Published as
"Tracking with Kalman snakes", D. Terzopoulos and R. Szeliski,
in *Active Vision*, A. Blake and A. Yuille (eds.), MIT Press, Cambridge,
MA, 1992, Chapter 1, Pages 3–20.

1 Tracking with Kalman Snakes

Demetri Terzopoulos and Richard Szeliski

Snakes (a.k.a. active contour models) are deformable contours that have been used in many image analysis applications, including the image-based tracking of rigid and nonrigid objects. Deformable contours and their multidimensional generalizations conform to object shapes and motions by numerically integrating differential equations of Lagrangian elastodynamics. The snake equations of motion provide flexible tracking mechanisms that are driven by simulated forces derived from time-varying images.

This chapter first reviews the physically motivated formulation of snake models. It then proposes a probabilistic interpretation of the approach that leads to optimal estimation as a means of extracting reliable information from noisy observations. For the purposes of real-time tracking, it is essential that the estimation proceed sequentially as new observations become available. This is the premise of Kalman filtering theory. We show how to construct continuous Kalman filters that incorporate the dynamic snake into their system and prior models. These *Kalman snakes* are a promising technique for image-based tracking of rigid and, especially, nonrigid objects; they forge a link between the physical and probabilistic modeling approaches to active vision.

Snakes were introduced in [21, 40]. The idea of edge and curve detection by optimizing models of curve contrast and smoothness can be traced back to [33] and [26]. The snake model generalizes this notion through the use of elastodynamic models and applied forces. This chapter builds upon the dynamical systems point of view and its connections to estimation theory. For simplicity we consider the original snake model formulated using controlled-continuity variational splines. There exist several noteworthy variants, however, including snakes based on the Fourier [36, 37], B-spline [29, 22, 11] (also see Chapter 4), finite element [9, 8], and discrete representations [41, 6].[1] Snakes and their variants have been applied to static images requiring edge, curve, and boundary detection, as well as region segmentation and skeletonization [21, 43, 16, 17, 12, 1, 34]. Since snakes conform readily to complex biological structures, many applications have been in the area of biomedical

[1] Chapter 3 considers parameterized models akin to snakes, known as deformable templates [25].

image interpretation [2, 24, 9, 22, 4, 8, 6].[2]

The idea of tracking objects in time-varying images using snakes was originally proposed in [21], where it was applied to track a speaker's lips through dynamic forces generated by the images. A more thorough tracking of articulate facial features using multiple snakes is described in [41]. In [23], closed snakes are applied to track amorphously deforming living cells that locomote using pseudopods. In [7, 11], real-time snakes are used to track the occluding contours of 3D objects as seen from a camera attached to a moving robot arm. These applications have demonstrated that snakes are very well suited to tracking rigid and nonrigid objects. As it tracks an object of interest, a snake can provide detailed quantitative information about its position, velocity, acceleration, and its evolving shape in the image plane.

Another popular approach to tracking, particulary in the context of active vision, is based on Kalman filtering theory [18] (see also Chapters 5, 6, 8, 18, and 19). The Kalman filtering approach was first applied to the incremental estimation of rigid object motion [20, 5, 35] and to the tracking of sparse rigid features such as points and lines [15, 28]. More recently, the approach has been extended to more complex representations, such as the depth maps obtained when a camera moves through a static scene [27].

Dynamic deformable models, such as snakes, can serve as system models for Kalman filter trackers [39, 30, 31]. The resulting shape and motion estimators are able to deal very effectively with the complex motions of nonrigid objects, because deformable models are governed by the principles of nonrigid dynamics. This chapter formulates image-plane Kalman filter trackers using snakes, while Chapter 6 considers the more general case of three dimensional tracking using 3D deformable models that are related to snakes.

1.1 Snakes: Dynamics and Tracking

Snakes are planar deformable contours that move under the influence of image forces. We can define a deformable contour by constructing a suitable deformation energy $\mathcal{E}_s(\mathbf{v})$, where $\mathbf{v}$ represents the contour as a mapping from the unit parametric domain $s \in [0, 1]$ into the image

[2] Deformable contours are also applicable to problems unrelated to vision [14, 13].

plane $\Re^2$.[3] The components of the mapping $\mathbf{v}(s) = (x(s), y(s))$ are the contour's coordinate functions. It is convenient to think of the external forces on the contour as deriving from a potential $\mathcal{P}$. A snake is a deformable contour that minimizes the energy

$$\mathcal{E}(\mathbf{v}) = \mathcal{E}_s(\mathbf{v}) + \mathcal{P}(\mathbf{v}). \tag{1.1}$$

For a simple (linear) snake, the internal deformation energy is

$$\mathcal{E}_s = \int_0^1 w_1(s)|\mathbf{v}_s|^2 + w_2(s)|\mathbf{v}_{ss}|^2 \, ds, \tag{1.2}$$

where the subscripts on $\mathbf{v}$ denote differentiation with respect to s. The energy models the deformation of a stretchy, flexible contour $\mathbf{v}(s)$ and includes two physical parameter functions: $w_1(s)$ controls the "tension" and $w_2(s)$ controls the "rigidity" of the contour. These functions are useful for manipulating the physical behavior and local continuity of the model. In particular, setting $w_1(s_0) = w_2(s_0) = 0$ permits a position discontinuity and setting $w_2(s_0) = 0$ permits a tangent discontinuity to occur at s_0.

To apply snakes to images $I(x, y)$, we specify external potentials $\mathcal{P}$ which attract the snake to intensity extrema, edges, or other perhaps more complex features of interest, depending on the application. We define

$$\mathcal{P}(\mathbf{v}) = \int_0^1 P(\mathbf{v}(s)) \, ds, \tag{1.3}$$

where $P(x, y)$ is a scalar potential function defined over the image plane and which is typically computed through image processing.[4] It is natural to interpret the local minima of P as snake "attractors." The snake will have an affinity for darkness or brightness if $P(x, y) = \pm c[G_\sigma * I(x, y)]$, depending on the sign, and for intensity edges if $P(x, y) = -c|\nabla[G_\sigma * I(x, y)]|$, where c controls the magnitude of the potential. $G_\sigma * I$ denotes the image convolved with a (Gaussian) smoothing filter whose characteristic width σ controls the spatial extent of the attractive

[3] For the purposes of this paper, we can think of an image intensity function as being defined over all of $\Re^2$, albeit with a value of zero outside some finite bounded region.
[4] It makes sense to compute $P(x, y)$ at all image points only if dedicated hardware is available that can do so in real time. According to (1.3), however, P need only be computed along the deformable contour $\mathbf{v}(s)$, which presents an opportunity for improving efficiency when such hardware is unavailable (see Chapter 4).

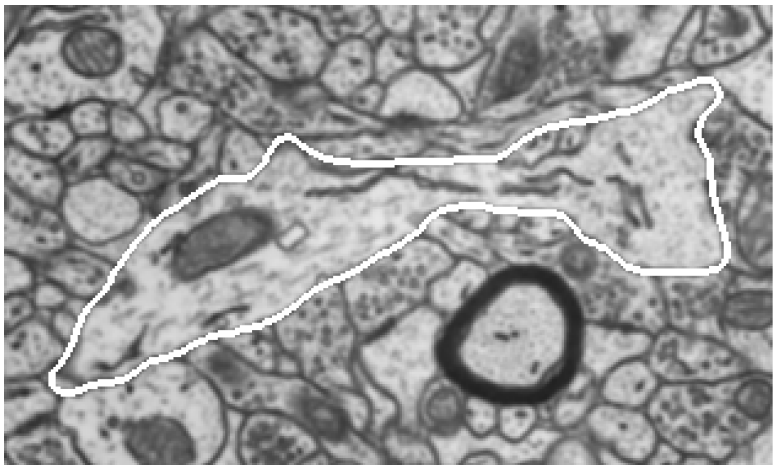


Figure 1.1
Snake attracted to cell membrane in an EM photomicrograph.

