Physically-Based and Probabilistic Models for Computer Vision

Richard Szeliski† and Demetri Terzopoulos‡,[1]

† Digital Equipment Corp., Cambridge Research Lab,
One Kendall Square, Bldg. 700, Cambridge, MA 02139
‡ Dept. of Computer Science, University of Toronto,
Toronto, Canada M5S 1A4

Abstract

Models of 2D and 3D objects are an essential aspect of computer vision. Physically-based models represent object shape and motion through dynamic differential equations and provide mechanisms for fitting and tracking visual data using simulated forces. Probabilistic models allow the incorporation of prior knowledge about shape and the optimal extraction of information from noisy sensory measurements. In this paper we propose a framework for combining the essential elements of both the physically-based and probabilistic approaches. The combined model is a Kalman filter which incorporates physically-based models as part of the prior and the system dynamics and is able to integrate noisy data over time. In particular, through a suitable choice of parameters we can build models which either return to a rest shape when external data are removed or remember shape cues seen previously. The proposed framework shows promise in a number of computer vision applications.

1. Introduction

Concepts from analytic, differential, and computational geometry have fueled a great deal of research on shape representation in computer vision. Although geometry may suffice to describe the shapes of static objects, it is often inadequate for the analysis and representation of complex real-world objects in motion. The deficiency, which becomes acute in the case of nonrigid motion, has motivated recent research into modeling methods based on computational physics. Physically-based models are fundamentally dynamic and are governed by the laws of rigid and nonrigid dynamics expressed through a set of Lagrangian equations of motion. These equations unify the description of object shape and object motion through space.

In addition to geometry, the formulation of physically-based models includes simulated forces, strain energies, and other physical quantities. External forces provide a general and highly intuitive means for coupling a physically-based model to various visual data, such as intensity and range images. Internal strain energies are a convenient tool for encoding constraints on the class of modeled shapes, e.g., surface smoothness and deformability. The evolution of the model's geometric variables or parameters under the action of internal constraints and external forces is computed by numerically simulating the equations of motion.

A variety of physically-based models have been developed for computer vision, including surface reconstruction models [Ter83, BZ87, Ter88, Sze89, TV91], snakes [KWT88], symmetry-seeking models [TWK88], deformable superquadrics [TM90, MT91], and modal models [HP91]. The generative power of physically-based models becomes important in applications to computer graphics (see, e.g., [TF88, ST89] and references therein). In situations where active control over models is desirable, the physically-based approach offers much more flexibility than manually adjusting geometric parameters. For example, the dynamics equations also provide a facile interface to the models through the use of force-based interaction tools. Efficient numerical simulation is an important consideration in supporting interactive dynamical models for real-time vision applications.

Physically-based models give us a potent approach for recovering the geometric and dynamic structure of the visual world by fitting models to the observed data through the use of forces. An alternative and complementary

[1] Fellow, Canadian Institute for Advanced Research

0-8194-0698-8/91/$4.00

approach is to cast the model fitting process in a probabilistic framework and to view model recovery as an *estimation* problem. The rationale for using probabilistic models lies in the inherently noisy nature of most sensors used in vision. Formulating vision tasks as statistical estimation problems allows us to model the noise or uncertainty in our sensors explicitly, and to compute statistically optimal estimates of the underlying features or objects we are trying to reconstruct. It also allows us to compute the uncertainty in these estimates, which can be used by higher-level processes or for multi-sensor integration [Sze89].

Sensor models can be used by themselves to compute maximum likelihood (ML) estimates. Probabilistic modeling becomes much more powerful, however, if we add prior distributions to the geometric parameters we are estimating. The combination of prior and sensor models results in a Bayesian model, since the posterior distribution of the variables we are trying to estimate conditioned on the data can be computed using Bayes' Rule. The greatest challenge in formulating Bayesian solutions for computer vision problems is to find prior models that both capture the inherent complexity of the visual world and are computationally tractable.

A realization of potentially enormous consequences in computer vision is that physically-based models can serve as prior models. A key step in forging the link is to apply a technique of statistical mechanics—conversion of energies into probabilities using the Boltzmann, or Gibbs, distribution. For example, after suitable discretization, a continuous strain energy that governs the deformation of a physically-based model away from its natural shape may be converted into a probability distribution over expected shapes, with lower energy shapes being the more likely.

The full impact of estimation with physically-based models is realized by optimal estimation, where the goal is to estimate the state of a system using an assumed model of the system and sensor dynamics in addition to assumed statistics of system inaccuracies and measurement errors. In this context, a physically-based model—not merely the internal strain energy, but the complete equations of motion—plays the role of a nonstationary prior model in the visual problem under analysis. For on-line vision applications involving time-varying sensory data, Kalman filtering theory provides the computational framework for optimally estimating dynamic model parameters in an efficient, recursive fashion.

In this paper, we examine the physically-based and probabilistic modeling approaches to computer vision and propose a framework which combines elements from both. Our framework is a Kalman filter which uses physically-based models both as part of the prior and as part of the system dynamics. Through a suitable choice of parameters, we can build models which either return to a rest shape when external data is removed or retain all of the shape cues seen previously. Such models are promising in a number of computer vision applications.

