Shape and Nonrigid Motion Estimation through Physics-Based Synthesis

Dimitri Metaxas, *Member, IEEE*, and Demetri Terzopoulos, *Member, IEEE*

Abstract— **This paper presents a physics-based framework for 3-D shape and nonrigid motion estimation aimed at real-time computer vision. The framework features dynamic models that incorporate the mechanical principles of rigid and nonrigid bodies into conventional geometric primitives. Through the efficient numerical simulation of Lagrange equations of motion, the models can synthesize physically correct behaviors in response to applied forces and imposed constraints. We exploit the shape and motion synthesis capabilities of our models for the purposes of visual estimation. Applying continuous nonlinear Kalman filtering theory, we construct a recursive shape and motion estimator that employs the Lagrange equations as a system model. We interpret the continuous Kalman filter physically: The system model continually synthesizes nonrigid motion in response to generalized forces that arise from the inconsistency between the incoming observations and the estimated model state. The observation forces also account formally for instantaneous uncertainties and incomplete information. A Riccati procedure updates a covariance matrix that transforms the forces in accordance with the system dynamics and prior observation history. The transformed forces modify the translational, rotational, and deformational state variables of the system model to reduce inconsistency, thus producing nonstationary shape and motion estimates from the time-varying visual data. We demonstrate the dynamic estimator in experiments involving model fitting and tracking of articulated and flexible objects from noisy 3-D data.**

Index Terms— **Analysis by synthesis, computer vision, constraints, deformable models, Kalman filtering, nonrigid motion estimation, physics-based modeling.**

I. Introduction

DESPITE THE large body of work on 3-D shape and motion estimation, most robust on-line estimators are limited to rigid objects and simple shapes. Many natural objects, however, cannot be adequately represented in terms of simple shape primitives. Furthermore, natural objects typically undergo motions that are highly nonrigid and subject to various constraints. Animal bodies, for instance, produce astonishingly complex motions, not only as a consequence of their articulated skeletons but also because of soft tissue deformations due to muscle actions and gravitational effects. To cope with the tough challenges of nonrigidity, several researchers have pursued physics-based modeling methods that are capable of synthesizing and/or estimating the shapes and motions of nonrigid 3-D objects to different levels of accuracy [26], [13], [15], [7], [19], [22], [28]. The 2-D problem has received similar attention [17], [11], [25].

This paper presents a physics-based framework for shape and nonrigid motion estimation. Our approach is motivated by real-time vision applications [2]. We build on our prior work with dynamic models that incorporate the mechanical principles of rigid and deformable bodies into conventional geometric modeling primitives [27]. The parameters associated with the geometric primitives, along with global (parameterized) and local (free-form) geometric deformations, acquire physical meanings and are systematically converted into generalized coordinates. Through the application of Lagrangian mechanics and the finite element method, we derive differential equations of motion that govern the dynamics of the models. These equations determine the evolution of the generalized coordinates in response to simulated forces associated with the visual data.

To deal with constrained multipart objects such as articulated anthropomorphic bodies, we introduce hard point-to-point constraints between model parts that should not be violated, regardless of the magnitude of the forces experienced by the parts. Attempting to approximate such constraints simply using stiff springs leads to numerical instability. In our approach, the equations of motion for multipart models include constraint forces that are computed using a stabilized Lagrange multiplier technique.

The kinematics of our models are stylized by the geometric origins of their generalized coordinates. The models can, however, synthesize physically correct motions, given prescribed mass distributions and elasticities. Consequently, our framework supports the physics-based computer animation of nonrigid objects with constraints, as we demonstrate in [20] and [21]. Figs. 1–3 illustrate three examples of real-time nonrigid motion synthesis. The dynamic modeling primitives are deformable superquadrics [27] that interact with one another and their simulated physical environments through point-to-point constraints, gravity, collisions, and friction against impenetrable surfaces. See [20] and [21] for further details.

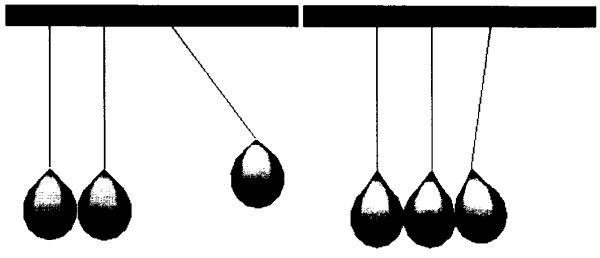
The present paper exploits the constrained nonrigid motion synthesis capabilities of our models in order to estimate shape and motion from incomplete, noisy observations that are available sequentially over time. Applying continuous nonlinear Kalman filtering theory [12], we construct a recursive estimator employing the Lagrange equations of motion as a system model. We interpret the continuous Kalman filter

Manuscript received October 10, 1991; revised December 1, 1992. This work was supported by the Natural Sciences and Engineering Research Council of Canada, the Institute for Robotics and Intelligent Systems, and the Information Technology Research Center. Recommended for acceptance by T. Huang and P. Stucki.

D. Metaxas is with the Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104-6389.

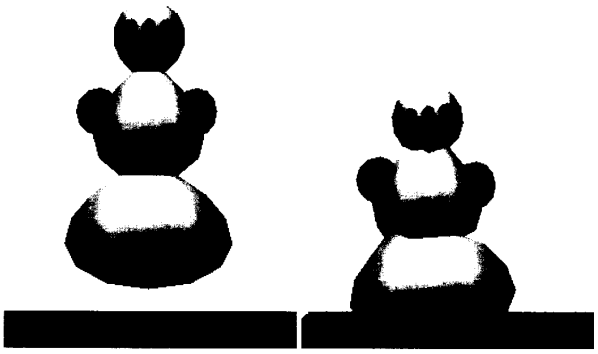
D. Terzopoulos is with the Department of Computer Science, University of Toronto, Toronto, Canada, M5S 1A4.

IEEE Log Number 9209279.



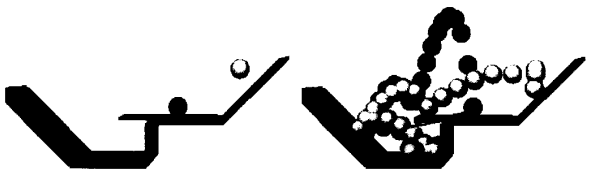
(a) (b)

Fig. 1. Balloon pendulum animation. Balloons are suspended by inextensible strings: (a) Initial state; (b) balloons swing and collide against each other until kinetic energy is dissipated.