"depression" of P. [21] discusses some of the motivations and issues surrounding the smoothing operation.

Figure 1.1 shows a snake, the closed white contour, which has been attracted by the dark membrane of a cell in an EM photomicrograph (see [6] for details). In the figure, $P = c[G_\sigma * I]$ where I is the EM image and σ is about 1 pixel.

1.1.1 Lagrangian Dynamics

It is natural to view energy minimization as a static problem. More generally, however, one can construct a dynamical system and allow it to arrive at a minimal energy state as it achieves equilibrium. Suitable dynamic models with intuitively appealing physical behaviors may be derived by applying the principles of Lagrangian mechanics. This approach leads to dynamic snakes which offer a variety of interesting possibilities that are not necessarily evident from the static energy minimization point of view. For example, a dynamic model may be guided by an adaptive control system or interactively by the user as it minimizes its energy. To visualize a dynamic snake in an image potential, think of P as slippery terrain and the deformable contour as having a tendency to slide downhill into the depressions in the terrain, conforming to their shapes as it does so. In Figure 1.1, for example, the snake has

slid into the ravine associated with the dark membrane that surrounds the large cell. The snake minimizes the total energy $\mathcal{E}$ when it attains the equilibrium state shown in the figure.

The Lagrangian formulation of snakes was first presented in [40]. We can represent a dynamic snake by introducing a time-varying mapping $\mathbf{v}(s,t)$ and a kinetic energy $\int_0^1 \mu|\mathbf{v}_t|^2\,ds$, where $\mu(s)$ is the mass density, and the subscript t denotes a time derivative. We combine the kinetic energy and the deformation potential energy functional $\mathcal{E}(\mathbf{v})$ to define the Lagrangian

$$\mathcal{L}(\mathbf{v}) = \frac{1}{2}\int_0^1 \mu|\mathbf{v}_t|^2\,ds - \frac{1}{2}\mathcal{E}(\mathbf{v}). \tag{1.4}$$

If the initial and final configurations are $\mathbf{v}(s,t_0)$ and $\mathbf{v}(s,t_1)$, then Hamilton's principle dictates that the deformable model's motion $\mathbf{v}(s,t)$ from $t=t_0$ to $t=t_1$ is such that the action integral $\int_{t_0}^{t_1}\mathcal{L}(\mathbf{v})\,dt$ is stationary, which implies that its variation with respect to $\mathbf{v}$ vanishes:

$$\delta_{\mathbf{v}}\left(\frac{1}{2}\int_{t_0}^{t_1}\int_0^1 \mu|\mathbf{v}_t|^2 - w_1(s)|\mathbf{v}_s|^2 - w_2(s)|\mathbf{v}_{ss}|^2 - P(\mathbf{v})\,ds\,dt\right) = \mathbf{0}. \tag{1.5}$$

The condition leads to Lagrange's equations of motion for the model [10]. Once set in motion, a snake with a mass distribution will move perpetually, unless kinetic energy is dissipated. To dampen the snake so that it can achieve static equilibrium, we incorporate the Rayleigh dissipation functional $\mathcal{D}(\mathbf{v}_t) = \frac{1}{2}\int_0^1 \gamma|\mathbf{v}_t|^2\,ds$ where $\gamma(s)$ is the damping density. Evaluating the appropriate variational derivatives of the integrands L in (1.5) and D in the dissipation functional, the Lagrange equations are

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \mathbf{v}_t}\right) + \frac{\partial D}{\partial \mathbf{v}_t} - \frac{\partial L}{\partial \mathbf{v}} + \frac{\partial}{\partial s}\left(\frac{\partial L}{\partial \mathbf{v}_s}\right) - \frac{\partial}{\partial s^2}\left(\frac{\partial L}{\partial \mathbf{v}_{ss}}\right) = 0. \tag{1.6}$$

Assuming constant mass density $\mu(s)=\mu$ and constant dissipation $\gamma(s)=\gamma$, the equations of motion for the snake may be written as

$$\mu\mathbf{v}_{tt} + \gamma\mathbf{v}_t - \frac{\partial}{\partial s}\left(w_1\mathbf{v}_s\right) + \frac{\partial^2}{\partial s^2}\left(w_2\mathbf{v}_{ss}\right) = -\nabla P(\mathbf{v}(s,t)) \tag{1.7}$$

with appropriate initial and boundary conditions. This can be interpreted as a force balance relationship. On the left hand side are inertial, damping, stretching, and bending forces. These forces balance the negative gradient of the potential on the right hand side, which may be

interpreted physically as generalized external forces coupling the snake to the image data.

1.1.2 Force-Based Tracking

The equation of motion (1.7) suggests a straightforward force-based mechanism for tracking dynamic data where the model maintains dynamic equilibrium in a time-varying potential [21].

If the image varies as a function of time, the potential $P(x, y, t)$ will also be time-varying. If the potential $P(x, y, t)$ changes after a snake has achieved static equilibrium, potential energy is converted to kinetic energy and the snake will move nonrigidly to achieve a new equilibrium. Thinking in terms of the force balance relationship in (1.7), the model will track the dynamic data as it attempts to maintain the generalized forces $-\nabla P$ in dynamic equilibrium against the inertial, damping, and deformation forces.

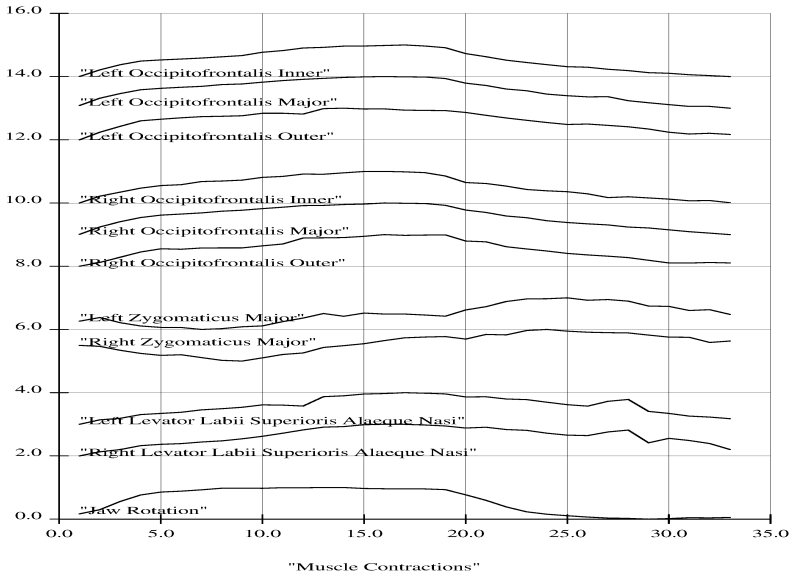
One way to visualize the tracking scheme on a discrete, frame-by-frame basis is to imagine once again the potential energy surface $P(x, y, t_k)$ for a given image frame k as a slippery terrain. Suppose that a deformable contour is at rest at the bottom of an extended depression in the terrain. The next frame $k + 1$ will induce a potential $P(x, y, t_{k+1})$ which is a perturbation of $P(x, y, t_k)$ such that the depression will have shifted. If the shift is not too great, the snake will now find itself somewhere up on a slope and will slide downhill into the shifted depression, moving (and deforming) appropriately as it does so. In this way, are convected over the image by the moving depressions of the potential surface.