2. Physically-based modeling

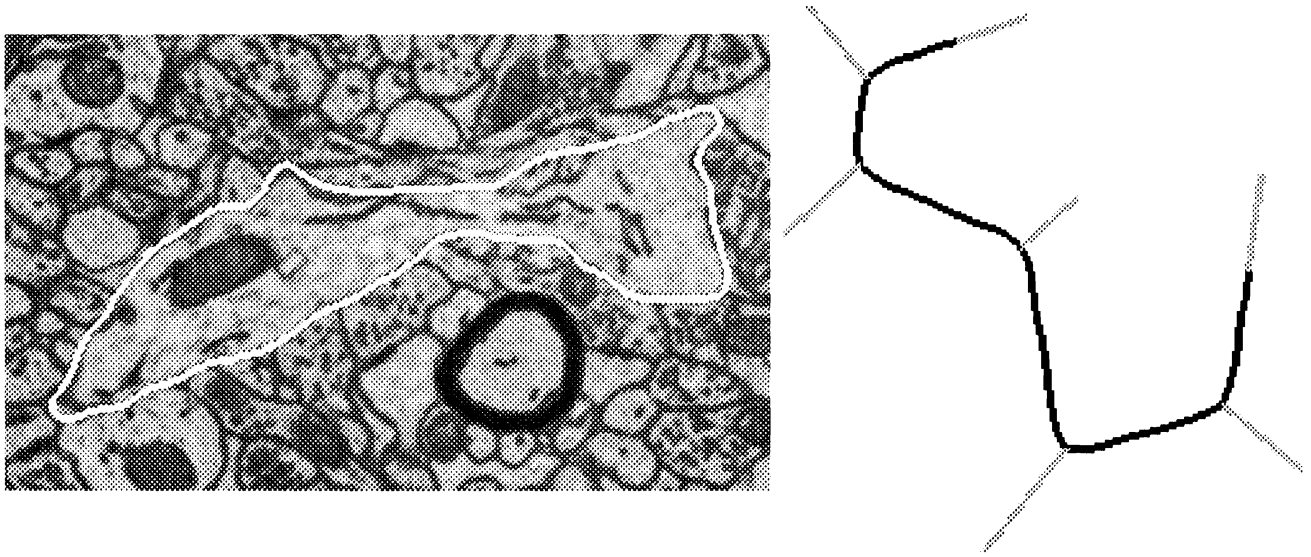
A convenient approach to creating physically-based models is to begin by devising energy functionals that have some physical interpretation. Physically-based deformable models, for example, are defined by constructing suitable deformation energies $\mathcal{E}(\mathbf{v})$, where $\mathbf{v}$ specifies the configuration of the model, generally a mapping from a parametric domain $\mathbf{x} \in \Omega$ into a spatial domain. The minimization of the energy characterizes the desired equilibrium configuration of the model. It is natural to view energy minimization as a static problem. The first part of this section examines the static point of view, while in the second part we emphasize the possibility of minimizing energies using dynamical systems derived by applying the principles of Lagrangian mechanics. This yields dynamic, physically-based models 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 by a human operator as it minimizes energy. For concreteness, we will illustrate these ideas by considering a simple physically-based deformable model called a *snake*.

2.1. Example: Snake models

Snakes [KWT88] are a class of energy minimizing deformable contours that move under the influence of external potentials $\mathcal{P}$. The local minima of $\mathcal{P}$ attract points on the snakes. External potentials whose local minima correspond to, for example, intensity extrema, edges, and other image features are readily designed by applying simple image processing. Fig. 1(a) shows a snake fitted to the membrane of a cell in an EM photomicrograph (see [CTH91] for details).

To formulate the snake model we parameterize the contour by $\mathbf{x} = s \in [0, 1] = \Omega$. The spatial domain of the model is the image plane (x, y), where the components of the mapping $\mathbf{v}(s) = (x(s), y(s))$ are image coordinates. We prescribe the energy

$$\mathcal{E}(\mathbf{v}) = \mathcal{E}_s(\mathbf{v}) + \mathcal{P}(\mathbf{v}). \tag{1}$$



(a) (b)

Figure 1: Snake fitting: (a) snake fitted to cell membrane in EM photomicrograph (b) snake constrained with spring forces

For a simple (linear) snake,

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

is a deformation energy, 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 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. See Appendix A for multidimensional generalizations of (2), suitable for modeling surfaces and volumes.

We typically define

$$\mathcal{P}(\mathbf{v}) = \int_\Omega P(\mathbf{v}(s))\, ds \tag{3}$$

where P is a potential function derived by processing of a grey level image $I(x, y)$. For example, the snake will have an affinity for darkness or brightness if $P[\mathbf{v}(s)] = \pm c[G_\sigma * I(\mathbf{v}(s))]$, depending on the sign, and for intensity edges if $P[\mathbf{v}(s)] = -c|\nabla[G_\sigma * I(\mathbf{v}(s))]|$, where c controls the magnitude of the potential and $G_\sigma * I$ denotes the image convolved with a (Gaussian) smoothing filter whose characteristic width is σ. Another useful variant is $P[\mathbf{v}(s)] = c(s)[\mathbf{v}(s) - \mathbf{d}(s)]^2$ which attracts the snake towards a target contour $\mathbf{d}(s)$ with a coupling strength $c(s)$. This is a continuous generalization of the discrete data energy

$$\mathcal{P}(\mathbf{v}) = \sum_i c_i[\mathbf{v}(s_i) - \mathbf{d}_i]^2 \tag{4}$$

which can be interpreted as coupling the snake to a collection of "nails" $\mathbf{d}_i$ in the image plane through a set of "springs" with stiffnesses c_i (Fig. 1(b)).

2.2. Lagrangian dynamics

If the potential P changes after a snake has achieved equilibrium, potential energy is converted to kinetic energy, the equilibrium is perturbed, and the model will move nonrigidly to achieve a new equilibrium. We can represent the motion explicitly by introducing a time-varying mapping $\mathbf{v}(s, t)$ and a kinetic energy $\int \mu|\mathbf{v}_t|^2\, ds$, where $\mu(s)$ is the mass density and the subscript t denotes a time derivative. If kinetic energy is dissipated by damping, then the transients decay until a new equilibrium is reached.

Given the deformation potential energy functional $\mathcal{E}(\mathbf{v})$, we define the Lagrangian functional

$$\mathcal{L}(\mathbf{v}) = \frac{1}{2}\int_\Omega \mu|\mathbf{v}_t|^2\,ds - \frac{1}{2}\mathcal{E}(\mathbf{v}), \tag{5}$$

as well as the (Rayleigh) dissipation functional $\mathcal{D}(\mathbf{v}_t) = \frac{1}{2}\int_\Omega \gamma|\mathbf{v}_t|^2\,ds$, where $\gamma(s)$ is the damping density.

If the initial and final configurations are $\mathbf{v}(s,t_0)$ and $\mathbf{v}(s,t_1)$, the deformable model's motion $\mathbf{v}(s,t)$ is such that

$$\frac{\delta}{\delta \mathbf{v}}\left(\int_{t_0}^{t_1} \mathcal{L}(\mathbf{v}) + \mathcal{D}(\mathbf{v}_t)\,dt\right) = \mathbf{0}. \tag{6}$$

The term in brackets is known as the action integral, and the above condition that it be stationary—i.e., that the variational derivative with respect to $\mathbf{v}$ vanishes—leads to Lagrange's equations of motion.

