Thus, in our paper we introduce a new energy term which combines a Kalman–Filter based prediction with an active contour energy description. For this, a new energy term is proposed which can be applied for all prediction steps for which a confidence of the predicted positions is available. We present results which show the improvement due to this new energy term for tracking a moving object in front of an inhomogeneous background and a partial occlusion during the tracking.

Key words: active contour models, tracking, prediction

1. Introduction

In the field of real–time computer vision the so called active contour models (*snakes*, ACMs) have proven to be a promising approach to data driven object tracking [6, 7]. There are different approaches for solving the problem of energy minimization [1, 4, 9]. Most of them have been developed for the analysis of static scenes and segmentation [5]. For the use in image sequences, they have mostly been transferred without any modification. Thus one gets the best results by assuming a homogeneous background and a small displacement of the object. Object tracking using snakes fails in most of the cases, when an heterogeneous background is in the scene, in the case of partial occlusions of the tracked object, and for weak object contours.

For object tracking in image sequences a so called prediction step is needed. Without prediction tracking is only possible for very simple objects, using a homogeneous background and a small displacement of the object between consecutive images. The displacement of course depends on the smoothing of the energy, but there is a trade–off between smoothing and accurate contour extraction. Small objects are likely to disappear if the smoothing is to large.

For ACM a prediction step can be applied in three different ways. First, the prediction can be done independent of the energy minimization. In the following we will call this an *explicit prediction* which means that in a separate prediction step the position of the ACM is predicted. Second, the prediction can be included in the energy minimization by modelling the coherence in time of the snake's new position with the positions observed during the past images. This means that positions in the image which are likely for some snake elements, i.e. which lie on the estimated path of the movement, will get a lower energy. The term *implicit prediction* for this prediction method takes into account that the prediction is implicitly done during the energy minimization. Third, one can combine the *explicit prediction* and *implicit prediction*; in the explicit prediction the positions of the snake elements are predicted and these predicted positions are weighted in the energy term.

Up to now mostly the first of the three prediction steps are mentioned in the literature. For example [2] takes advantage of the normal flow near the snake elements to iteratively predict the movement of the whole contour before before the energy minimization is done. One example which could be classified as an implicit prediction is the Kalman–Snake of [10]. But this approach does not model the coherence in time through an extra energy term for the ACM. The disadvantage of the explicit prediction is that despite of an accurate prediction, missing energy minima can result in errors during the energy minimization (see Sect. 4).

In our approach we show that a combination of implicit and explicit prediction results in an improvement of our real–time tracking system [7]. In the explicit prediction, we use a Kalman–Filter. The estimated positions in the next image are weighted in the energy term of the snake by using the error covariance of the estimations which one gets automatically from the Kalman–Filter approach for each estimation. This can be done without increasing the computational effort.

In the next section the mathematical preliminaries are introduced, both for the ACM and for the Kalman–Filter approach. In Sect. 3 this results in the definition of a new energy term which directly combines an implicit and explicit prediction. In Sect. 4 we will present results for this new approach which compare ACMs for object tracking in the case of no prediction, a pure explicit prediction by a Kalman–Filter and a combination of Kalman–Filter and a new energy term for ACMs. The paper concludes with a discussion of the approach.

2. Mathematical Background

2.1. Active Contour Models

In the following we describe extraction of objects contours by ACMs for static images. Thus we have left out the indices for the time t of the image. An ACM can be described as a parametric function $\boldsymbol{v}(s) = (x_s, y_s)$, $s = 0, 1, \ldots, n-1$, with $x_s \in [0, x_{max}]$, $y_s \in [0, y_{max}]$. Such an ACM has an energy E^* defined by

$$E^* = \sum_{0}^{n-1} \left[E_i(\boldsymbol{v}(s)) + E_f(\boldsymbol{v}(s)) + E_c(\boldsymbol{v}(s)) \right] . \tag{1}$$

In most cases the internal energy E_i is given by

$$E_i(\boldsymbol{v}(s)) = \frac{1}{2} \left(\alpha(s)|\boldsymbol{v}_s(s)|^2 + \beta(s)|\boldsymbol{v}_{ss}(s)|^2 \right) , \tag{2}$$

where $\boldsymbol{v}_s$ and $\boldsymbol{v}_{ss}$ are the first and second derivatives of $\boldsymbol{v}$ with respect to s. The parameters $\alpha(s)$ and $\beta(s)$ describe the stiffness and elasticity of the ACM. E_f describes the forces of the image on the snake and E_c summarizes all the other constraints of the snake, for example, connections of snake elements to image features (spring forces) or the limitation of the distance between the snake elements [9].