(a) (b)

Fig. 2. Dynamic snowman: (a) Body parts are connected with point-to-point constraints; (b) snowman locomotes through controlled bouncing on floor.



(a) (b)

Fig. 3. Ball-and-springboard circus stunt. Two balls interact with apivoting springboard mounted on immobile planes: Initial state; (b) strobed motion tracks.

physically: The system model continually synthesizes nonrigid motion in response to generalized forces that arise from inconsistencies between the state variables and the incoming observations. The observation forces account formally for instantaneous uncertainties in the data. A Riccati procedure updates an error covariance matrix that transforms the forces in accordance with the system dynamics and the prior observation history. The transformed forces induce changes in the translational, rotational, and deformational state variables of the system model, thereby reducing the inconsistency with the observations. Thus, the system model synthesizes nonstationary shape and motion estimates in response to the visual data.

Kalman filtering techniques have been applied in the vision literature to the estimation of surface depth [18], [14], dynamic features [8], and rigid motion parameters [10], [5], [6] of objects from image sequences. To date, these Kalman filters have been discrete filters with the simplest possible system models, which are usually constant velocity models. Additional restrictions such as constant error covariance matrices may also be found in the literature [22]. These sorts of simplifications can severely limit the ability of an estimator to recover shape and motion parameters accurately and robustly from realistic data, especially when confronted with articulated or fully nonrigid motion.

By contrast, the continuous Kalman filters that we develop in this paper incorporate sophisticated Lagrange equations of 3-D nonrigid motion as system models (simpler dynamic models for the estimation of rigid body motion parameters [9], [10] or 2-D nonrigid motion parameters [25] are also available). To gain efficiency with minimal loss of accuracy, we design an efficient large-scale Kalman estimator through state decoupling. Our work establishes a direct connection with existing dynamic vision models derived from physical principles. We show, for example, that the dynamic tracking scheme proposed in [17] and [26] that is applied to deformable superquadrics in [27] amounts to a degenerate "Kalman filter" with a constant unit error covariance matrix.

In earlier work we explored efficient, first-order dynamic models [27], [19]. First-order dynamics suffices in reconstruction applications where static shape representation is paramount. The model formulations presented in Section II and constraint formulations presented in Section III emphasize general, second-order dynamics. The inertial properties of second-order models can be exploited to track moving objects more robustly. Section IV formulates the dynamic nonrigid motion Kalman estimator. Section V discusses implementation issues. Section VI presents selected experimental results demonstrating our algorithms. The experiments involve model fitting and tracking of articulated and flexible objects from noise-corrupted 3-D data. Section VII draws conclusions from our work.

II. Dynamic Models with Global and Local Deformations

This section reviews and extends the systematic technique that we developed in [27] for transforming geometric primitives and deformations into dynamic models. We extend the formulation to solid models that include parameterized global deformations such as tapers and bends. We again exemplify the approach with deformable superquadric models (see also [19]). We also review the concept of dynamic shape fitting and motion tracking, which is a precursor to the more powerful Kalman estimator developed in Section IV.

A. Geometry

In general, our models are 3-D solids whose intrinsic (material) coordinates $\mathbf{u} = (u, v, w)$ are defined in a domain Ω. The positions of points on the model relative to an inertial frame of reference Φ in space are given by a vector-valued, time-varying function $\boldsymbol{x}(\mathbf{u}, t) = (x_1(\mathbf{u}, t), x_2(\mathbf{u}, t), x_3(\mathbf{u}, t))^T$, where ^{T} denotes transposition. Referring to Fig. 4, we set up

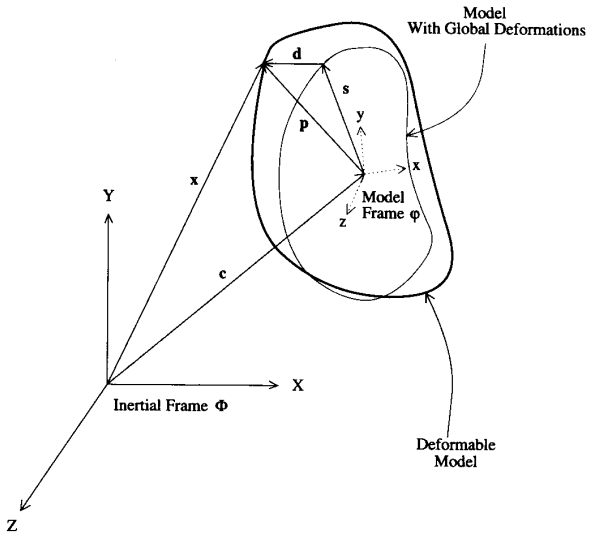


Fig. 4. Geometry of deformable model.

a noninertial, model-centered reference frame ϕ and express the position function as

$$\boldsymbol{x} = \boldsymbol{c} + \boldsymbol{R}\boldsymbol{p} \tag{1}$$

where $\boldsymbol{c}(t)$ is the origin of ϕ at the center of the model, and the rotation matrix $\boldsymbol{R}(t)$ gives the orientation of ϕ relative to Φ. Thus, $\boldsymbol{p}(\mathrm{u}, t)$ gives the positions of points on the model relative to the model frame.

We further express

$$\boldsymbol{p} = \boldsymbol{s} + \boldsymbol{d} \tag{2}$$

as the sum of a reference shape $\boldsymbol{s}(\mathrm{u}, t)$ and a displacement $\boldsymbol{d}(\mathrm{u}, t)$. We define the reference shape as

$$\boldsymbol{s} = \boldsymbol{T}(\boldsymbol{e}(\mathrm{u}; a_0, a_1, \ldots); b_0, b_1, \ldots). \tag{3}$$

Here, a geometric primitive $\boldsymbol{e}$, which is defined parametrically in u and parameterized by the variables a_i, is subjected to the *global deformation* $\boldsymbol{T}$, which depends on the parameters b_i. Although generally nonlinear, $\boldsymbol{e}$ and $\boldsymbol{T}$ are assumed to be differentiable (so that we may compute the Jacobian of $\boldsymbol{s}$), and $\boldsymbol{T}$ may be a composite sequence of primitive deformation functions $\boldsymbol{T}(\boldsymbol{e}) = \boldsymbol{T}_1(\boldsymbol{T}_2(\ldots \boldsymbol{T}_n(\boldsymbol{e})))$. We concatenate the global deformation parameters into the vector

$$\boldsymbol{q}_s = (a_0, a_1, \ldots, b_0, b_1, \ldots)^T. \tag{4}$$

Next, we express the displacement as a linear combination of basis functions. The basis functions can be local or global; however, finite element shape functions [16] are the natural choice for representing *local deformations*

$$\boldsymbol{d} = \boldsymbol{S}\boldsymbol{q}_d \tag{5}$$

where $\boldsymbol{S}$ is a shape matrix whose entries are the shape functions and

$$\boldsymbol{q}_d = (\ldots, \boldsymbol{d}_i^T, \ldots)^T \tag{6}$$

is the vector of local deformation parameters. See [27] for details.