Assuming constant mass density $\mu(s) = \mu$ and constant dissipation $\gamma(s) = \gamma$, the Lagrange equations for the snake model with deformation energy (2) and external potential (3) are

$$\mu\mathbf{v}_{tt} + \gamma\mathbf{v}_t + \frac{\partial}{\partial s}\left(w_1\mathbf{v}_s\right) + \frac{\partial^2}{\partial s^2}\left(w_2\mathbf{v}_{ss}\right) = -\nabla P(\mathbf{v}(s,t)) \tag{7}$$

with appropriate initial and boundary conditions.

2.3. Discretization

Physically-based models are often defined continuously in the parametric domain Ω, as is the snake model above. It is necessary to discretize the energy $\mathcal{E}(\mathbf{v})$ in order to numerically compute the minimal energy solution. 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 (7), associated with the model.

The discrete form of quadratic energies such as (1) may be written as

$$E(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} + \mathrm{P}(\mathbf{u}), \tag{8}$$

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 (8) to $\mathbf{0}$, which is equivalent to solving the set of algebraic equations

$$\mathbf{K}\mathbf{u} = -\nabla \mathrm{P} = \mathbf{g} \tag{9}$$

where $\mathbf{g}$ may be interpreted as a *generalized force vector*.

Quadratic external potentials such as (4) may be written as quadratic forms

$$\mathrm{P}(\mathbf{u}) = \frac{1}{2}(\mathbf{H}\mathbf{u} - \mathbf{d})^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{u} - \mathbf{d}), \tag{10}$$

where $\mathbf{H}$ is the *interpolation* or *measurement* matrix which maps from the nodal variables to the locations of the discrete data measurements $\mathbf{d}$, and $\mathbf{R}^{-1}$ encodes the confidence in the external data measurements. According to (9),

$$\mathbf{g} = \mathbf{H}^T\mathbf{R}^{-1}(\mathbf{d} - \mathbf{H}\mathbf{u}), \tag{11}$$

which can be interpreted as external spring forces coupling the physical model to the data. For the snake model, $\mathbf{H}\mathbf{u}$ indicates the points on the model where springs are attached, and the entries of $\mathbf{R}^{-1}$ are the individual spring stiffnesses c_i. Note that if the dimensionality of $\mathbf{d}$ is smaller than that of $\mathbf{u}$, the model will interpolate 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 model will provide a least squares fit to the data. Both cases are handled by the same measurement equation.

For this quadratic external potential, the set of linear equations (9) can be written as

$$(\mathbf{K} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})\mathbf{u} = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{d}, \tag{12}$$

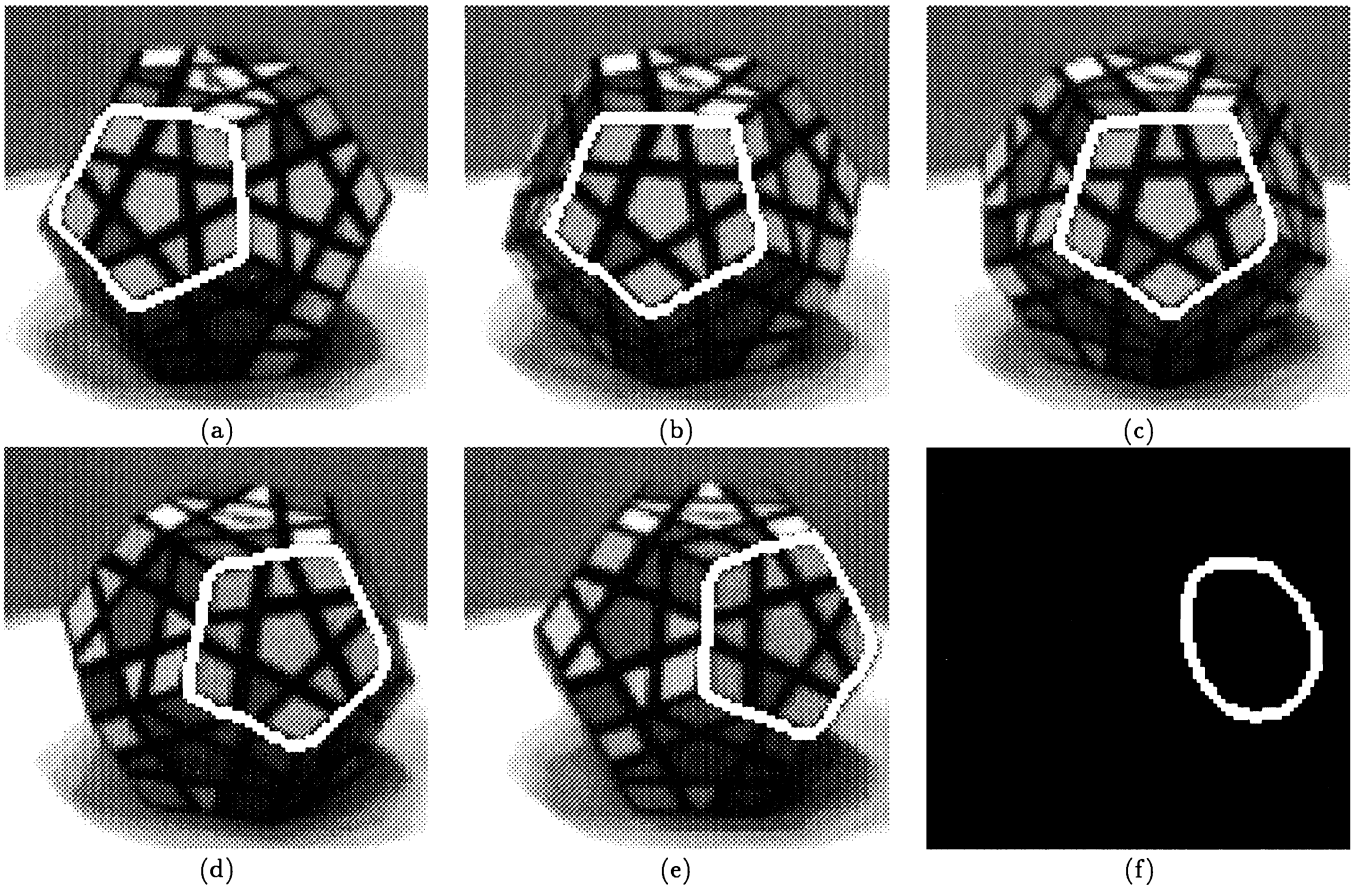


Figure 2: Snake tracking a rotating object: (a)–(e): frames 0–16 (steps of 4), (f): image data is removed