Figure 1.2 illustrates the force based tracking procedure. At the top are three frames from an image sequence of a surprise expression. In the middle are the negative potential functions $-P = c|\nabla[G_\sigma * I]|$. The snakes track the nonrigidly moving image features, making it possible to glean useful information about the face. For example, it is possible to derive dynamic estimates of facial muscle contractions shown in the plotted traces (see [41] for details).

In Section 1.3, we will identify force driven tracking as a special case of a general framework for sequential estimation of dynamic data which is capable of dealing with uncertainties in image data and modeling them in an optimal way.

(a)

(b)



(c)

Figure 1.2
Tracking facial features using snakes. (a) Three frames from an image sequence with snakes (black contours) tracking hairline, eyebrows, nose, mouth, and chin. (b) Snakes (white contours) on negative potential function. (c) Estimated facial muscle contractions plotted as time series.

1.1.3 Discretization and Numerical Simulation

In order to numerically compute a minimal energy solution, it is necessary to discretize the energy $\mathcal{E}(\mathbf{v})$. A general approach to discretizing energies $\mathcal{E}(\mathbf{v})$ is to represent the function of interest $\mathbf{v}$ in approximate form as a linear superposition of basis functions weighted by *nodal variables* $\mathbf{u}_i$. The nodal variables may be collected into a vector $\mathbf{u}$ to be computed. The local-support polynomial basis functions prescribed by the finite element method are convenient for most applications. An alternative to the finite element method is to apply the finite difference method to the continuous Euler equations, such as (1.7), associated with the model.

The discrete form of quadratic energies such as (1.1) may be written as

$$E(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} + \mathrm{P}(\mathbf{u}), \tag{1.8}$$

where $\mathbf{K}$ is called the *stiffness matrix*, and $\mathrm{P}(\mathbf{u})$ is the discrete version of the external potential. The minimum energy (equilibrium) solution can be found by setting the gradient of (1.8) to $\mathbf{0}$, which is equivalent to solving the set of algebraic equations

$$\mathbf{K}\mathbf{u} = -\nabla\mathrm{P} = \mathbf{f}, \tag{1.9}$$

where $\mathbf{f}$ is the generalized external force vector.

Finite elements and finite differences generate local discretizations of the continuous snake model, hence the stiffness matrix will have a sparse and banded structure. To illustrate the discretization process, suppose we apply the finite difference method to discretize the energy (1.2) on a set of nodes $\mathbf{u}_i = \mathbf{v}(ih)$ for $i = 0, \ldots, N-1$ where $h = 1/(N-1)$. Suppose we use the finite differences $\mathbf{v}_s \approx (\mathbf{u}_{i+1} - \mathbf{u}_i)/h$ and $\mathbf{v}_{ss} \approx (\mathbf{u}_{i+1} - 2\mathbf{u}_i + \mathbf{u}_{i-1})/h^2$. For cyclic boundary conditions (i.e., a closed contour), we obtain the following symmetric pentadiagonal matrix (unspecified entries are 0):

$$\mathbf{K} = \begin{bmatrix} a_0 & b_0 & c_0 & & & & c_{N-2} & b_{N-1} \\ b_0 & a_1 & b_1 & c_1 & & & & c_{N-1} \\ c_0 & b_1 & a_2 & b_2 & c_2 & & & \\ & c_1 & b_2 & a_3 & b_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & c_{N-5} & b_{N-4} & a_{N-3} & b_{N-3} & c_{N-3} \\ c_{N-2} & & & & c_{N-4} & b_{N-3} & a_{N-2} & b_{N-2} \\ b_{N-1} & c_{N-1} & & & & c_{N-3} & b_{N-2} & a_{N-1} \end{bmatrix}, \quad (1.10)$$

where

$$a_i = (w_{1i-1} + w_{1i})/h^2 + (w_{2i-1} + 4w_{2i} + w_{2i+1})/h^4, \quad (1.11)$$

$$b_i = -w_{1i}/h^2 - 2(w_{2i} + w_{2i+1})/h^4, \quad (1.12)$$

$$c_i = w_{2i+1}/h^4, \quad (1.13)$$

assuming that $w_{1i} = w_1(ih)$ and $w_{2i} = w_2(ih)$ are sampled at the same nodes. All indices in these expressions are interpreted modulo N.

The discretized version of the Lagrangian dynamics equation (1.7) may be written as a set of second order ordinary differential equations for $\mathbf{u}(t)$:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}, \quad (1.14)$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is a damping matrix; both matrices are typically also sparse and banded. In the simple case of a finite difference discretization where we assume that mass is lumped at the nodes, $\mathbf{M}$ and $\mathbf{C}$ will be diagonal matrices. Note that for static $\mathbf{f}$, the dynamic equilibrium condition $\ddot{\mathbf{u}} = \dot{\mathbf{u}} = \mathbf{0}$ leads to the static solution (1.9), as expected.

To simulate the snake dynamics, this system of ordinary differential equations must be integrated forward through time. The finite element literature offers several suitable explicit and implicit direct integration methods, including the central difference, Houbolt, Newmark, or Wilson methods [3]. We can illustrate the basic idea with a semi-implicit Euler method that takes time steps Δt. We replace the time derivatives of $\mathbf{u}$ with the backward finite differences $\ddot{\mathbf{u}} \approx (\mathbf{u}^{(t+\Delta t)} - 2\mathbf{u}^{(t)} + \mathbf{u}^{(t-\Delta t)})/(\Delta t)^2$, and $\dot{\mathbf{u}} \approx (\mathbf{u}^{(t+\Delta t)} - \mathbf{u}^{(t-\Delta t)})/2\Delta t$, where the superscripts denote the quantity evaluated at the time given in parentheses. This yields the update formula

$$\mathbf{A}\mathbf{u}^{(t+\Delta t)} = \mathbf{b}^{(t)}, \tag{1.15}$$

where $\mathbf{A} = \mathbf{M}/(\Delta t)^2 + \mathbf{C}/2\Delta t + \mathbf{K}$ is a pentadiagonal matrix and $\mathbf{b} = (2\mathbf{M}/(\Delta t)^2)\mathbf{u}^{(t)} - (\mathbf{M}/(\Delta t)^2 - \mathbf{C}/2\Delta t)\mathbf{u}^{(t-1)} + \mathbf{f}^{(t)}$.

The pentadiagonal system can be solved very efficiently ($O(N)$ complexity) by factorizing $\mathbf{A}$ into lower and upper triangular matrices, then solving the two resulting sparse triangular systems. We compute the unique normalized factorization $\mathbf{A} = \mathbf{LDU}$ where $\mathbf{L}$ is a lower triangular matrix, $\mathbf{D}$ is a diagonal matrix, and $\mathbf{U} = \mathbf{L}^T$ is an upper triangular matrix [3]. The solution $\mathbf{u}^{(t+\Delta t)}$ to (1.15) is obtained by first solving $\mathbf{Ls} = \mathbf{b}^{(t)}$ by forward substitution, then $\mathbf{Uu} = \mathbf{D}^{-1}\mathbf{s}$ by backward substitution. For the linear snakes described above, only a single factorization is necessary, since $\mathbf{A}$ is constant. Note that the LDU factorization and forward/backward substitutions are inherently sequential, recursive operations.