B. Kinematics and Dynamics

Differentiating (1), the velocity of points on the model is given by

$$\begin{aligned}\dot{\boldsymbol{x}} &= \dot{\boldsymbol{c}} + \dot{\boldsymbol{R}}\boldsymbol{p} + \boldsymbol{R}\dot{\boldsymbol{p}} \\ &= \dot{\boldsymbol{c}} + \boldsymbol{B}\dot{\boldsymbol{\theta}} + \boldsymbol{R}\dot{\boldsymbol{s}} + \boldsymbol{R}\boldsymbol{S}\dot{\boldsymbol{q}}_d\end{aligned} \tag{7}$$

where $\boldsymbol{\theta} = (\ldots, \theta_i, \ldots)^T$ is the vector of rotational parameters of the model and $\boldsymbol{B} = [\ldots \partial(\boldsymbol{R}\boldsymbol{p})/\partial\theta_i \ldots]$ [27]. Furthermore, $\dot{\boldsymbol{s}} = [\partial \boldsymbol{s}/\partial \boldsymbol{q}_s]\dot{\boldsymbol{q}}_s = \boldsymbol{J}\dot{\boldsymbol{q}}_s$, where $\boldsymbol{J}$ is the Jacobian of the reference shape with respect to the global deformation parameter vector. We can therefore write the model kinematics compactly as

$$\boldsymbol{x} = \boldsymbol{c} + \boldsymbol{R}(\boldsymbol{s} + \boldsymbol{d}) = \boldsymbol{\xi}(\boldsymbol{q}), \tag{8}$$

$$\dot{\boldsymbol{x}} = [\boldsymbol{I}\boldsymbol{B}\boldsymbol{R}\boldsymbol{J}\boldsymbol{R}\boldsymbol{S}]\dot{\boldsymbol{q}} = \boldsymbol{L}\dot{\boldsymbol{q}}. \tag{9}$$

Here

$$\boldsymbol{q} = (\boldsymbol{q}_c^T, \boldsymbol{q}_\theta^T, \boldsymbol{q}_s^T, \boldsymbol{q}_d^T)^T \tag{10}$$

with $\boldsymbol{q}_c = \boldsymbol{c}$ and $\boldsymbol{q}_\theta = \boldsymbol{\theta}$ serves as the vector of generalized coordinates of the model.

To make the kinematic model dynamic, we assume that it is made of a simulated material that has a mass distribution $\mu(\mathrm{u})$ and is subject to frictional damping. We also assume that the material may deform elastically (or, more generally, viscoelastically). From Lagrangian mechanics, we obtain second-order equations of motion that take the form

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{g}_q + \boldsymbol{f}_q \tag{11}$$

(see [27] and the Appendix for derivations). The mass matrix $\boldsymbol{M} = \int \mu \boldsymbol{L}^T \boldsymbol{L}\, d\mathrm{u}$. The stiffness matrix $\boldsymbol{K}$ may be obtained from a deformation strain energy (a quadratic form $(\boldsymbol{q}^T\boldsymbol{K}\boldsymbol{q})/2$). The Raleigh damping matrix $\boldsymbol{D} = \alpha\boldsymbol{M} + \beta\boldsymbol{K}$. The generalized inertial forces $\boldsymbol{g}_q = -\int \mu \boldsymbol{L}^T \dot{\boldsymbol{L}}\dot{\boldsymbol{q}}\, d\mathrm{u}$ include generalized centrifugal, Coriolis, and transverse forces due to the dynamic coupling between $\boldsymbol{q}_\theta$, $\boldsymbol{q}_s$, and $\boldsymbol{q}_d$. Finally, $\boldsymbol{f}_q = \int \boldsymbol{L}^T \boldsymbol{f}\, d\mathrm{u}$ are generalized external forces associated with the components of $\boldsymbol{q}$, where $\boldsymbol{f}(\mathrm{u}, t)$ is the force distribution applied to the model.

For convenience, we rewrite the second-order equations (11) in standard dynamical system form, as the coupled set of first-order equations

$$\dot{\boldsymbol{u}} = \boldsymbol{F}\boldsymbol{u} + \boldsymbol{g} \tag{12}$$

with state vector $\boldsymbol{u}$, system matrix $\boldsymbol{F}$, and driving function $\boldsymbol{g}$ as follows:

$$\boldsymbol{u} = \begin{bmatrix}\dot{\boldsymbol{q}} \\ \boldsymbol{q}\end{bmatrix}, \quad \boldsymbol{F} = \begin{bmatrix}-\boldsymbol{M}^{-1}\boldsymbol{D} & -\boldsymbol{M}^{-1}\boldsymbol{K} \\ \boldsymbol{I} & \boldsymbol{0}\end{bmatrix},$$
$$\boldsymbol{g} = \begin{bmatrix}\boldsymbol{M}^{-1}(\boldsymbol{g}_q + \boldsymbol{f}_q) \\ \boldsymbol{0}\end{bmatrix}. \tag{13}$$

C. Dynamic Estimation

We take a physics-based approach to visual estimation. The visual data are converted into traction forces that act on the dynamic model [26], [27]. We fit the model by integrating the differential equations of motion (12) driven by these forces.