or as

$$\mathbf{A}\mathbf{u} = \mathbf{b} \tag{13}$$

with $\mathbf{A} = \mathbf{K} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ and $\mathbf{b} = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{d}$.

The discretized version of the Lagrangian dynamics equations (7) may in turn be written as a set of second order differential equations for $\mathbf{u}(t)$:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{g}, \tag{14}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is a damping matrix, and $\mathbf{K}$ is a stiffness matrix. These matrices will have a sparse and banded structure as a consequence of a finite element or finite difference discretization of the continuous model.

Denoting the nodal velocities $\dot{\mathbf{u}}$ as $\mathbf{v}$, (14) may be written as a coupled set of first order equations

$$\begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} -\mathbf{M}^{-1}\mathbf{C} & -\mathbf{M}^{-1}\mathbf{K} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{u} \end{bmatrix} + \begin{bmatrix} \mathbf{M}^{-1}\mathbf{g} \\ \mathbf{0} \end{bmatrix}. \tag{15}$$

Note that for static $\mathbf{g}$, the dynamic equilibrium condition $\ddot{\mathbf{u}} = \dot{\mathbf{u}} = \mathbf{0}$ leads to the static solution (9), as expected. If we assume that the system is massless (i.e., that $\mathbf{M} = \mathbf{0}$) (14) reduces to the simpler set of first-order equations

$$\dot{\mathbf{u}} = -\mathbf{C}^{-1}\mathbf{K}\mathbf{u} + \mathbf{C}^{-1}\mathbf{g}. \tag{16}$$

2.4. Tracking dynamic data

The interpretation of $\mathbf{g}$ as generalized external forces that couple the physically-based model to the data—recall, in particular, the interpretation of (11) as spring forces—suggests a straightforward mechanism for tracking dynamic data [KWT88, TWK88]: If the data vector $\mathbf{d}(t)$ varies as a function of time, the forces exerted on the model will also be time-varying. The action of the forces will pull the model through space as governed by the equations of

motion (14) or (15). The model will track the dynamic data as it attempts to maintain the generalized forces $\mathbf{g}(t)$ in dynamic equilibrium against the inertial, damping, and deformation forces. A snake model illustrates the force driven tracking procedure in Fig. 2(a)–(e), which shows several frames from an image sequence of a rotating dodecahedral object in which a closed snake is tracking the left-to-right motion of one of the pentagonal faces.

In Section 4, 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 the data and model them in an optimal way. This generalization will utilize the probabilistic analysis of the static physically-based modeling problem, which is presented next.

3. Probabilistic models

Physically-based models allow us to recover the geometric and dynamic structure of the visual world by fitting models to the observed data through the use of forces. An alternative and complementary approach is to cast the same problem in a probabilistic framework and to view model recovery as an estimation problem.

A particularly powerful probabilistic model is a *Bayesian model*, where the posterior distribution $p(\mathbf{u}|\mathbf{d})$ of the unknown we are trying to recover $\mathbf{u}$ conditioned on the data $\mathbf{d}$ can be computed using Bayes' Rule

$$p(\mathbf{u}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{u})\,p(\mathbf{u})}{p(\mathbf{d})} \tag{17}$$

with the normalizing denominator

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

In the above equation, 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 $\mathbf{d}$. These two probabilistic models are combined using Bayes' rule to obtain a *posterior model* $p(\mathbf{u}|\mathbf{d})$ which is a probabilistic description of the current estimate of $\mathbf{u}$ given the data $\mathbf{d}$ [Mey70].

The physically-based modeling approach provides a good source of prior models that both capture the physical complexity of the visual world and are computationally tractable. The resulting prior distributions can be used to bias Bayesian solutions towards low energy configurations. The link between physically-based models and suitable priors is conveniently established using a Gibbs (or Boltzmann) distribution of the form

$$p(\mathbf{u}) = \frac{1}{Z_{\mathrm{p}}} \exp(-E_{\mathrm{p}}(\mathbf{u})), \tag{18}$$

where $E_{\mathrm{p}}(\mathbf{u})$ is the discretized version of the internal smoothness energy $\mathcal{E}_s$ of the model, and Z_{p} (called the *partition function*) is a normalizing constant. What was originally an elastic energy restoring a model towards a rest state now becomes a probability distribution over expected shapes, with lower energy shapes being more likely.

If $E_{\mathrm{p}}(\mathbf{u})$ is a quadratic energy of the form $E_{\mathrm{p}}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u}$, the prior distribution is a correlated zero-mean Gaussian with a covariance $\mathbf{P} = \mathbf{K}^{-1}$. For physically-based models such as elastic curves or surfaces, $\mathbf{K}$ will typically be sparse and banded, but $\mathbf{P}$ will not. In general, when the energy function $E_{\mathrm{p}}(\mathbf{u})$ can be written as a sum of local clique energies, e.g., when $E_{\mathrm{p}}(\mathbf{u})$ arises from a finite-element discretization, the distribution (18) is a Markov Random Field [GG84].