The position of the ACM in an image is computed by minimizing the energy (1). For minimization two approaches can be found in the literature. The first approach treats the minimization as a search problem in the 2D image plane [1, 11]. The second approach is based on the variational calculus [9], by iteratively solving the Euler–Lagrange differential equations. In the discrete case the following equations must be solved [9]:

$$\begin{aligned}\alpha(s)(\boldsymbol{v}(s)-\boldsymbol{v}(s-1)) &- \alpha(s+1)(\boldsymbol{v}(s+1)-\boldsymbol{v}(s)) + \\ \beta(s-1)(\boldsymbol{v}(s-2)-2\boldsymbol{v}(s-1)+\boldsymbol{v}(s)) &- 2\beta(s)(\boldsymbol{v}(s-1)-2\boldsymbol{v}(s)+\boldsymbol{v}(s+1)) + \\ \beta(s+1)(\boldsymbol{v}(s)-2\boldsymbol{v}(s+1)+\boldsymbol{v}(s+2)) &+ \left(\frac{\partial(E_f+E_c)}{\partial x_s}, \frac{\partial(E_f+E_c)}{\partial y_s}\right)^T = \mathbf{o}\end{aligned}$$

Let us rewrite this in matrix form for $s = 0 \ldots n-1$:

$$\boldsymbol{A}\boldsymbol{x} + \boldsymbol{f}_x(\boldsymbol{x},\boldsymbol{y}) = \mathbf{o} \quad \text{and} \quad \boldsymbol{A}\boldsymbol{y} + \boldsymbol{f}_y(\boldsymbol{x},\boldsymbol{y}) = \mathbf{o} \tag{3}$$

with

$$\boldsymbol{x} = (x_0, x_1, \ldots, x_{n-1})^T, \boldsymbol{y} = (y_0, y_1, \ldots, y_{n-1})^T,$$

$$\boldsymbol{f}_x(\boldsymbol{x},\boldsymbol{y}) = \left(\left.\frac{\partial(E_f+E_c)}{\partial x_s}\right|_{s=0}, \left.\frac{\partial(E_f+E_c)}{\partial x_s}\right|_{s=1}, \ldots, \left.\frac{\partial(E_f+E_c)}{\partial x_s}\right|_{s=(n-1)} \right)^T,$$

and

$$\boldsymbol{f}_y(\boldsymbol{x},\boldsymbol{y}) = \left(\left.\frac{\partial(E_f+E_c)}{\partial y_s}\right|_{s=0}, \left.\frac{\partial(E_f+E_c)}{\partial y_s}\right|_{s=1}, \ldots, \left.\frac{\partial(E_f+E_c)}{\partial y_s}\right|_{s=(n-1)} \right)^T.$$

For the computation of the unknown vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ an iterative procedure is used that converges if $\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)}$ and $\boldsymbol{y}^{(k)} = \boldsymbol{y}^{(k-1)}$, where $\boldsymbol{x}^{(k)}$ and $\boldsymbol{y}^{(k)}$ are the solutions at iteration step k. Thus, the equations (3) can be written as:

$$\boldsymbol{A}\boldsymbol{x}^{(k)} + \boldsymbol{f}_x(\boldsymbol{x}^{(k-1)},\boldsymbol{y}^{(k-1)}) = \mathbf{o} = \gamma(\boldsymbol{x}^{(k-1)} - \boldsymbol{x}^{(k)}) \tag{4}$$

$$\boldsymbol{A}\boldsymbol{y}^{(k)} + \boldsymbol{f}_y(\boldsymbol{x}^{(k-1)},\boldsymbol{y}^{(k-1)}) = \mathbf{o} = \gamma(\boldsymbol{y}^{(k-1)} - \boldsymbol{y}^{(k)}) \tag{5}$$

with $\gamma \in \mathbb{R}$ being the stepsize, and transformed to:

$$\boldsymbol{x}^{(k)} = (\boldsymbol{A} + \gamma\boldsymbol{I})^{-1}\left(\gamma\boldsymbol{x}^{(k-1)} - \boldsymbol{f}_x(\boldsymbol{x}^{(k-1)},\boldsymbol{y}^{(k-1)})\right) \tag{6}$$

$$\boldsymbol{y}^{(k)} = (\boldsymbol{A} + \gamma\boldsymbol{I})^{-1}\left(\gamma\boldsymbol{y}^{(k-1)} - \boldsymbol{f}_y(\boldsymbol{x}^{(k-1)},\boldsymbol{y}^{(k-1)})\right) \tag{7}$$