More specifically, let the observation vector $\boldsymbol{z}(t)$ denote time-varying input data. We can relate $\boldsymbol{z}(t)$ to the model's state vector $\boldsymbol{u}(t)$ through the nonlinear observation equation

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{u}) + \boldsymbol{v} \tag{14}$$

where $\boldsymbol{v}(t)$ represents uncorrelated measurement errors as a zero mean white noise process with known covariance $\boldsymbol{V}(t)$, i.e., $\boldsymbol{v}(t) \sim \boldsymbol{N}(\boldsymbol{0}.\boldsymbol{V}(t))$.[1] In dynamic estimation, the rate of change of the estimated model state vector $\hat{\boldsymbol{u}}$ is given from (12) by

$$\dot{\hat{\boldsymbol{u}}} = \boldsymbol{F}\hat{\boldsymbol{u}} + \boldsymbol{g}. \tag{15}$$

Furthermore, according to (9), (8), and (14), the driving function $\boldsymbol{g}$ in (13) includes the term

$$\boldsymbol{f}_q = \boldsymbol{H}^T\boldsymbol{V}^{-1}(\boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{u}})) \tag{16}$$

where

$$\boldsymbol{H} = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}=\hat{\boldsymbol{u}}}. \tag{17}$$

Matrix $\boldsymbol{H}$ maps the three-space observation forces $(\boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{u}}))$ scaled by $\boldsymbol{V}^{-1}$ to $\boldsymbol{q}$-space generalized forces $\boldsymbol{f}_q$.[2] If the data are very noisy, the entries of $\boldsymbol{V}$ have large values, yielding small generalized forces, hence, nominal changes in $\boldsymbol{q}$. If the data are accurate, $\boldsymbol{V}$ will have small entries, and the generalized forces will have a significant effect on the model.

When tracking moving objects, the tracked object may become temporarily occluded in whole or in part. The second-order model (11) (i.e., (12) and (13)) is appropriate for dealing with occlusion since its mass provides inertia; the generalized coordinates continue to evolve even when the data forces $\boldsymbol{f}_q$ vanish. Since the model extrapolates in time, it stands a better chance of regaining lock on the object when it becomes disoccluded.

For static shape reconstruction problems (i.e., $\boldsymbol{z}(t) = \boldsymbol{z}$), it makes sense to simplify the motion equations by setting the mass density to zero. Lacking inertia, the model will come to rest as soon as all the internal and applied forces equilibrate. Since $\boldsymbol{M}$ and $\boldsymbol{g}_q$ are zero when $\mu(\mathrm{u}) = 0$, (11) reduces to the first-order system

$$\boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{f}_q \tag{18}$$

and in (12) and (15)

$$\boldsymbol{u} = \boldsymbol{q}. \quad \boldsymbol{F} = -\boldsymbol{D}^{-1}\boldsymbol{K}. \quad \boldsymbol{g} = \boldsymbol{D}^{-1}\boldsymbol{f}_q. \tag{19}$$

[1] For example, if $\boldsymbol{z}$ consists of observations of time-varying positions of model points at material coordinates u_k on the model's surface, the components of $\boldsymbol{h}$ are computed using (8) evaluated at u_k. If $\boldsymbol{z}$ includes velocity estimates, then we also use (9).

[2] In particular, for the function $\boldsymbol{h}$, which is associated with observations of model positions at material coordinates u_k, then $\boldsymbol{H}$ is a matrix whose entries are computed using $\boldsymbol{L}|_{\boldsymbol{u}=\hat{\boldsymbol{u}}}$ evaluated at u_k.

D. Example: Deformable Superquadrics

For concreteness, we consider the case of superquadric ellipsoids with tapering and bending deformations [3], [24]. The parametric equation of a superquadric ellipsoid solid $\boldsymbol{e} = (e_1, e_2, e_3)^T$ is

$$\boldsymbol{e} = a_0 w \begin{pmatrix} a_1 C_u{}^{\epsilon_1} C_v{}^{\epsilon_2} \\ a_2 C_u{}^{\epsilon_1} S_v{}^{\epsilon_2} \\ a_3 S_u{}^{\epsilon_1} \end{pmatrix} \tag{20}$$

where $-\pi/2 \le u \le \pi/2$, $-\pi \le v < \pi$, and $0 \le w \le 1$, and where $S_\theta{}^\epsilon = \mathrm{sgn}(\sin\theta)|\sin\theta|^\epsilon$, and $C_\theta{}^\epsilon = \mathrm{sgn}(\cos\theta)|\cos\theta|^\epsilon$, respectively. Here, $a \ge 0$ is a scale parameter, $0 \le a_1, a_2, a_3 \le 1$ are aspect ratio parameters, and $\epsilon_1, \epsilon_2 \ge 0$ are "squareness" parameters.

We combine linear tapering along the z axis and bending along the x axis of the superquadric $\boldsymbol{e}$ into a single parameterized deformation and express the reference shape $\boldsymbol{s} = \boldsymbol{T}(\boldsymbol{e})$ as

$$\boldsymbol{s} = \begin{pmatrix} (\frac{t_1 e_3}{a_0 a_3} + 1)e_1 + b_1 \cos(\frac{e_3 + b_2}{a_0 a_3}\pi b_3) \\ (\frac{t_2 e_3}{a_0 a_3} + 1)e_2 \\ e_3 \end{pmatrix} \tag{21}$$

where $-1 \le t_1, t_2 \le 1$ are the tapering parameters in the x and y axes, respectively, b_1 defines the magnitude of the bending and can be positive or negative, $-1 \le b_2 \le 1$ defines the location on the z axis where bending is applied, and $0 < b_3 \le 1$ defines the region of influence of bending. This method of incorporating global deformations is not restricted to only tapering and bending deformations. Any other deformation that can be expressed as a differentiable function can be incorporated into the global deformations in a similar way. We collect the global deformation parameters in $\boldsymbol{s}$ into vector $\boldsymbol{q}_s = (a_0, a_1, a_2, a_3, \epsilon_1, \epsilon_2, t_1, t_2, b_1, b_2, b_3)^T$. The Appendix gives the expression of the Jacobian $\boldsymbol{J}$ needed to compute the velocity (9).

We have implemented efficient deformable superquadric shells where the material domain is restricted to a membrane surface $\mathrm{u} = (u, v, 1)$ and the interior mass density $\mu(\mathrm{u}) = 0$ for $0 \le w < 1$. We triangulate the surface of the model into linear elastic elements. The elements have nodes at their vertices, and each generalized coordinate $\boldsymbol{q}_i$ in (6) denotes a displacement vector associated with node i of the model. We derive the stiffness matrix $\boldsymbol{K}$ in (11) from a loaded membrane deformation energy whose two parameters $w_0 \ge 0$ and $w_1 \ge 0$ control the local magnitude and variation of the deformation, respectively. See [27] for details.