To complete the formulation of the estimation problem, we combine this prior model with a simple sensor model based on linear measurements with Gaussian noise

$$p(\mathbf{d}|\mathbf{u}) = \frac{1}{Z_{\mathrm{d}}} \exp(-E_{\mathrm{d}}(\mathbf{u}, \mathbf{d})), \tag{19}$$

with

$$E_{\mathrm{d}}(\mathbf{u}, \mathbf{d}) = \frac{1}{2}\sum_i \frac{1}{\sigma_i^2}|\mathbf{H}_i\mathbf{u} - \mathbf{d}_i|^2 = \frac{1}{2}(\mathbf{H}\mathbf{u} - \mathbf{d})^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{u} - \mathbf{d}), \tag{20}$$

Combining the prior (18) and the sensor (19) models using Bayes' rule, we obtain the posterior distribution

$$p(\mathbf{u}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{u})\,p(\mathbf{u})}{p(\mathbf{d})} = \frac{1}{Z}\exp(-E(\mathbf{u})), \tag{21}$$

where

$$E(\mathbf{u}) = E_{\mathrm{p}}(\mathbf{u}) + E_{\mathrm{d}}(\mathbf{u}, \mathbf{d}). \tag{22}$$

Note that this is the same energy equation as (8), which describes the energy of a constrained physically based model. Thus, computing the *Maximum A Posteriori* (MAP) estimate [GG84], 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 physically-based model.

Although both physically-based and probabilistic models may be used to produce the same estimate, there are several advantages to using a probabilistic formulation. First, the statistical assumptions corresponding to the internal energy model can be explored by randomly generating samples from the prior model [Sze87] (this also gives us a powerful method for generating stochastic models such as fractals [ST89]). Second, the parameters for the prior model itself can be estimated by gathering statistics over the ensemble of objects (as is done in image coding) [Pen84]. Third, the external force fields (data constraints) can be derived in a principled fashion taking into account the known noise characteristics of the sensors or algorithms (Appendix B). Fourth, alternative estimates can be computed by changing the cost functions whose expected value is being minimized [Sze89]. Fifth, the uncertainty in the posterior model can be quantified and used by higher levels stages of processing (Appendix C).

4. Sequential estimation using the Kalman filter

The most compelling reason for using probabilistic models is to develop sequential estimation algorithms, where measurements are integrated over time to improve the accuracy of estimates. Such sequential estimation algorithms become even more potent when they are combined with the dynamic physically-based models we described in Section 2. The resulting estimation algorithm is known as the continuous Kalman filter [Gel74].

The Kalman filter is designed by adding a *system model* to the prior and sensor models already present in the Bayesian formulation. This system model describes the expected evolution of the state $\mathbf{u}(t)$ over time[1]. In the case of linear system dynamics, e.g., the Lagrangian dynamics of (15) or (16), the system model can be written as a differential equation

$$\dot{\mathbf{u}} = \mathbf{F}\mathbf{u} + \mathbf{q}, \qquad \mathbf{q} \sim N(\mathbf{0}, \mathbf{Q}), \tag{23}$$

where $\mathbf{F}$ is called the *transition matrix* and $\mathbf{q}$ is a white Gaussian noise process with covariance $\mathbf{Q}_k$. The system noise is used to model unknown disturbances or imperfections in the system dynamics model. The sensor model component $p(\mathbf{d}|\mathbf{u})$ of our Bayesian formulation is rewritten as

$$\mathbf{d} = \mathbf{H}\mathbf{u} + \mathbf{r}, \qquad \mathbf{r} \sim N(\mathbf{0}, \mathbf{R}), \tag{24}$$

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 a state estimate $\hat{\mathbf{u}}$ and an error covariance matrix $\mathbf{P}$. The state estimate equation

$$\dot{\hat{\mathbf{u}}} = \mathbf{F}\hat{\mathbf{u}} + \mathbf{G}[\mathbf{d} - \mathbf{H}\hat{\mathbf{u}}] \tag{25}$$

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* $\mathbf{G}$. From Kalman filter theory, we can show that the optimal choice for the gain is

$$\mathbf{G} = \mathbf{S}^{-1}\mathbf{H}^T\mathbf{R}^{-1}, \tag{26}$$

where $\mathbf{S}$ 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.[2]

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

[2] To relate the second term in (25) back to the physically-based modeling approach, the weighted residual error $(\mathbf{d} - \mathbf{H}\hat{\mathbf{u}})$ can be interpreted as the deformations of springs coupling selected state variables $\mathbf{H}\mathbf{u}$ 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 information matrix $\mathbf{S}$ itself is updated over time using the *matrix Riccati equation* [Gel74, p. 122] expressed in terms of the inverse covariance (Appendix D)

$$\dot{\mathbf{S}} = -\mathbf{SF} - \mathbf{F}^T\mathbf{S} - \mathbf{SQS} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}. \tag{27}$$

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 equilibrates into a quasi-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.

Our reason for choosing to use the inverse covariance formulation rather than the more commonly used covariance formulation has to do with the nature of the prior distributions that arise in physically-based modeling. As we showed in Section 3, a sensible choice for the prior distribution is a multivariate Gaussian with a covariance $\mathbf{P}(0) = \mathbf{K}_s^{-1}$, where $\mathbf{K}_s$ is the stiffness matrix computed using finite element analysis. $\mathbf{K}_s$ will typically be sparse and banded, whereas $\mathbf{P}(0)$ (and hence $\mathbf{P}(t)$) will not. For finite element models with large numbers of nodal variables, storing and updating this dense covariance matrix is not practical.

We can derive a more convenient approximation to the true Kalman filter 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 $\mathbf{S}'$

$$\mathbf{S}(t) = \mathbf{K}_s + \mathbf{S}'(t). \tag{28}$$

We then apply the Riccati equation (27) directly to $\mathbf{S}'$, and ignore any off-diagonal terms that arise. The state update equation (25) becomes

$$\dot{\hat{\mathbf{u}}} = \mathbf{F}\hat{\mathbf{u}} + (\mathbf{K}_s + \mathbf{S}')^{-1}\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{d} - \mathbf{H}\hat{\mathbf{u}}) \tag{29}$$