Researchers have investigated alternative approaches to numerically simulating snake models, including dynamic programming [1] and greedy [42] algorithms.

1.2 Snakes and Bayesian Estimation

The dynamic response of snakes to forces computed from images addresses the problem of inferring the image-based shapes and motions of rigid or nonrigid objects. An alternative to this physical point of view is to cast the inference task in a probabilistic framework and to view it as an estimation problem. Snake models may then be interpreted in terms of Bayesian estimation, where the posterior distribution $p(\mathbf{u}|\mathbf{d})$ of the unknown quantity $\mathbf{u}$ conditioned on the data $\mathbf{d}$ is computed using Bayes' rule

$$p(\mathbf{u}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{u})\, p(\mathbf{u})}{p(\mathbf{d})} \tag{1.16}$$

with the normalizing denominator

$$p(\mathbf{d}) = \sum_{\mathbf{u}} p(\mathbf{d}|\mathbf{u}).$$

The *prior model* $p(\mathbf{u})$ is a probabilistic description of the state we are trying to estimate before any sensor data is collected. The *sensor model* $p(\mathbf{d}|\mathbf{u})$ is a description of the noisy or stochastic processes that relate the

original (unknown) state $\mathbf{u}$ to the sampled input image or sensor values $\mathbf{d}$. Bayes' rule combines these two probabilistic models to form a *posterior model* $p(\mathbf{u}|\mathbf{d})$ which describes probabilistically the best estimate of $\mathbf{u}$ given the data $\mathbf{d}$ [32].

The snake serves as a prior model of the shapes and nonrigid motions of features of interest in the image. Intuitively, it should yield prior distributions that bias Bayesian estimates towards low energy configurations as measured by the deformation energy $\mathcal{E}_s$ in (1.2). The trick is to convert an energy which elastically restores the deformable model to its natural shape into a prior distribution over expected shapes, with lower energy shapes being the more likely. This is done conveniently using a Gibbs (or Boltzmann) distribution of the form

$$p(\mathbf{u}) = \frac{1}{Z_{\mathrm{p}}} \exp(-E_{\mathrm{p}}(\mathbf{u})), \tag{1.17}$$

where $E_{\mathrm{p}}(\mathbf{u})$ is a discrete version of $\mathcal{E}_s$, and Z_{p} is a normalizing constant (called the partition function). When $E_{\mathrm{p}}(\mathbf{u})$ stems from a finite element or finite difference discretization, it can be written as a sum of local energy terms, and the distribution (1.17) reduces to a Markov random field. Since, according to (1.8), $E_{\mathrm{p}}(\mathbf{u})$ is a quadratic energy of the form

$$E_{\mathrm{p}}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u}, \tag{1.18}$$

the prior distribution is a correlated zero-mean Gaussian with a covariance matrix $\mathbf{P} = \mathbf{K}^{-1}$. Although, $\mathbf{K}$ is sparse and banded, $\mathbf{P}$ will not be.

The external potential applied to the snake is equivalent to a sensor model. We can see this by examining a linear sensor model involving Gaussian uncertainty. For simplicity, consider the physical situation depicted in Figure 1.3, where a snake is constrained by a set of zero-length "springs" with stiffnesses c_i. The springs couple the snake to "nails" $\mathbf{d}_i$ in the plane which represent data points with independent additive Gaussian noise. If spring i is attached to the snake at s_i, it is stretched by an amount $|\mathbf{v}(s_i) - \mathbf{d}_i|$. Hence the continuous potential energy functional of all the stretched springs is

$$\mathcal{P}(\mathbf{v}) = \sum_i c_i[\mathbf{v}(s_i) - \mathbf{d}_i]^2. \tag{1.19}$$

The discrete version of this potential may be written as the quadratic form

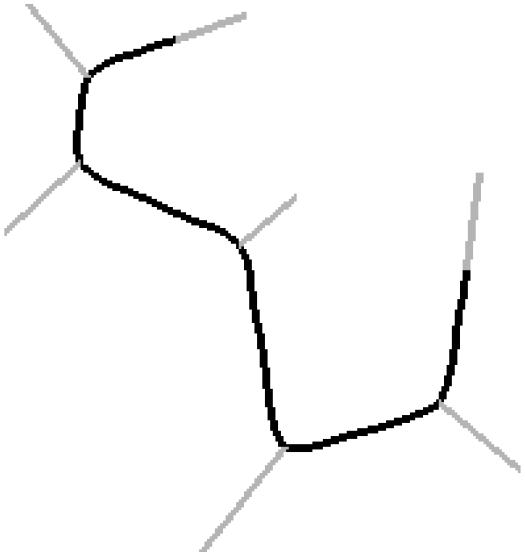


Figure 1.3
Snake (black) constrained with spring (grey) forces.

$$\mathrm{P}(\mathbf{u}) = E_{\mathrm{d}}(\mathbf{u}, \mathbf{d}) = \frac{1}{2}(\mathbf{H}\mathbf{u} - \mathbf{d})^T \mathbf{R}^{-1}(\mathbf{H}\mathbf{u} - \mathbf{d}). \tag{1.20}$$

The associated spring forces (cf. (1.9)) are expressed by the linear form

$$-\nabla \mathrm{P} = \mathbf{f} = \mathbf{H}^T \mathbf{R}^{-1}(\mathbf{d} - \mathbf{H}\mathbf{u}). \tag{1.21}$$

Here, $\mathbf{H}$, known as the interpolation or measurement matrix, maps nodal variables to the nails which are interpreted as measurement data $\mathbf{d}$.[5] [6] The entries of $\mathbf{R}^{-1}$ are spring stiffnesses c_i. Using the Gibbs distribution once again, we arrive at the sensor model

[5] In general, (e.g., for nonlinear snake models) the measurements $\mathbf{d}$ may be related to the state variables $\mathbf{u}$ through a non-linear function $\mathbf{d} = \mathbf{h}(\mathbf{u})$. The *extended Kalman filter* formulation is applicable in this case, although the resulting estimator may be sub-optimal [18]. The extended Kalman filter formulation is necessary for the 3D physical models described in Chapter 6.

[6] Note that if the dimensionality of $\mathbf{d}$ is smaller than that of $\mathbf{u}$, the snake will approximate the data using the deformation energy as a smoothness constraint to constrain the extra degrees of freedom. On the other hand, if the dimensionality of $\mathbf{d}$ is greater than that of $\mathbf{u}$, the snake will provide a least squares fit to the data. Both cases are handled by the same measurement equation.