Introducing a prediction step within this approach means that an additional energy term must be found which handles the coherence in time of the movement during the tracking. As one can see in equations (6) and (7) the derivative of the energy must be computed. Thus it would be advantageous, if one can find an energy term for which a closed form solution for the derivatives with respect to x_s and y_s exists. Such an energy term will be developed in Sect. 3.

2.2. Kalman–Filter

In [10] a Kalman–Snake has been introduced. We describe in the following a more simple and thus computationally less expensive dynamic system for the motion of the snake elements in the 2D image plane. The snake elements are uncoupled. For this reason, only n 3×3 matrices have to be inverted, instead of one $3n \times 3n$ matrix.

Now for time varying images we have to add the time parameter to the snake model. Thus, $x_{s,t}$ and $y_{s,t}$ means the position of the s–th snake element in x– and y–direction at time t. The dynamic system for the motion without distortion is then formulated by (see [3])

$$x_{s,t+1} = x_{s,t} + h\dot{x}_{s,t} + \frac{h^2}{2}\ddot{x}_{s,t}, \quad \dot{x}_{s,t+1} = \dot{x}_{s,t} + h\ddot{x}_{s,t}, \quad \ddot{x}_{s,t+1} = \ddot{x}_{s,t} \tag{8}$$

$$y_{s,t+1} = y_{s,t} + h\dot{y}_{s,t} + \frac{h^2}{2}\ddot{y}_{s,t}, \quad \dot{y}_{s,t+1} = \dot{y}_{s,t} + h\ddot{y}_{s,t}, \quad \ddot{y}_{s,t+1} = \ddot{y}_{s,t}, \tag{9}$$

with $s = 0 \ldots n-1$, which means that we assume constant acceleration which is distorted by the model noise ${}^m\boldsymbol{n}_t = (0, 0, {}^m n_{3,t}, \ldots, 0, 0, {}^m n_{3n+3,t})^T \in \mathbb{R}^{6n}$ (see equation (12)), where the left upper index m denotes that the noise is related to the model. Only the positions of the snake elements can be observed, thus we get for the observation equations

$$z_{s,t} = x_{s,t} + {}^o n_{s,t}, \quad z_{n+s,t} = y_{s,t} + {}^o n_{n+s,t} \tag{10}$$

where ${}^o\boldsymbol{n}_{s,t} = ({}^o n_{0,t}, {}^o n_{1,t}, \ldots, {}^o n_{2n-1,t})^T$ is the observation noise, where the left upper index o stands for observation. The complete description of the system can be summarized as follows:

$$\boldsymbol{x}_t = (x_{0,t}, \dot{x}_{0,t}, \ddot{x}_{0,t}, x_{1,t}, \dot{x}_{1,t}, \ldots, \ddot{x}_{n-1,t}, y_{0,t}, \dot{y}_{0,t}, \ddot{y}_{0,t}, y_{1,t}, \ldots, \ddot{y}_{n-1,t})^T \tag{11}$$

$$\boldsymbol{x}_{t+1} = \boldsymbol{B}\boldsymbol{x}_t + {}^m\boldsymbol{n}_t \qquad \text{with } \boldsymbol{E}[{}^m\boldsymbol{n}_t {}^m\boldsymbol{n}_t^T] = {}^m\boldsymbol{N}_t \tag{12}$$

$$\boldsymbol{z}_t = \boldsymbol{C}\boldsymbol{x}_t + {}^o\boldsymbol{n}_t \qquad \text{with } \boldsymbol{E}[{}^o\boldsymbol{n}_t {}^o\boldsymbol{n}_t^T] = {}^o\boldsymbol{N}_t \tag{13}$$