III. Constrained Nonrigid Motion

We now extend (11) to account for the motions of composite models with interconnected deformable parts that are constrained not to separate. Shabana [23] describes the well-known Lagrange multiplier method for multibody systems [29]. We form a composite generalized coordinate vector $\boldsymbol{q}$ and force vectors $\boldsymbol{g}_q$ and $\boldsymbol{f}_q$ for an n-part model by concatenating the $\boldsymbol{q}_i$, $\boldsymbol{g}_{q_i}$, and $\boldsymbol{f}_{q_i}$ associated with each part $i = 1, \ldots, n$. Similarly, the composite matrices $\boldsymbol{M}$, $\boldsymbol{D}$, and $\boldsymbol{K}$ for the n-part model are block diagonal matrices with submatrices $\boldsymbol{M}_i$, $\boldsymbol{D}_i$,

and K_i, respectively, for each part i. Given a set of constraint equations

$$C(q,t) = 0 \tag{22}$$

where $C = [C_1^T, C_2^T, \ldots, C_k^T]^T$ expresses k holonomic constraints among the n parts of the model, the Lagrange multiplier method prescribes the composite equations of motion

$$M\ddot{q} + D\dot{q} + Kq = g_q + f_q - C_q^T\lambda \tag{23}$$

where $f_{g_c} = -C_q^T\lambda$ are generalized constraint forces acting among the parts. The term C_q^T is the transpose of the constraint Jacobian matrix, and $\lambda = (\lambda_1^T, \ldots, \lambda_n^T)^T$ is a vector of Lagrange multipliers that must be determined.

Equation (23) comprises fewer equations than unknowns. To obtain the additional equations, we differentiate (22) twice with respect to time

$$\ddot{C}(q,t) = 0 \tag{24}$$

yielding $C_q\ddot{q} + C_{tt} + (C_q\dot{q})q\dot{q} + 2C_{q_t}\dot{q} = 0$. Rearranging terms, we get

$$\gamma = C_q\ddot{q} = -C_{tt} - (C_q\dot{q})q\dot{q} - 2C_{q_t}\dot{q}. \tag{25}$$

Appending this equation to (23) and rearranging terms, we arrive at the augmented equations of motion

$$\begin{bmatrix} M & C_q^T \\ C_q & 0 \end{bmatrix}\begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} -D\dot{q} - Kq + g_q + f_q \\ \gamma \end{bmatrix}. \tag{26}$$

In principle, these equations may be integrated from initial conditions $q(0)$ and $\dot{q}(0)$ satisfying $C(q(0),0) = 0$ and $\dot{C}(q(0),0) = 0$.

There are two practical problems in applying (26) to model-based visual estimation. First, the constraints must be satisfied initially. Unfortunately, partial information and errors in the data make it unlikely that the parameter values of the various parts can be initialized such that the parts satisfy the constraints (i.e., $C(q,0) \neq 0$). Second, even if the constraints may be satisfied at a given time step of the dynamic estimation process (i.e., $C(q,t) = 0$), they may not be satisfied at the next time step (i.e., $C(q,t+\Delta t) \neq 0$) because of numerical integration errors, noise, etc. In fact, $C(q,t)$ may grow without bounds.

To remedy these two problems, we apply a method proposed by Baumgarte that stabilizes the constrained equations through linear feedback control [1], [29]. The method replaces the differential equation (24) with equations that have similar solutions but are asymptotically stable in the sense of Ljapunov, such as the damped second-order differential equations

$$\ddot{C} + 2\alpha\dot{C} + \beta^2 C = 0 \tag{27}$$

with stabilization factors α and β. This replaces the lower entry of the vector on the RHS of (26) with $\gamma - 2\alpha\dot{C} - \beta^2 C$. We choose $\beta = \alpha$ to obtain the critically damped solution $C(q,0)e^{-\alpha t}$ that, for a given value α, has the quickest asymptotic decay towards constraint satisfaction $C = 0$. A caveat in applying the constraint stabilization method is that it introduces additional eigenfrequencies into the dynamical system. Increasing α in an attempt to increase the rate of constraint satisfaction will eventually lead to numerically stiff constrained motion equations dominated by the stabilizing terms.

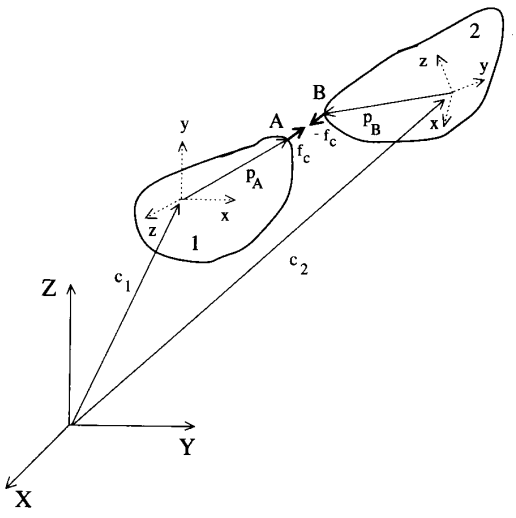


Fig. 5. Point-to-point constraint.

A. Fast Point-to-Point Constraint Force Computation

The Lagrange multiplier method is general but potentially expensive for our models since the matrix in (26) can be large, according to the number of finite elements in the discrete model. We have devised a specialized solver for the unknown constraint forces f_{g_c} in the case of point-to-point constraints. The specialized method requires the solution of linear systems of size proportional to the number of constraints, which is usually small. In this sense, it is similar to the dynamic constraint technique of [4], but it is suitable for nonrigid parts. We derived the method for first-order dynamic systems in [19], and we extend it to second-order systems below.