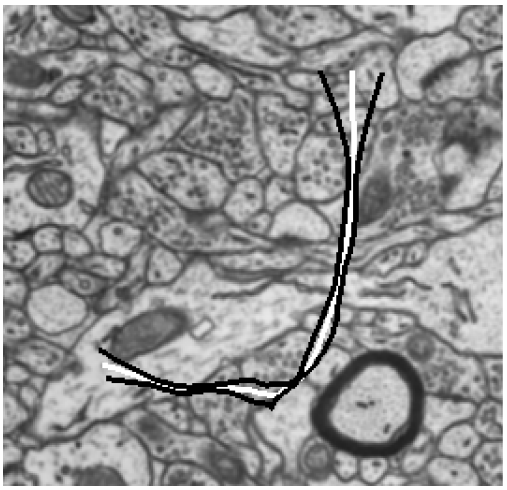


Figure 1.4
Confidence envelope of snake estimate. The snake is shown in white, and the confidence envelope in black.

$$p(\mathbf{d}|\mathbf{u}) = \frac{1}{Z_{\mathrm{d}}} \exp(-E_{\mathrm{d}}(\mathbf{u}, \mathbf{d})), \tag{1.22}$$

This implies that the sensor distribution is a correlated Gaussian with covariance matrix $\mathbf{R}^{-1}$. Hence, if each datapoint has independent noise with standard deviation σ_i^2, the optimal spring stiffnesses for estimation are $c_i = 1/\sigma_i^2$.

Combining the prior (1.17) and the sensor (1.22) models using Bayes' rule, we obtain the posterior distribution

$$p(\mathbf{u}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{u})\, p(\mathbf{u})}{p(\mathbf{d})} = \frac{1}{Z} \exp(-E(\mathbf{u})), \tag{1.23}$$

where

$$E(\mathbf{u}) = E_{\mathrm{p}}(\mathbf{u}) + E_{\mathrm{d}}(\mathbf{u}, \mathbf{d}). \tag{1.24}$$

Note that this is the same energy equation as (1.8), which describes the energy of a discrete snake. Thus, computing the *maximum a posteriori* (MAP) estimate [19], i.e., the value of $\mathbf{u}$ that maximizes the conditional probability $p(\mathbf{u}|\mathbf{d})$, provides the same result as finding the minimum energy configuration of the snake.

Although in this case both the physical and the Bayesian models may be used to produce the same solution, there are several advantages to the

probabilistic approach [38]. Most importantly, the external force fields (data constraints) can be derived in a principled fashion taking into account the known noise characteristics of the sensors and the uncertainty in the posterior model can be quantified and used by higher level stages of processing.

As an example of quantifying the uncertainty of a snake, we can compute its in equilibrium by generating random samples from the posterior distribution and accumulating the desired statistics. In general, generating good random samples can be tricky [38]. However, for a snake with spring constraints, the posterior energy is quadratic

$$E(\mathbf{u}) = \frac{1}{2}(\mathbf{u} - \mathbf{u}^*)^T \mathbf{A}(\mathbf{u} - \mathbf{u}^*) + k, \tag{1.25}$$

where $\mathbf{u}^* = \mathbf{A}^{-1}\mathbf{b}$ is the minimum energy solution. The Gibbs distribution (1.23) corresponding to this quadratic form is a multivariate Gaussian with mean $\mathbf{u}^*$ and covariance $\mathbf{A}^{-1}$. It is straightforward to generate an unbiased random sample because the discrete snake energy factorizes into LDU form. Substituting $\mathbf{u} = \mathbf{L}^T\mathbf{v}$ into (1.25), we obtain

$$E(\mathbf{v}) = \frac{1}{2}(\mathbf{v} - \mathbf{v}^*)^T \mathbf{D}(\mathbf{v} - \mathbf{v}^*) + k, \tag{1.26}$$

where $\mathbf{v}^* = \mathbf{D}^{-1}\mathbf{L}^{-1}\mathbf{b}$ is the intermediate solution in the LDU solution of the banded snake system $\mathbf{L}\mathbf{D}\mathbf{L}^T\mathbf{u} = \mathbf{b}$. Thus, to generate a random sample, we simply add white Gaussian noise with variance $\mathbf{D}^{-1}$ to $\mathbf{v}^*$ and continue with the solution for $\mathbf{u}$. The resulting collection of random snakes are used to compute the local variance at each point on the snake, and hence a confidence envelope (Figure 1.4).

1.3 Sequential Estimation and Kalman Snakes

The probabilistic interpretation of snakes allows us to generalize and ameliorate the force-based tracking scheme. It enables us to design, in a principled way, sequential estimation algorithms which integrate visual measurements over time to improve the accuracy of estimates. Such sequential estimation algorithms become even more potent when they are combined with the equations of motion of a physical model. The resulting estimation algorithm is known as the continuous Kalman filter [18].

1.3.1 The Kalman Filter

The Kalman filter is formulated by adding a *system model* to the prior and sensor models of the Bayesian formulation. The system model describes the expected evolution of the vector of state variables $\mathbf{u}(t)$ over time.[7] The continuous Kalman filter assumes the system model

$$\frac{d}{dt}\mathbf{u} = \mathbf{F}\mathbf{u} + \mathbf{q}, \qquad \mathbf{q} \sim N(\mathbf{0}, \mathbf{Q}), \tag{1.27}$$

where $\mathbf{F}$ is the system matrix and $\mathbf{q}$ is a white Gaussian noise process with covariance $\mathbf{Q}$. The system noise is used to model unknown disturbances or uncertainty about the true system dynamics.

The sensor model component $p(\mathbf{d}|\mathbf{u})$ of the Bayesian formulation is rewritten as

$$\mathbf{d} = \mathbf{H}\mathbf{u} + \mathbf{r}, \qquad \mathbf{r} \sim N(\mathbf{0}, \mathbf{R}), \tag{1.28}$$

where each measurement is assumed to be corrupted by a Gaussian noise vector $\mathbf{r}$ whose covariance $\mathbf{R}$ is known.

The Kalman filter operates by continuously updating an estimated state vector $\hat{\mathbf{u}}$ and an error covariance matrix $\mathbf{P}$. The state estimate equation

$$\dot{\hat{\mathbf{u}}} = \mathbf{F}\hat{\mathbf{u}} + \mathbf{S}^{-1}\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{d} - \mathbf{H}\hat{\mathbf{u}}) \tag{1.29}$$

consists of two terms. The first term predicts the estimate using the system model, while the second term updates the estimate using the residual error $(\mathbf{d} - \mathbf{H}\mathbf{u})$ weighted by the *Kalman filter gain matrix* $\mathbf{G} = \mathbf{S}^{-1}\mathbf{H}^T\mathbf{R}^{-1}$, where $\mathbf{S} = \mathbf{P}^{-1}$ is the inverse covariance (or *information matrix*) of the current estimate. The size of the Kalman gain depends on the relative sizes of the $\mathbf{S}$ and the noise measurement covariance $\mathbf{R}$. As long as the measurements are relatively accurate compared to the state estimate, the Kalman gain is high and new data measurements are weighted heavily. Once the system has stabilized, the state estimate covariance becomes smaller than the measurement noise, and the Kalman filter gain is reduced.

The information matrix $\mathbf{S}$ is updated over time using the *matrix Riccati equation* expressed in terms of inverse covariance

[7] In the remainder of this paper, we will assume that all quantities are continuous functions of time, and we will omit (t).

$$\dot{\mathbf{S}} = -\mathbf{SF} - \mathbf{F}^T\mathbf{S} - \mathbf{SQS} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}. \tag{1.30}$$

This equation is derived from the standard matrix Riccati equation [18, p. 122] using simple matrix algebra.[8] Here again, we see the competing influences of the system and measurement noise processes. As long as the measurement noise $\mathbf{R}$ is small or the Kalman filter gain $\mathbf{G}$ is high, the information (or certainty) $\mathbf{S}$ will continue to increase. As the system begins to reach steady state, the relative influence of the system noise $\mathbf{Q}$, which decreases the certainty, and the measurement inverse covariance $\mathbf{R}^{-1}$, which increases it, counterbalance each other. Of course the absence of new measurements or sudden bursts of new or accurate information can cause fluctuations in this certainty.