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{B}_h & \ldots & \ldots & \mathbf{o} \\ \mathbf{o} & \boldsymbol{B}_h & \ldots & \mathbf{o} \\ \vdots & & \ddots & \vdots \\ \mathbf{o} & \ldots & \ldots & \boldsymbol{B}_h \end{pmatrix} \in \mathbb{R}^{3n \times 3n} \text{ with } \boldsymbol{B}_h = \begin{pmatrix} 1 & h & h^2/2 \\ 0 & 1 & h \\ 0 & 0 & 1 \end{pmatrix} \tag{14}$$

$$\boldsymbol{C} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & \ldots & \ldots & \ldots & 1 & 0 & 0 \end{pmatrix} \tag{15}$$

In these equations (11) is the state vector, (12) the state transition equation, (13) the observation equation, and the matrices (14) and (15) the transition and observation matrix, respectively.

Without going into detail, the following recursive update equations can be computed [8], starting with an initial state estimation the error covariance matrix of which is $\boldsymbol{P}_0^{(-)}$:

- error covariance matrix extrapolation $\boldsymbol{P}_t^{(-)}$

$$\boldsymbol{P}_t^{(-)} = \boldsymbol{B}\boldsymbol{P}_{t-1}^{(+)}\boldsymbol{B}^T + {}^m\boldsymbol{N}_t \tag{16}$$

- Kalman–Gain matrix $\boldsymbol{K}_t$

$$\boldsymbol{K}_t = \boldsymbol{P}_t^{(-)}\boldsymbol{C}^T\left[\boldsymbol{C}\boldsymbol{P}_t^{(-)}\boldsymbol{C}^T + {}^o\boldsymbol{N}_t\right]^{-1} \tag{17}$$

- error covariance matrix update $\boldsymbol{P}_t^{(+)}$

$$\boldsymbol{P}_t^{(+)} = \left[\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{C}\right]\boldsymbol{P}_t^{(-)} \tag{18}$$

We have left out the complete state estimation equations in this brief overview of the theory. The important part of this approach for our new energy term is the error covariance matrix $\boldsymbol{P}_t^{(+)}$ (equation (18)) which gives a measure for the confidence of the snake elements' estimated positions. This is exactly the measure which is needed in the energy, as it will be described in the next section.

Instead of equations (11) to (15) any other system description is possible. We have also tested a model description in which a common displacement in the image plane for all snake elements have been introduced, to satisfy the constraint that contour elements of the moving object have a common motion direction. Then the system cannot be uncoupled and due to the increased computational effort our real–time experiments showed a worse result compared to the model description in equation (8).

3. Introduction of a new Energy Term

In Sect. 1 we have motivated the advantage of a combination of implicit and explicit prediction. Let us now assume that we get a prediction for the position $\boldsymbol{v}_p$ of the ACM in one image as a result of an explicit prediction step. Again we have left out the time index t. Such a prediction step is in our case a prediction by the Kalman–Filter (Sect. 2.2).

The following energy term for an implicit prediction takes into account the results of the explicit prediction step. To be more precise, we recursively calculate the error covariance $\boldsymbol{P}^{(+)}$ (compare equation (18)) of the predicted positions. On the one hand, if the error covariance is small, the prediction is rather accurate; on the other hand, a large covariance means, that the predicted positions are not very reliable. An energy term $E_p(\boldsymbol{v}^{(k)}, \boldsymbol{v}_p)$, depending on the actual position $\boldsymbol{v}^{(k)}$ at iteration step k and the predicted position $\boldsymbol{v}_p$ is shown in (19). It shows the demanded behavior, i.e. small energy on a predicted position, if this position is very reliable, and large energy, if the position is unreliable,

$$E_p(\boldsymbol{v}^{(k)}, \boldsymbol{v}_p) = -\frac{1}{|2\pi \boldsymbol{P}^{(+)}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{v}^{(k)} - \boldsymbol{v}_p)^T \left(\boldsymbol{P}^{(+)}\right)^{-1} (\boldsymbol{v}^{(k)} - \boldsymbol{v}_p)\right) \tag{19}$$

The advantage of this term is that there exists a closed form of ∇E_p which is required in the Euler–Lagrange differential equations (3).

$$\nabla E_p(\boldsymbol{v}^{(k)}, \boldsymbol{v}_p) = \frac{\partial E_p}{\partial \boldsymbol{v}^{(k)}} = -E_p(\boldsymbol{v}^{(k)}, \boldsymbol{v}_p) \left(\boldsymbol{P}^{(+)}\right)^{-1} (\boldsymbol{v}^{(k)} - \boldsymbol{v}_p) \tag{20}$$