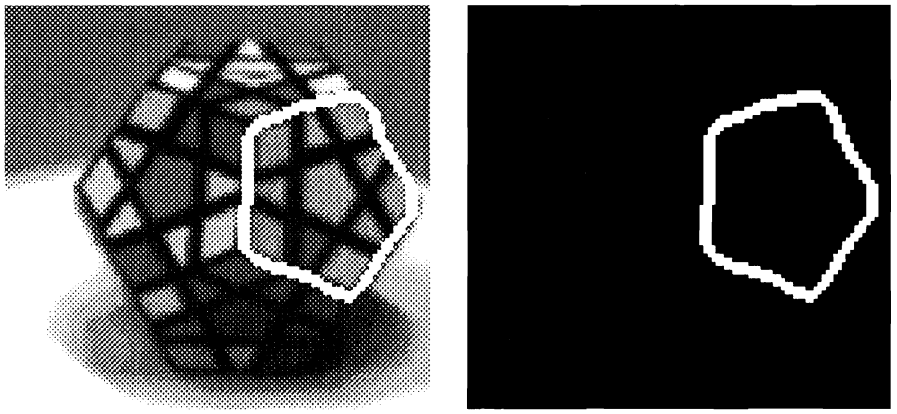
The resulting physically-based Kalman filter estimator has the following structure. The state update equation (29) changes the current state estimate according to both the dynamics of the system described by $\mathbf{F}$ (which may include internal elastic forces) and according to the filtered difference between the sampled state $\mathbf{Hu}$ and the data values $\mathbf{d}$ (these can be replaced by other external forces). The Kalman gain filter 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 (29).

We therefore have two mechanisms for introducing physically-based behaviors into our estimation system. The first is through 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.

This behavior may be demonstrated using a snake tracking data such as image-based gradients over time. If the model smoothness and shape structure are totally in the dynamics $\mathbf{F}$, 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 Fig. 2(e) it relaxes to the equilibrium state in Fig. 2(f). If 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 Fig. 3. Compare the equilibrium shape of the Kalman snake in Fig. 3(b) to Fig. 2(f).

The right blend of the aforementioned sources of a priori knowledge is application specific and dependent on the accuracy of what is known about the real-world physics of the objects being modeled and about the sensor characteristics. The advantage of the Kalman filter which incorporates the Lagrangian dynamics of physically-based models is that it gives us the flexibility to design behaviors that are not possible with pure physically-based models. 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 (as opposed to the fixed weights used in [TWK88, HP91]).

In many applications of physically-based modeling, the measurements $\mathbf{d}$ may be related to the state variables $\mathbf{u}$ through a non-linear function $\mathbf{d} = \mathbf{h}(\mathbf{u})$. For example, when the shape of the object is described by a locally and globally deformable geometric primitive [TM90, MT91], the relationship between the surface points that are



(a) (b)

Figure 3: Kalman snake: (a) snake at equilibrium in last frame of image sequence (b) snake retains shape after image data is removed

attracted to data and the global shape parameters is non-linear. The Kalman filter which we have developed here can be applied in this case as well, although the estimates thus provided may be sub-optimal [Gel74]. To develop this *extended Kalman filter*, we replace the measurement equation (24) with

$$\mathbf{d} = \mathbf{h}(\mathbf{u}) + \mathbf{r}, \qquad \mathbf{r} \sim N(\mathbf{0}, \mathbf{R}). \tag{30}$$

The same Kalman filter updating equations can be used as before, except that we now continuously evaluate the measurement matrix

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}(\mathbf{u})}{\partial \mathbf{u}}\right|_{\mathbf{u}=\hat{\mathbf{u}}} \tag{31}$$

as the partial derivative of the measurement function. Similarly, if the dynamics themselves are nonlinear $\dot{\mathbf{u}} = \mathbf{f}(\mathbf{u})$, e.g., if the physically-based model contains non-zero rest length springs, we use the partial of $\mathbf{f}$ with respect to $\mathbf{u}$ instead of $\mathbf{F}$ [Gel74].

5. Conclusions

Physically-based models have proven useful in various low-level vision tasks involving the recovery of shape and motion from incomplete, noisy data. When applied to static data, these models reduce to energy minimization methods, which include regularization, a familiar approach to formulating and solving visual inverse problems. Physically-based techniques also lead to more general dynamic models and associated procedures for fitting them to visual data using virtual forces. This latter view is particularly beneficial when dealing with the dynamic world, suggesting flexible models that can track nonstationary data and continuously adapt to complex objects that undergo rigid or non-rigid motion.

Probabilistic models have proven successful in dealing with noisy measurements and integrating information from multiple sources (sensor fusion) and over time (sequential estimation). Powerful prior models to support Bayesian estimation procedures can be derived from physically-based models by using the deformation energies of the models to define prior probability distributions, with lower energy states being more likely.

The full impact of these two techniques has been realized in a new sequential estimation algorithm, a Kalman filter where the physically-based model encodes constraints about the dynamics of objects of interest and provides energy-based constraints on their shapes. The resulting estimator resembles a conventional physically-based dynamic model, except that it also optimally blends new measurements with old estimates. Through a suitable choice of parameters, we can build models which either return to a rest shape when external data is removed or retain shape cues seen previously. We have demonstrated the behavior of such a system using a simple deformable contour (snake) as an example.

We are currently applying this approach to more sophisticated models, such as deformable part/surface models with rigid-body dynamics [TM90, MT91] and evolving surface descriptions estimated from monocular image sequences [Sze91a, Sze91b]. The incorporation of this repertoire of models within a sequential estimation framework appears very promising for a variety of computer vision applications.

References

[BW76] K.-J. Bathe and E. L. Wilson. *Numerical Methods in Finite Element Analysis.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

[BZ87] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, Cambridge, Massachusetts, 1987.

[CTH91] 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.

[DSY89] R. Durbin, R. Szeliski, and A. Yuille. An analysis of the elastic net approach to the travelling salesman problem. In *Snowbird Neural Networks Meeting*, Snowbird, Utah, April 1989.

[DW87a] 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.

[DW87b] H. F. Durrant-Whyte. Consistent integration and propagation of disparate sensor observations. *International Journal of Robotics Research*, 6(3):3–24, Fall 1987.

[Gel74] Arthur Gelb, editor. *Applied Optimal Estimation.* MIT Press, Cambridge, Massachusetts, 1974.

[GG84] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, November 1984.

[HP91] B. Horowitz and A. Pentland. Recovery of non-rigid motion and structure. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 325–330, Maui, Hawaii, June 1991. IEEE Computer Society Press.