1.3.2 The Kalman Snake

We create Kalman snakes by employing the snake equations of motion (1.14) as the system model of a continuous Kalman filter. To this end, we write these equations in the standard dynamical system form

$$\frac{d\mathbf{u}}{dt} = \mathbf{Fu} + \mathbf{g} \tag{1.33}$$

which corresponds to the system model (1.27) of the Kalman filter. For a massless snake (i.e., for $\mathbf{M} = \mathbf{0}$), (1.14) reduces to the first-order system

$$\frac{d}{dt}\mathbf{u} = -\mathbf{C}^{-1}\mathbf{Ku} + \mathbf{C}^{-1}\mathbf{f} \tag{1.34}$$

which is in standard form. To express the second-order system (1.14) similarly, we include the nodal velocities $d\mathbf{u}/dt = \dot{\mathbf{u}}$ as explicit state variables along with the positions $\mathbf{u}$ to obtain the set of coupled first order equations

---

[8] To convert the standard Riccati equation

$$\begin{aligned}\dot{\mathbf{P}} &= \mathbf{FP} + \mathbf{PF}^T + \mathbf{Q} - \mathbf{GRG}^T \\ &= \mathbf{FP} + \mathbf{PF}^T + \mathbf{Q} - \mathbf{PH}^T\mathbf{R}^{-1}\mathbf{HP}\end{aligned} \tag{1.31}$$

into the inverse covariance form, we use the lemma

$$\dot{\mathbf{S}} = -\mathbf{S}\dot{\mathbf{P}}\mathbf{S} \tag{1.32}$$

which can easily be derived from the identity $\mathbf{SP} = \mathbf{I}$,

$$\frac{d}{dt}(\mathbf{SP}) = \dot{\mathbf{S}}\mathbf{P} + \mathbf{S}\dot{\mathbf{P}} = \mathbf{0}.$$

Substituting (1.31) into (1.32), we obtain the desired result (1.30).

$$\frac{d}{dt}\begin{bmatrix}\dot{\mathbf{u}}\\ \mathbf{u}\end{bmatrix} = \begin{bmatrix}-\mathbf{M}^{-1}\mathbf{C} & -\mathbf{M}^{-1}\mathbf{K}\\ \mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{u}}\\ \mathbf{u}\end{bmatrix} + \begin{bmatrix}\mathbf{M}^{-1}\mathbf{f}\\ \mathbf{0}\end{bmatrix}. \tag{1.35}$$

We can then rewrite (1.35) in the standard form (1.33) by simply renaming the augmented state vector $[\dot{\mathbf{u}}^T \quad \mathbf{u}^T]^T$ to $\mathbf{u}$.

We can relate the second term in (1.29) back to the physical snake model, thus arriving at a physical interpretation of the Kalman filter. The residual error $(\mathbf{d} - \mathbf{H}\hat{\mathbf{u}})$ can be interpreted as the deformations of springs coupling selected state variables $\mathbf{Hu}$ to the data $\mathbf{d}$, the matrix $\mathbf{R}^{-1}$ contains the spring stiffnesses (inversely proportional to the variances in the measurement noise), and $\mathbf{H}^T$ converts the spring forces to generalized forces that can then be applied directly to the state variables of the model. The general idea is very similar to the simple force-based tracking scheme described in Section 1.1.2, but there is one significant difference. Before applying the generalized spring forces to the state variables of the model, the Kalman snake transforms them with the covariance matrix $\mathbf{S}^{-1} = \mathbf{P}$ which has accumulated the history of prior observations, their uncertainties, etc., through the evolving matrix Riccatti equation. A simple way of describing the relationship is that force-based tracking amounts to "Kalman filtering" with a constant unit covariance/information matrix.

Both the first-order (1.34) and second-order (1.35) system models may be employed in Kalman snake trackers, but their behaviors will differ. Suppose a Kalman snake is tracking an object that is translating across the image and the object suddenly becomes temporarily occluded. The first-order Kalman snake, although simpler to compute, lacks an inertial term and it will therefore stop moving as soon as data forces vanish because of the occlusion. When the object reappears, it may have moved too far away for the first-order snake to recapture it. The inertial term of the second-order Kalman snake, however, will enable it to continue moving for a distance which will depend on the magnitude of the damping term, and it may stand a better chance of regaining its lock the object when it reappears.

It is worth explaining why we prefer the inverse covariance formulation of the matrix Riccati equation $\mathbf{S}$ rather than the more commonly used covariance formulation involving $\mathbf{P}$. This has to do with the nature of the prior distributions that arise from the snake model's elastic energy. As we showed in Section 1.2, the prior distribution is a multivariate

Gaussian with a covariance matrix $\mathbf{P}(0) = \mathbf{K}^{-1}$, the inverse stiffness matrix. While $\mathbf{K}$ is sparse and banded, $\mathbf{P}(0)$ (and hence $\mathbf{P}(t)$) will not. For snake models with many nodal variables, it is impractical to store and update the dense covariance matrix.

1.3.3 A Simplified Kalman Snake

We can derive a more convenient approximation to the Kalman snake equations if we assume that the inverse covariance matrix $\mathbf{S}$ can be partitioned into a time-invariant internal stiffness component $\mathbf{K}_s$ and a time-varying diagonal component $\mathbf{S}'$

$$\mathbf{S}(t) = \mathbf{K}_s + \mathbf{S}'(t). \tag{1.36}$$

We can set $\mathbf{K}_s = \mathbf{K}$, the snake stiffness matrix. We then apply the Riccati equation (1.30) directly to $\mathbf{S}'$, and ignore any off-diagonal terms that arise. The state update equation (1.29) becomes

$$\dot{\hat{\mathbf{u}}} = \mathbf{F}\hat{\mathbf{u}} + (\mathbf{K}_s + \mathbf{S}')^{-1}\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{d} - \mathbf{H}\hat{\mathbf{u}}) \tag{1.37}$$

The state update equation (1.37) changes the current state estimate according to both the dynamics of the system described by $\mathbf{F}$ and according to the filtered difference between the sampled state $\mathbf{Hu}$ and the data values $\mathbf{d}$ (these can be replaced by other external forces). The Kalman gain contains a weighting component $\mathbf{R}^{-1}$ which is inversely proportional to the noise in the new measurements, a weighting term $\mathbf{S}'$ which varies over time and represents the current (local) certainty in the estimate, and a spatial smoothing component corresponding to the internal stiffness matrix $\mathbf{K}_s$. Note that we do not explicitly compute $\mathbf{G}$. Instead, we solve the system of equations $(\mathbf{K}_s + \mathbf{S}')\tilde{\mathbf{u}} = \mathbf{H}^T\mathbf{R}^{-1}(\mathbf{d} - \mathbf{Hu})$ for $\tilde{\mathbf{u}}$, and use this as the second term in (1.37).

Equation (1.37) employs the snake elasticity, as expressed through the stiffness matrix $\mathbf{K}$, in two places. The first is in the system dynamics $\mathbf{F}$, which result in the model returning to a rest state in the absence of new measurements. The second is through the "prior smoothness" $\mathbf{K}_s$, which filters the new measurements to ensure smoothness and interpolation, without destroying shape estimates that may already be built up.