1) Single Constraint: Fig. 5 illustrates two parts: 1 and 2. We constrain points A and B to remain in contact and must compute the necessary constraint forces $f_c(t)$ at point A and $-f_c(t)$ at point B. From (11), the motion equations of the parts are

$$\ddot{q}_1 = M_1^{-1}(g_{q_1} + f_{q_1} + f_{g_{c_A}} - K_1q_1 - D_1\dot{q}_1) \tag{28}$$

$$\ddot{q}_2 = M_2^{-1}(g_{q_2} + f_{q_2} + f_{g_{c_B}} - K_2q_2 - D_2\dot{q}_2) \tag{29}$$

where the generalized constraint forces at points A and B are, respectively

$$f_{g_{c_A}} = L_A^T f_c, \qquad f_{g_{c_B}} = -L_B^T f_c \tag{30}$$

and L_A, L_B are computed using (9).

From (9) and (8), the constraint equation and its time derivatives are

$$\begin{aligned} C &= x_A - x_B = (c_1 + R_1p_A) - (c_2 + R_2p_B) \\ \dot{C} &= L_A\dot{q}_1 - L_B\dot{q}_2 \\ \ddot{C} &= L_A\ddot{q}_1 + \dot{L}_A\dot{q}_1 - L_B\ddot{q}_2 - \dot{L}_B\dot{q}_2. \end{aligned} \tag{31}$$

Replacing these expressions into the Baumgarte equation (27) (with $\alpha = \beta$), we obtain the linear equations

$$\boldsymbol{N}\boldsymbol{f}_c + \boldsymbol{r} = \boldsymbol{0} \tag{32}$$

where the 3×3 matrix $\boldsymbol{N}$ is

$$\boldsymbol{N} = (\boldsymbol{L}_A \boldsymbol{M}_1^{-1} \boldsymbol{L}_A^T + \boldsymbol{L}_B \boldsymbol{M}_2^{-1} \boldsymbol{L}_B^T) \tag{33}$$

and the vector $\boldsymbol{r}$ is

$$\begin{aligned} \boldsymbol{r} = {} & \dot{\boldsymbol{L}}_A \dot{\boldsymbol{q}}_1 - \dot{\boldsymbol{L}}_B \dot{\boldsymbol{q}}_2 + 2\alpha \dot{\boldsymbol{C}} + \alpha^2 \boldsymbol{C} + \\ & \boldsymbol{L}_A \boldsymbol{M}_1^{-1} (\boldsymbol{g}_{q_1} + \boldsymbol{f}_{q_1} - \boldsymbol{K}_1 \boldsymbol{q}_1 - \boldsymbol{D}_1 \dot{\boldsymbol{q}}_1) - \\ & \boldsymbol{L}_B \boldsymbol{M}_2^{-1} (\boldsymbol{g}_{q_2} + \boldsymbol{f}_{q_2} - \boldsymbol{K}_2 \boldsymbol{q}_2 - \boldsymbol{D}_2 \dot{\boldsymbol{q}}_2). \end{aligned} \tag{34}$$

2) Multiple Constraints: For multiple point-to-point constraints, the constraint force computation must take into account all of the constraint forces acting on the various parts of the model. This requires the solution of a system of equations whose size is on the order of the total number of constraints.

Suppose we specify k constraints among model parts. Let $\boldsymbol{f}_{c_i}$ be the constraint force for constraint i. We assemble the multipart model's constraint force vector $\boldsymbol{f}_c = (\boldsymbol{f}_{c_1}^T, \boldsymbol{f}_{c_2}^T, \ldots, \boldsymbol{f}_{c_k}^T)^T$ and express the equation for constraint forces $\boldsymbol{f}_{c_i}$ as in (32): $\boldsymbol{N}_i \boldsymbol{f}_c + \boldsymbol{r} = \boldsymbol{0}$, where $\boldsymbol{r} = (\boldsymbol{r}_1^T, \boldsymbol{r}_2^T, \ldots, \boldsymbol{r}_k^T)^T$, and $\boldsymbol{N}_i$ is a $3k \times 3k$ matrix. The pattern of nonzero entries in $\boldsymbol{N}_i$ depends on the connectivity of the parts. Assembling the k systems, we arrive at a composite system in the form of (32), where $\boldsymbol{N} = \sum_{i=1}^{k} \boldsymbol{N}_i$.

IV. Kalman Filter Formulation

In dynamic estimation, the data forces (16) take into account the current observations only. This section proposes a more sophisticated, recursive estimator that transforms the generalized forces through an error covariance matrix before applying them to the model. The covariance matrix takes into account the modeling uncertainty, along with the history of prior observations and their uncertainties.

Our estimator is a *continuous* extended Kalman filter. The basic idea behind Kalman filtering is to perform optimal least squares estimation recursively as new measurements arrive, making use of nonstationary models known as system models [12]. We can exploit our dynamic nonrigid object models within the Kalman framework by treating their differential equations of motion as system models.

To do this, we assume that vectors $\boldsymbol{w}(t)$ and $\boldsymbol{v}(t)$ represent uncorrelated modeling and measurement errors, respectively, as zero mean white noise processes with known covariances, i.e., $\boldsymbol{w}(t) \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{Q}(\boldsymbol{t}))$ and $\boldsymbol{v}(t) \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{V}(\boldsymbol{t}))$.[3] In view of (12) and (14), the nonlinear Kalman filter equations for our dynamic model take the form

$$\begin{aligned} \dot{\boldsymbol{u}} &= \boldsymbol{F}\boldsymbol{u} + \boldsymbol{g}_1 + \boldsymbol{w} \\ \boldsymbol{z} &= \boldsymbol{h}(\boldsymbol{u}) + \boldsymbol{v} \end{aligned} \tag{35}$$

[3] Kalman filtering is optimal for linear system and measurement models, assuming the associated noise processes are Gaussian [12]. The Gaussian noise assumption is unrealistic in many applications. In practice, however, all we can often economically measure about the characteristics of a noise process is its autocorrelation function; hence, a Gaussian model is a convenient choice.

where

$$\boldsymbol{g}_1 = \begin{bmatrix} \boldsymbol{M}^{-1}(\boldsymbol{g}_q + \boldsymbol{f}_{g_c}) \\ \boldsymbol{0} \end{bmatrix} \tag{36}$$