Thus the equations which must be solved (compare equations (4) and (5)), become

$$\boldsymbol{A}\boldsymbol{v}^{(k)} + E_p(\boldsymbol{v}^{(k)}, \boldsymbol{v}_p) \left(\boldsymbol{P}^{(+)}\right)^{-1} (\boldsymbol{v}^{(k)} - \boldsymbol{v}_p) + \nabla \boldsymbol{f}(\boldsymbol{v}^{(k)}) = \mathbf{o} \tag{21}$$

The principle of this new energy term (19) is clarified in Figure 1. Without the new

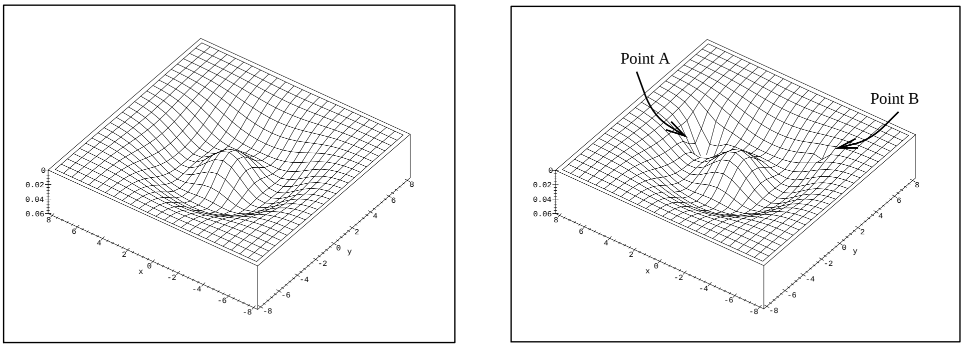

Figure 1. Left: external energy of the contour of a circle without prediction. Right: influence of the new energy term. Predicted positions of snake elements are weighted with respect to their confidence (point A, strong confidence; point B, weak confidence).

energy term, the external energy which mainly determines the object's contour, looks like Figure 1, left. One can see that the predicted position of the explicit prediction has no influence on the shape of the ACM. If there is a strong background edge near a weak object's contour, the snake elements will move toward the strong background edge.

In Figure 1, right, the predicted positions are weighted in the external energy by the confidence of the prediction which is described by the error covariance matrix $\boldsymbol{P}_t(+)$. The snake–element on the left side (point A) has been strongly weighted due to a small error covariance, while the prediction of the element on the right side (point B) has only a small influence on the external energy. Thus, this element can move into the true minimum, while the first element will remain on its predicted position, even if the object's contour is weak or if there is a strong background edge near the object.