We now demonstrate these two behaviors of Kalman snakes. Figure 1.5 (a)–(e), which shows several frames from an image sequence of a rotating dodecahedral puzzle in which a closed snake is tracking the left-to-right motion of the dark boundary of one of the pentagonal faces.

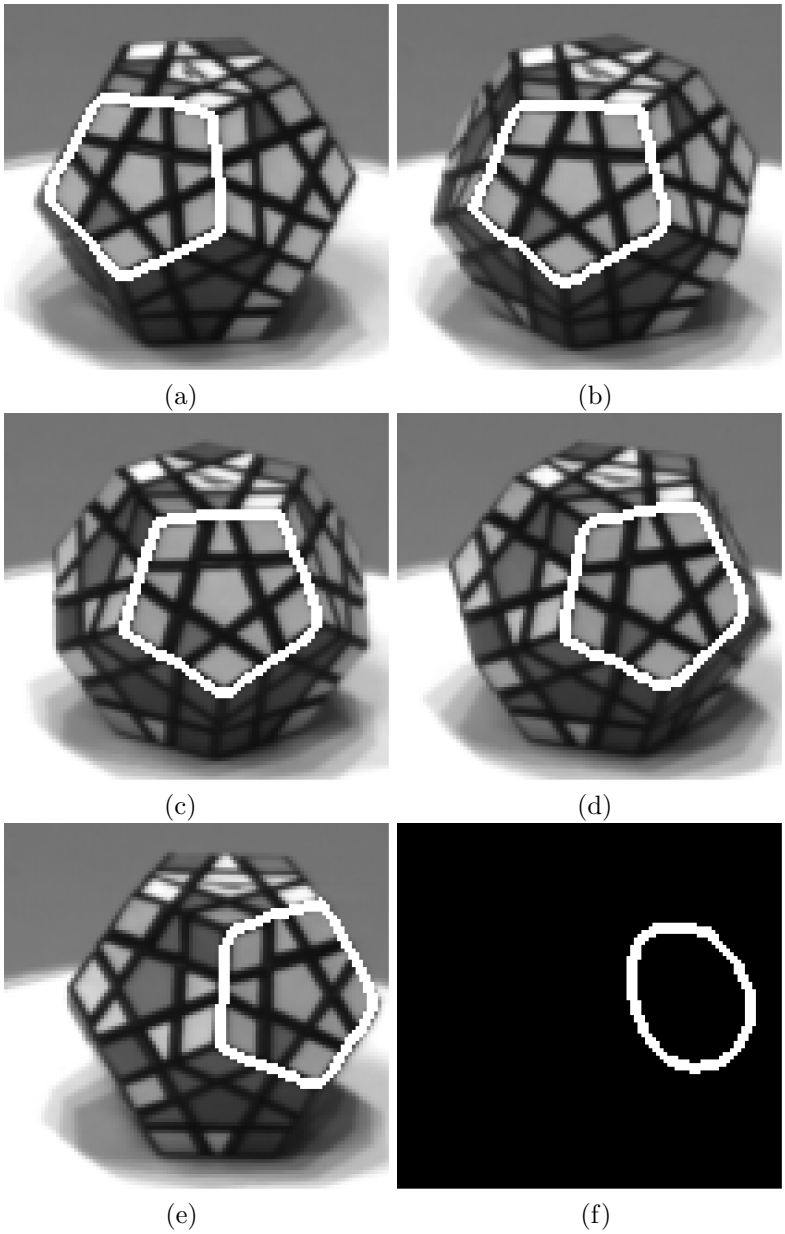
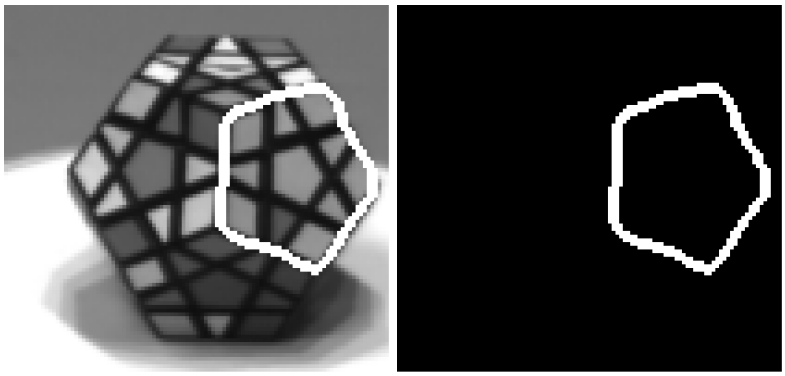


Figure 1.5
Snake tracking a rotating object. (a)–(e): frames 0–16 (steps of 4), (f): image data is occluded.



(a) (b)

Figure 1.6
Kalman snake. (a) snake at equilibrium in last frame of image sequence (b) snake retains shape after image data is occluded.

If the model smoothness and shape structure are totally in the dynamics **F**, then the snake will return to its natural, relaxed rest configuration when the image data is temporarily removed (such as when the object being tracked becomes occluded). For example, after the image data is removed from the snake in Figure 1.5(e) it relaxes to the equilibrium state in Figure 1.5(f).

If, on the other hand, the smoothness is totally in the prior, then the snake will retain its shape during occlusion, but will find it increasing difficult to adapt to non-rigid motion because of its adherence to old measurements ("sticky data"). The latter behavior is illustrated in Figure 1.6. Compare the equilibrium shape of the Kalman snake in Figure 1.6(b) to Figure 1.5(f).

The right blend of the aforementioned sources of *a priori* knowledge is application specific and depends on what is known about the real-world physics of the objects being modeled and about the sensor characteristics. For example, a rigid object should have a system model **F** which only supports rigid transformations, whereas a nonrigid object should have a deformable system model. The advantage of the Kalman filter which incorporates the Lagrangian dynamics of snakes is that it gives us the flexibility to design tracking behaviors that are not possible with conventional snakes. Moreover, the model parameters, such as how much to weight new measurements versus old shape estimates, can be derived

from statistical models of sensors, rather than being heuristically chosen, and they can vary over time.

1.4 Conclusion

The class of deformable models known as snakes has proven useful in several image analysis tasks. When applied to static images, the physical behavior of these models is useful mainly to support user interaction. The dynamical nature of snakes has a much larger impact, however, when they are applied to time-varying imagery. The dynamical system approach yields snakes and other deformable models that can track nonstationary data using forces, continuously adapting to to images of complicated rigidly or non-rigidly moving objects of interest. The probabilistic approach, especially the well-known techniques of Bayesian estimation and Kalman filtering, has proven successful in dealing with noisy measurements and integrating information from multiple sources (sensor fusion) and over time (sequential estimation). This chapter realizes the full potential of the dynamic and probabilistic approaches, by combining them to create a new sequential motion estimator, the Kalman snake. The Kalman snake uses snake dynamics as a system model to constrain and predict possible motions and, under the right conditions, it optimally acquires new information from uncertain measurements in a sequential fashion and optimally blends them with previous estimates. Chapter 5 will extend this approach to more sophisticated deformable models.

References

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

[2] R. Ayache, J. D. Boissonat, E. Brunet, L. D. Cohen, J. P. Chièze, B. Geiger, O. Monga, J.M. Rocchisani, and P. Sander. Building highly structured volume representations in 3D medical images. In *Proc. 3rd Symposium Computer Assisted Radiology*, pages 765–772, Berlin, June 1989.