($\boldsymbol{g}_1 = \boldsymbol{D}^{-1}\boldsymbol{f}_{g_c}$ for the first-order model (18)). The vector $\boldsymbol{g}_1$, which generally includes generalized inertial and constraint forces, plays the role of deterministic disturbances in the system model. The state estimation equation for uncorrelated system and measurement noises (i.e., $E[\boldsymbol{w}(t)\boldsymbol{v}^T(\tau)] = 0$) is

$$\dot{\hat{\boldsymbol{u}}} = \boldsymbol{F}\hat{\boldsymbol{u}} + \boldsymbol{g}_1 + \boldsymbol{P}\boldsymbol{H}^T\boldsymbol{V}^{-1}(\boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{u}})) \tag{37}$$

where $\boldsymbol{H}$ is computed using (17). The expression $\boldsymbol{G}(t) = \boldsymbol{P}\boldsymbol{H}^T\boldsymbol{V}^{-1}$ is known as the Kalman gain matrix. The symmetric error covariance matrix $\boldsymbol{P}(t)$ is the solution of the matrix Riccati equation

$$\dot{\boldsymbol{P}} = \boldsymbol{F}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{F}^T + \boldsymbol{Q} - \boldsymbol{P}\boldsymbol{H}^T\boldsymbol{V}^{-1}\boldsymbol{H}\boldsymbol{P}. \tag{38}$$

Note that the term $\boldsymbol{P}\boldsymbol{H}^T\boldsymbol{V}^{-1}(\boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{u}}))$ in (37) may be interpreted as a generalized force. It results from the instantaneous disparity between the measurements $\boldsymbol{z}$ and the estimated state of the model $\hat{\boldsymbol{u}}$. For unit covariance matrix $\boldsymbol{P}(t) = \boldsymbol{I}$, this term reduces to the generalized data forces (16), which play a role in dynamic estimation [27], [26].

The improvement offered by the Kalman filter dynamic estimator can be explained intuitively. The covariance matrix $\boldsymbol{P}(t)$ comprises a time-varying measure of the uncertainty in the estimate $\hat{\boldsymbol{u}}$ [12]. The measure depends on current and prior observations, system dynamics, and modeling errors. Consequently, the Kalman gain matrix $\boldsymbol{G}$ is "proportional" to the uncertainty in the estimate and "inversely proportional" to the measurement noise. If, on the one hand, there is significant measurement noise, and the state estimate errors are small, the term in parentheses in (37) is due mainly to noise, and only modest changes in the state estimates should be made. On the other hand, small measurement noise and large uncertainty in the state estimates suggest that the same term contains significant new information. Therefore, the difference between the actual and the predicted measurement will be used as a basis for making strong corrections to the estimates.

V. Implementation

A. Integrating the Unconstrained Motion Equations

The equations of motion of the model are numerically well conditioned. Our approach partitions complicated nonrigid motions into rigid-body motions, global deformations, and local deformations.[4] This partitioning improves the speed and stability of numerical simulation algorithms. We achieve interactive response by employing explicit methods to integrate (12). The simplest method is the first-order Euler procedure that updates the state vector over a time step from time t to $t + \Delta t$ according to the formula

$$\boldsymbol{u}^{(t+\Delta t)} = \boldsymbol{u}^{(t)} + \Delta t(\boldsymbol{F}^{(t)}\boldsymbol{u}^{(t)} + \boldsymbol{g}^{(t)}). \tag{39}$$

[4] A represented shape is partitioned into pose, abstract form, and shape details.

The following details about our numerical solution are noteworthy. First, we represent the rotation component of the models using quaternions (see the Appendix). This simplifies the updating of $\boldsymbol{q}_\theta$ and $\dot{\boldsymbol{q}}_\theta$. Second, the explicit Euler method does not require the assembly and factorization of a finite element stiffness matrix, as is common practice when applying the finite element method [16]. Rather, we compute the product $\boldsymbol{Kq}$ very efficiently in an element-by-element fashion for the local deformation coordinates $\boldsymbol{q}_d$ [27]. Third, for added efficiency, we may lump masses to obtain a diagonal $\boldsymbol{M}$, and we may assume mass-proportional damping, i.e., $\boldsymbol{D} = \alpha \boldsymbol{M}$, where α is the damping coefficient [16].

B. Integrating the Constrained Motion Equations

In integrating the constrained motion equations, at each time step, we may solve (26) for $\ddot{\boldsymbol{q}}^{(t)}$ and $\boldsymbol{\lambda}^{(t)}$ with known $\boldsymbol{q}^{(t)}$ and $\dot{\boldsymbol{q}}^{(t)}$, and then, we integrate $\ddot{\boldsymbol{q}}$ and $\dot{\boldsymbol{q}}$ from t to $t + \Delta t$ to obtain $\dot{\boldsymbol{q}}^{(t+\Delta t)}$ and $\boldsymbol{q}^{(t+\Delta t)}$, respectively, using the Euler method (39).

In applying the fast point-to-point constraint algorithm, at each time step, we assemble $\boldsymbol{N}$ and $\boldsymbol{r}$, as was explained in Section III-A, and compute the constraint forces $\boldsymbol{f}_c$ by solving (32) using LU factorization. We then augment the equations of motion of each part with the applicable constraint forces, and we integrate using the Euler method as before.

C. Kalman Filter Implementation

The continuous Kalman estimation equations (37) and (38) can be integrated using standard numerical techniques, where the simplest is Euler's method.

The cost of computing (38) can be large because the size of the error covariance matrix is proportional to the size of $\boldsymbol{q}_d$, which depends on the number of finite elements used to discretize the model. We take a common approach to implementing practical large-scale Kalman filters: suboptimal filter design [12]. There are several ways to simplify the Kalman filter equations, such as decoupling state variables, deleting state variables, simplifying filter gains, etc. Suitable simplifications are often based on knowledge about the particular system model used in the Kalman filter equations. They can lead to a significantly reduced computational burden with little reduction in estimation accuracy.

We simplify the Kalman filter equations by decoupling state variables. A natural decoupling is suggested by the structure of the state vector $\boldsymbol{u}$ in (12), which is comprised of the translation ($\boldsymbol{q}_c$, $\dot{\boldsymbol{q}}_c$), rotation ($\boldsymbol{q}_\theta$, $\dot{\boldsymbol{q}}_\theta$), global deformation ($\boldsymbol{q}_s$, $\dot{\boldsymbol{q}}_s$), and local deformation ($\boldsymbol{q}_d$, $\dot{\boldsymbol{q}}_d$) state variables for the various part models. To decouple (38), we ignore the off-diagonal block submatrices of $\boldsymbol{P}$ that specify covariances among these different sets of state variables. Each set can then be updated independently. Note that the submatrix of $\boldsymbol{P}$ associated with the local deformations is structured like the stiffness matrix $\boldsymbol{K}$ [27], i.e., it is a sparse matrix that may be updated on a per-element basis. Note that the error covariance submatrices associated with the elements are full symmetric matrices since the elemental stiffness submatrices have the same structure, i.e., the nodal displacements within each element are interdependent.