[Hub81] P. J. Huber. *Robust Statistics.* John Wiley & Sons, New York, New York, 1981.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, January 1988.

[Mey70] P. L. Meyer. *Introductory Probability and Statistical Applications.* Addison-Wesley, Reading, MA, 2 edition, 1970.

[MT91] D. Metaxas and D. Terzopoulos. Constrained deformable superquadrics and nonrigid motion tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 337–343, Maui, Hawaii, June 1991. IEEE Computer Society Press.

[Pen84] A. P. Pentland. Fractal-based description of natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):661–674, November 1984.

[ST89] R. Szeliski and D. Terzopoulos. From splines to fractals. *Computer Graphics (SIGGRAPH'89)*, 23(4):51–60, July 1989.

[Sze87] R. Szeliski. Regularization uses fractal priors. In *Sixth National Conference on Artificial Intelligence (AAAI-87)*, pages 749–754, Seattle, Washington, July 1987. Morgan Kaufmann Publishers.

[Sze89] R. Szeliski. *Bayesian Modeling of Uncertainty in Low-Level Vision.* Kluwer Academic Publishers, Boston, Massachusetts, 1989.

[Sze91a] R. Szeliski. Probabilistic surface modeling. In *SPIE Vol. 1570 Geometric Methods in Computer Vision*, San Diego, July 1991. Society of Photo-Optical Instrumentation Engineers.

[Sze91b] R. Szeliski. Shape from rotation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 625–630, Maui, Hawaii, June 1991. IEEE Computer Society Press.

[Ter83] D. Terzopoulos. Multilevel computational processes for visual surface reconstruction. *Computer Vision, Graphics, and Image Processing*, 24:52–96, 1983.

[Ter86] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4):413–424, July 1986.

[Ter88] D. Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10(4):417–438, July 1988.

[TF88] D. Terzopoulos and K. Fleischer. Deformable models. *The Visual Computer*, 4(6):306–331, December 1988.

[TM90] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. In *Third International Conference on Computer Vision (ICCV'90)*, pages 606–615, Osaka, Japan, December 1990.

[TV91] D. Terzopoulos and M. Vasilescu. Sampling and reconstruction with adaptive meshes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 70–75, Maui, Hawaii, June 1991. IEEE Computer Society Press.

[TWK88] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Rcovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123, August 1988.

A. Multidimensional deformable models

Multidimensional generalizations of the snake model are readily obtained [Ter86]. Let $\mathbf{x} = (x^1, \ldots, x^p) \in \Re^p$ be a point in parameter space. Let Ω be a subset of $\Re^p$ with boundary $\partial\Omega$. A model is given by the image of the mapping $\mathbf{v}(\mathbf{x}) = (v_1(\mathbf{x}), \ldots, v_d(\mathbf{x}))$.

A generalized deformable model of order $q > 0$ minimizes the deformation energy functional

$$\mathcal{E}_q(\mathbf{v}) = \sum_{m=1}^{q} \sum_{|j|=m} \frac{m!}{j_1! \ldots j_p!} \int_\Omega w_j(\mathbf{x}) \left| \frac{\partial^m \mathbf{v}(\mathbf{x})}{\partial x_1^{j_1} \ldots \partial x_p^{j_p}} \right|^2 d\mathbf{x} + \int_\Omega P[\mathbf{v}(\mathbf{x})]\, d\mathbf{x}. \tag{32}$$

Here, $j = (j_1, \ldots, j_p)$ is a multi-index with $|j| = j_1 + \ldots + j_p$, and P is a generalized potential function associated with the externally applied force field. The deformation of the model is controlled by the vector $\mathbf{w}(\mathbf{x})$ of distributed parameter functions $w_j(\mathbf{x})$. For instance, a discontinuity of order $k < q$ is permitted to occur at $\mathbf{x}_0$ in the limit as $w_j(\mathbf{x}_0) \to 0$ for $|j| > k$. described below.

The Lagrange equations for the general model are given by

$$\frac{\partial}{\partial t}\left(\mu \frac{\partial u_i}{\partial t}\right) + \gamma \frac{\partial u_i}{\partial t} + \sum_{m=0}^{q} (-1)^m \Delta_{\mathbf{w}_m}^m u_i = -\frac{\partial P}{\partial u_i}, \qquad i = 1, \ldots, d, \tag{33}$$

where $\Delta_{\mathbf{w}_m}^m$ is the weighted iterated Laplacian operator defined by

$$\Delta_{\mathbf{w}_m}^m = \sum_{j_1 + \cdots + j_p = m} \frac{m!}{j_1! \ldots j_d!} \frac{\partial^m}{\partial x_1^{j_1} \ldots \partial x_d^{j_d}} \left(w_j(\mathbf{x}) \frac{\partial^m}{\partial x_1^{j_1} \ldots \partial x_d^{j_d}} \right) \tag{34}$$

and $j = (j_1, \ldots, j_p)$ is a multi-index. Associated with these equations are appropriate initial conditions on $\mathbf{u}$ and boundary conditions for this function on $\partial\Omega$.

B. More sophisticated sensor models

In general, knowledge of how individual measurements (be they intensities, direct position measurements, or the results of a low-level algorithm) were derived from the underlying model can be used to construct a sensor model. This model can, in turn, be converted into external constraints on the physically-based model using (19). The simplest example, which we explored in Section 3, is the case of sparse position measurements contaminated with white (independent) Gaussian noise. In this case, the inverse variance σ_i^{-2} of the noise associated with each measurement determines the stiffness k_i of the spring coupling the deformable model to each data point (4).

A Gaussian noise model is appropriate when the error in the measurement is the result of the aggregation of many small random disturbances. Many sensors, however, not only have a normal operating range characterized by a small σ^2, but also occasionally produce gross errors. A more appropriate model for such a sensor is the *contaminated Gaussian* [DW87b], which has the form

$$p(\mathbf{d}_i|\mathbf{v}(\mathbf{x}_i)) = \frac{1-\epsilon}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{|\mathbf{v}(\mathbf{x}_i) - \mathbf{d}_i|^2}{2\sigma_1^2}\right) + \frac{\epsilon}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{|\mathbf{v}(\mathbf{x}_i) - \mathbf{d}_i|^2}{2\sigma_2^2}\right), \tag{35}$$

with $\sigma_2^2 \gg \sigma_1^2$ and $0.05 < \epsilon < 0.1$. This model behaves as a sensor with small variance σ_1^2 most of the time, but occasionally generates a measurement with a large variance σ_2. By taking the negative logarithm of the probability density function, we can obtain a constraint energy which is similar in shape to the weak springs that arise in the weak continuity models of [BZ87]. Similar ideas have been included in many recent vision algorithms using methods from *robust statistics* [Hub81]. Note that with this new constraint energy, the total energy (internal plus external) is no longer quadratic and may therefore have multiple local minima.