[3] K.-J. Bathe and E. L. Wilson. *Numerical Methods in Finite Element Analysis.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1976.

[4] M.-O. Berger. Snake growing. In O. Faugeras, editor, *Proc. First European Conf. on Computer Vision*, pages 570–572, Antibes, France, April 1990.

[5] T. J. Broida and R. Chellappa. Kinematics and structure of a rigid object from a sequence of noisy images. In *Proc. Workshop on Motion: Representation and Analysis*, pages 95–100. IEEE, May 1986.

[6] I. Carlbom, D. Terzopoulos, and K. M. Harris. Reconstructing and visualizing models of neuronal dendrites. In N. M. Patrikalakis, editor, *Scientific Visualization of Physical Phenomena*, pages 623–638. Springer-Verlag, New York, 1991.

[7] R. Cipolla and A. Blake. The dynamic analysis of apparent contours. In *Proc. 3rd Int. Conf. on Computer Vision*, pages 616–623, Osaka, December 1990.

[8] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, March 1991.

[9] L. D. Cohen and I. Cohen. A finite element method applied to new active contour models and 3D reconstruction from cross sections. In *Proc. 3rd Int. Conf. on Computer Vision*, pages 587–591, Osaka, December 1990.

[10] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume I. Interscience, New York, 1953.

[11] R. M. Curwen, A. Blake, and R. Cipolla. Parallel implementation of Lagrangian dynamics for real-time snakes. In P. Mowforth, editor, *British Machine Vision Conference 1991*, pages 29–35. Springer-Verlag, London, 1991.

[12] C. David and S. W. Zucker. Potentials, valleys, and dynamic global coverings. *Int. Journal of Computer Vision*, 5(3):219–238, December 1990.

[13] R. Durbin, R. Szeliski, and A. Yuille. An analysis of the elastic net approach to the travelling salesman problem. *Neural Computation*, 1(3):348–358, Fall 1989.

[14] R. Durbin and D. Willshaw. An analogue approach to the traveling salesman problem using an elastic net method. *Nature*, 326:689–691, 16 April 1987.

[15] O. D. Faugeras, N. Ayache, and B. Faverjon. Building visual maps by combining noisy stereo measurements. In *Proc. Int. Conf. on Robotics and Automation*, pages 1433–1438, San Francisco, California, April 1986. IEEE Computer Society Press.

[16] F. P. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes, and superquadrics: Geometry from the bottom up. In *Proc. Workshop on Interpretation of 3D Scenes*, pages 170–176, Austin, TX, November 1989.

[17] P. Fua and Y. G. Leclerc. Model driven edge detection. *Machine Vision and Applications*, 3:45–56, 1990.

[18] Arthur Gelb, editor. *Applied Optimal Estimation.* MIT Press, Cambridge,

MA, 1974.

[19] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intell.*, PAMI-6(6):721–741, November 1984.

[20] J. Hallam. Resolving observer motion by object tracking. In *International Joint Conference on Artificial Intelligence*, 1983.

[21] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. Journal of Computer Vision*, 1(4):321–331, 1987.

[22] F. Leitner, I. Marque, S. Lavallée, and P. Cinquin. Dynamic segmentation: Finding the edge with differential equations and 'spline snakes'. Technical Report TIMB - TIM 3 - IMAG, Faculte de Medecine, La Tronche, France, 1990.

[23] F. Leymaire. Tracking and describing deformable objects using active contour models. Technical Report TR-CIM-90-9, McGill Research Center for Intelligent Machines, McGill University, Montreal, February 1990.

[24] F. Leymaire and M. D. Levine. Snakes and skeletons. Technical Report TR-CIM-89-3, McGill Research Center for Intelligent Machines, McGill University, Montreal, January 1989.

[25] P. Lipson, A. Yuille, D. O. Keefe, J. Cavanauch, J. Taffe, and D. Rosenthal. Deformable templates for feature extraction from medical images. In O. Faugeras, editor, *Proc. First European Conf. on Computer Vision*, pages 413–417, Antibes, France, April 1990.

[26] A. Martelli. Edge detection using heuristic search methods. *Computer Graphics and Image Processing*, 1:169–182, 1972.

[27] L. H. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *Int. Journal of Computer Vision*, 3:209–236, 1989.

[28] L. H. Matthies and S. A. Shafer. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, pages 239–248, June 1987.

[29] S. Menet, P. Saint-Marc, and G. Medioni. B-snakes: Implementations and applications to stereo. In *Proc. DARPA Image Understanding Workshop*, pages 720–726, Pittsburgh, PA, September 1990.

[30] D. Metaxas and D. Terzopoulos. Constrained deformable superquadrics and nonrigid motion tracking. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 337–343, Lahaina, HI, June 1991. IEEE Computer Society Press.

[31] D. Metaxas and D. Terzopoulos. Shape representation and nonrigid motion tracking using deformable superquadrics. In B. C. Vemuri, editor, *Proc. SPIE 1570, Geometric Methods in Computer Vision*, pages 12–20, San Diego, CA, July 1991. Society of Photo-Optical Instrumentation Engineers.

[32] P. L. Meyer. *Introductory Probability and Statistical Applications.* Addison-Wesley, Reading, MA, 2nd edition, 1970.

[33] U. Montaneri. On the optimal detection of curves in noisy pictures. *Commun. ACM*, 14(5):335–345, May 1971.

[34] T. Pavlidis and Y.-T. Liow. Integrated region growing and edge detection. *IEEE Trans. Pattern Analysis and Machine Intell.*, 12(3):225–233, March 1990.

[35] J. Schick and E. D. Dickmanns. Simultaneous estimation of 3D shape and motion of objects by computer vision. In *Proc. IEEE Workshop on Visual Motion*, pages 256–261, Princeton, NJ, October 1991. IEEE Computer Society Press.

[36] G. L. Scott. The alternative snake – and other animals. In J.-O. Eklundh, editor, *The 1987 Stockholm Workshop on Computational Vision,*, Stockholm, 1987. Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, TRITA-NA-P8714 CVAP 47.

[37] L. H. Staib and J. S. Duncan. Parametrically deformable contour models. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 98–103, San Diego, CA, June 1989. IEEE Computer Society Press.

[38] R. Szeliski. *Bayesian Modeling of Uncertainty in Low-Level Vision.* Kluwer Academic Publishers, Boston, MA, 1989.

[39] R. Szeliski and D. Terzopoulos. Physically-based and probabilistic modeling for computer vision. In B. C. Vemuri, editor, *Proc. SPIE 1570, Geometric Methods in Computer Vision*, pages 140–152, San Diego, CA, July 1991. Society of Photo-Optical Instrumentation Engineers.

[40] D. Terzopoulos. On matching deformable models to images: Direct and iterative solutions. In *Topical Meeting on Machine Vision, Technical Digest Series Vol. 12*, pages 160–167, Washington, D. C., March 1987. Optical Society of America.

[41] D. Terzopoulos and K. Waters. Analysis of facial images using physical and anatomical models. In *Third International Conference on Computer Vision (ICCV'90)*, pages 727–732, Osaka, Japan, December 1990.

[42] D. J. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *CVGIP: Image Understanding*, 55(1):14–26, January 1992.

[43] S.W. Zucker, C. David, A. Dobbins, and L. Iverson. The organization of curve detection: Coarse tangent fields and fine spline coverings. In *Proc. 2nd Int. Conf. on Computer Vision*, pages 568–577, Tampa, FL, December 1988.