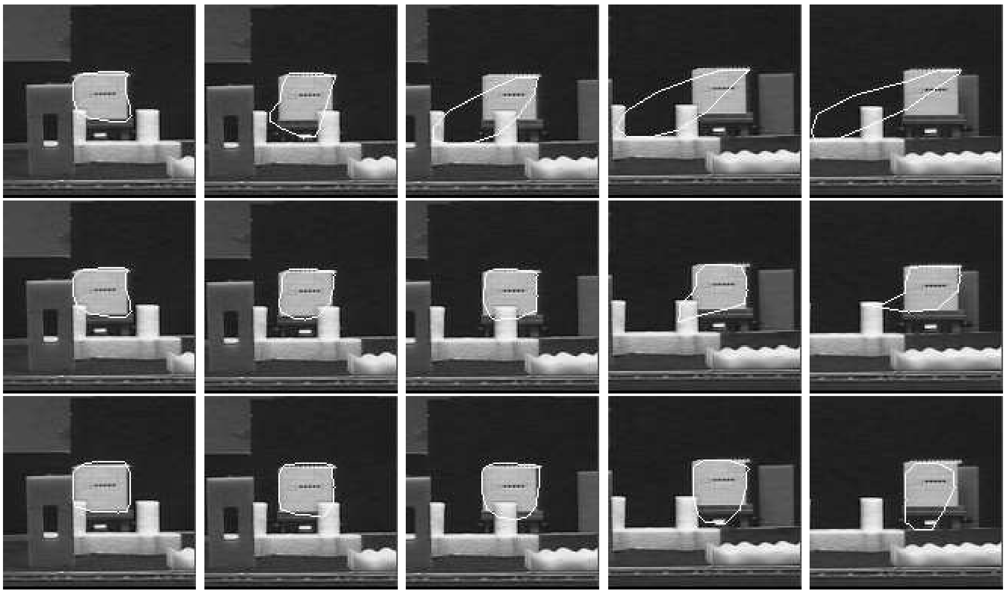

Figure 2. Images 420, 430, 440, 450, 460 of a real–time experiment. First row: without prediction. Second row: prediction using a 2D motion model and a Kalman–Filter (see Sect. 2.2). Third row: prediction in combination with the new energy term, described in Sect. 3.

4. Experiments and Results

We have used an experimental setup described earlier in [7]. A toy train is moving in front of a robot on which a camera is mounted. The movement of the robot is calculated by matching the center of the snake with the center of the image. We have chosen a light object in front of an inhomogeneous background (see Figure 2). We have tested the quality of tracking using snakes for three different approaches: without prediction (Figure 2, first row), with an explicit prediction step, based on the a 2D motion description in the image plane (Figure 2, second row), and with an combination of implicit and explicit prediction based on the new energy term in the ACM (Figure 2, third row).

Using the standard variational approach without any prediction (Figure 2, first row) the strong background edges near the moving object cause an error in the contour extraction. Using the explicit prediction step, described in Sect. 2.2 the tracking can be improved. But after the partial occlusion of the object, the ACM is caught by the occluding background object (Figure 2, second row). This behavior can be eliminated by introducing the new energy term described in Sect. 3. In Figure 2, third row, the prediction which has a strong confidence after 40 successfully tracked images, is strongly weighted and thus the ACM does not lose the object contour.

5. Conclusion

In this paper we have developed a new approach for combining Kalman–Filter and ACMs in the context of object tracking. For object tracking with ACMs a prediction step is essential. Three different methods for including a prediction step have been introduced. For the special case of a prediction by Kalman–Filter, we have proposed an energy term which handles the confidence in the predicted positions within the energy of the snake. The confidence which can be described by the error covariance matrix is one value which is returned by the iterative Kalman–Filter equations. Thus, for any model of the object's motion which can be described within the Kalman–Filter approach, a mixture of implicit and explicit prediction can be done using the proposed energy term in the ACM energy. One advantage of our approach is that a closed form solution of the derivative of this energy term with respect to the snake elements exists which is needed in the variational calculus to iteratively solve the Euler–Lagrange differential equations. Another advantage is the small additional computational effort compared to other purely implicit and explicit prediction steps, presented earlier in the literature

Our experiments showed for a 2D movement of the snake elements that the performance in real–time object tracking can be increased, even for partial occlusions and strong background edges near the object.

References

[1] A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(9):855–867, 1990.

[2] M. Berger. Tracking rigid and non polyhedral objects in an image sequence. In *Scandinavian Conference on Image Analysis*, pages 945–952, Tromso (Norway), 1993.

[3] C.K. Chui and G. Chen. *Kalman Filtering*. Springer Verlag, Berlin, Heidelberg, 1990.

[4] L.D. Cohen and I. Cohen. Finite-element method for active contour models and balloons for 2–D and 3–D images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, 1993.

[5] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models — their training and applications. *Computer Vision Graphics and Image Processing*, 61(1):38–59, 1995.

[6] R. Curwen and A. Blake. Dynamic contours: Real–time active splines. In A. Blake and A. Yuille, editors, *Active Vision*, pages 39–58. MIT Press, Cambridge, Massachusetts, London, England, 1992.

[7] J. Denzler and H. Niemann. Combination of simple vision modules for robust real–time motion tracking. *European Transactions on Telecommunications*, 5(3):275–286, 1995.

[8] A. Gelb, editor. *Applied Optimal Estimation*. The MIT Press, Cambridge, Massachusetts, 1979.

[9] M. Kass, A. Wittkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 2(3):321–331, 1988.

[10] D. Terzopoulos and R. Szeliski. Tracking with Kalman snakes. In A. Blake and A. Yuille, editors, *Active Vision*, pages 3–20. MIT Press, Cambridge, Massachusetts, London, England, 1992.

[11] D.J. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *Computer Vision Graphics and Image Processing*, 55(1):14–26, 1992.