In the above two examples, we have assumed that we know which model point $\mathbf{v}(\mathbf{x}_i)$ generated each data point $\mathbf{d}_i$ (in the physically-based analogy, each constraint has a fixed attachment point). In general, this mapping is often unknown. In this case, the probability distribution for a measurement is

$$p(\mathbf{d}_i|\mathbf{v}) = \frac{1}{L} \int_\Omega p_i(\mathbf{d}_i, \mathbf{v}(\mathbf{x}))\, d\mathbf{x}, \tag{36}$$

where $p_i(\mathbf{d}_i, \mathbf{v}(\mathbf{x}))$ is the usual measurement probability distribution (e.g., a Gaussian or contaminated Gaussian), and L is the length or size of Ω. The constraint energy corresponding to this distribution is

$$E_{\text{ff}}(\mathbf{d}_i, \mathbf{v}) = -\log \int_{\Omega} p_i(\mathbf{d}_i, \mathbf{v}(\mathbf{x}))\, d\mathbf{x}. \tag{37}$$

This energy acts like a force field, attracting nearby curve or surface points towards the data point, rather than tying the data to a particular fixed location. When the surface is intrinsically parameterized (as is the case with a snake), the energy equation behaves like a "slippery spring," allowing the curve to slide by the data point. An energy equation similar to (37) has been used for solving the Traveling Salesman Problem [DW87a, DSY89].

C. Uncertainty estimation from posterior models

Another application of probabilistic modeling is the computation of the *uncertainty* (variance or covariance) associated with the posterior estimate $\hat{\mathbf{u}}$. For many physically based models such as the discrete version of the snake with spring constraints, the posterior energy is quadratic

$$E(\mathbf{u}) = \frac{1}{2}(\mathbf{u} - \mathbf{u}^*)^T \mathbf{A}(\mathbf{u} - \mathbf{u}^*) + k, \tag{38}$$

where $\mathbf{u}^* = \mathbf{A}^{-1}\mathbf{b}$ is the minimum energy solution. The Gibbs distribution (21) corresponding to this quadratic form is a multivariate Gaussian with mean $\mathbf{u}^*$ and covariance $\mathbf{A}^{-1}$.

In practice, computing and storing $\mathbf{A}^{-1}$ is not feasible for models of reasonable size, because while $\mathbf{A}$ is sparse and banded, $\mathbf{A}^{-1}$ is not. We can obtain a reduced description of the uncertainty if we only compute the diagonal elements of $\mathbf{A}^{-1}$, i.e., the variance of each nodal variable estimate independently.[3]

Two methods can be used to compute this variance. The first involves computing the values sequentially by solving

$$\mathbf{A}\mathbf{r}_i = \mathbf{e}_i$$

where $e_{ij} = \delta_{ij}$. Each $\mathbf{r}_i$ gives us one column of the covariance matrix $\mathbf{A}^{-1}$, from which we keep only the diagonal element. This method is thus similar to the usual matrix inversion algorithm for $\mathbf{A}$ except that we evaluate each column of $\mathbf{A}^{-1}$ separately to save on storage.

The second method for computing the variance uses a Monte-Carlo approach to generate random samples from the posterior distribution and accumulate the desired statistics. In general, generating good random samples can be tricky [ST89, Sze89]. For the snake energy, however, which is easily decomposed into LDU form, generating an unbiased random sample is straightforward. Substituting $\mathbf{u} = \mathbf{L}^T\mathbf{v}$ into (38), we obtain

$$E(\mathbf{v}) = \frac{1}{2}(\mathbf{v} - \mathbf{v}^*)^T \mathbf{D}(\mathbf{v} - \mathbf{v}^*) + k, \tag{39}$$

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 can be used to compute the local variance at each point, and hence a confidence envelope (Figure 4). A similar approach can be used to any system where *modal analysis* is used [BW76, HP91], since noise can be added independently to each mode.

D. Inverse covariance Riccati equation

To convert the standard *matrix Riccati equation* [Gel74, p. 122]

$$\begin{aligned} \dot{\mathbf{P}} &= \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^T + \mathbf{Q} - \mathbf{G}\mathbf{R}\mathbf{G}^T \\ &= \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^T + \mathbf{Q} - \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P} \end{aligned} \tag{40}$$

into the inverse covariance form, we use the lemma

$$\dot{\mathbf{S}} = -\mathbf{S}\dot{\mathbf{P}}\mathbf{S} \tag{41}$$

[3] We use the notation ρ_{ii}^2 to avoid confusion with the measurement error variances σ_i^2.

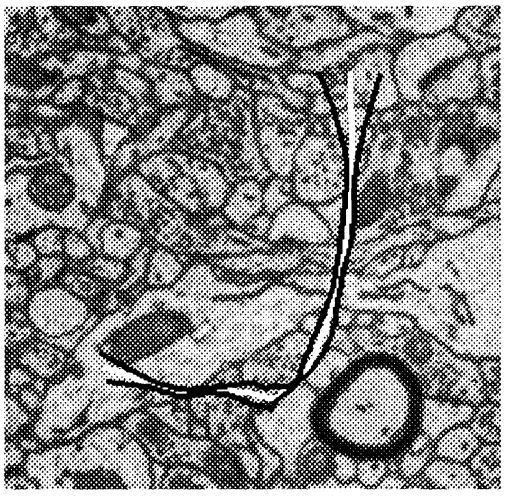


Figure 4: Confidence envelope of snake estimate. The snake is shown in white, and the confidence envelope in black.

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 (40) into (41), we obtain the desired result

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