For additional savings, we may assume independent modeling and measurement errors. This assumption leads to covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{V}$, which are scalar multiples of the identity matrix. Note that although the entries of the measurement covariance matrix $\boldsymbol{V}$ are often known, it is not easy to determine substantive values for the system covariance matrix $\boldsymbol{Q}$ [12]. The latter are therefore treated as filter tuning parameters. If the entries of $\boldsymbol{Q}$ are inappropriate for the particular application, the filter may diverge or converge to the wrong value [6]. This is due to the nonlinear measurement equations (35). The entries of $\boldsymbol{Q}$ must therefore be tuned such that the filter exhibits the desired operating characteristics and converges to correct estimates.

VI. Experiments

We have carried out various shape and nonrigid motion estimation experiments utilizing 3-D human motion and synthetic data. The synthetic data sets consist of time-varying 3-D positions of points sampled from the surfaces of synthesized deformable superquadrics undergoing nonrigid motion in response to externally applied forces. The human motion data were collected using WATSMART, which is a commercial noncontact, 3-D motion digitizing and analysis system. Using multiple optoelectric measurement cameras, WATSMART can track as many as 64 infrared light-emitting diode markers attached to various body parts of a moving subject. WATSMART produces 3-D coordinates of the markers at sampling rates of 19 to 400 Hz. At least two cameras must see a marker before its 3-D coordinates can be calculated using a direct linear transformation technique. Our data were collected using four cameras and 32 markers at a 50-Hz sampling rate.

In the experiments, we couple the models to the data points, which are indicated by dark dots in the forthcoming figures, by searching for the node on the model that is nearest to each datapoint.[5] This is a simple and robust, albeit brute-force, method for assigning data points to model points. Despite the method's inefficiency compared wiht the other methods proposed in [27] and [21], in the following experiments, our algorithms execute at rates of 2–3 s/frame of input data on a single processor of a Silicon Graphics 4D-340VGX workstation, including real-time 3-D graphics. The estimator advances to the next frame of data when the change in each of the estimated parameters falls below 10^{-4}. The Euler method time step was $4.0 \times 10^{-5} s$, and we used a unit damping matrix $\boldsymbol{D}$.

We first present several quantitative experiments involving synthetic data. We applied our Kalman estimation algorithm to 3-D data sampled over time from free-form deforming objects. In Fig. 6, we fit a deformable superquadric model with

[5] Note that the force assignment algorithm need not be executed at every time step of the fitting process. It can be executed every l iterations, where l is directly dependent on some measure of model deformation. Furthermore, note that if the assignment of datapoints to model points varies over time, only the normal component of surface motion may be estimated accurately, in principle. Hence, in applications where the tangential component of surface motion is significant (e.g., twisting deformations), the datapoint to model point attachment should remain invariant in time.

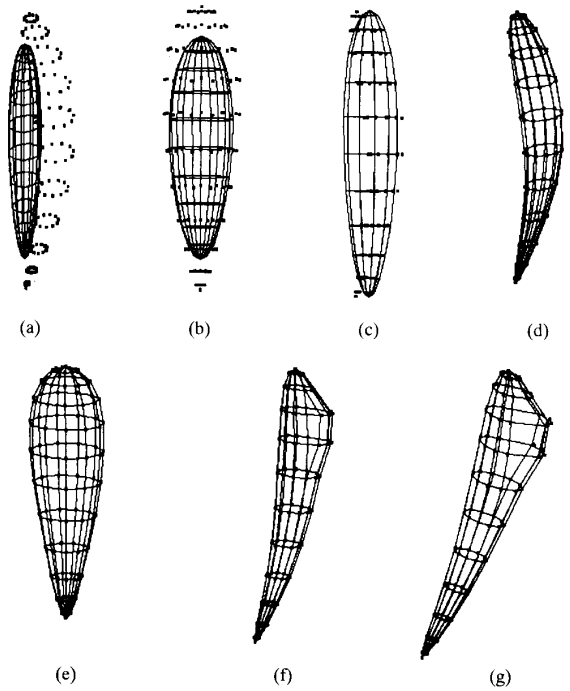


Fig. 6. Tracking of fully deformable object.

123 nodes to 123 3-D datapoints. The datapoints are sampled over 20 frames from the surface of a deformable superquadric object that undergoes both global and local deformations in response to a time-varying applied force. Fig. 6(a) and (b) shows two views of the data and the initial model. The local deformation stiffness parameters of the model were $w_0 = 0.1$ and $w_1 = 0.5$ (see [27]). The initial model was an ellipsoid ($\boldsymbol{q}_s = (2.7, 0.1, 0.2, 0.7, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.2)^T$). The diagonal entries of the initial covariance matrix $\boldsymbol{P}(0)$ were set to 1.0, whereas the off-diagonal entries used in the estimation of local deformations were set to 0.4. We also set $\boldsymbol{Q} = 0.1\boldsymbol{I}$ and $\boldsymbol{V} = 0.1\boldsymbol{I}$. In order to demonstrate the performance of the model-fitting algorithm, we did not initialize the part models at the center of gravity of the data. Fig. 6(c) shows an intermediate step of the fitting process driven by data forces from the first frame of the motion sequence, whereas Fig. 6(d) and (e) shows the model fitted to the initial data with visible tapering and bending global deformations. Fig. 6(f) shows an intermediate frame of the model tracking the nonrigid motion of the object, whereas Fig. 6(g) shows the final position of the model; the local deformations are readily apparent.

To assess the performance of our estimation algorithm quantitatively, we conducted a series of error analyses with data points from the above experiment at 100, 75, 50, and 25% data densities. We corrupted the data with zero-mean independent Gaussian noise of variance 0, 1, 5, and 10. We initialized the estimator as in the above experiment. In each run, we compute the average error per frame in the estimated translation, rotation, global, and local deformation, as well as the error in the estimated nodal positions of the model using

TABLE I
ERROR ANALYSIS. DATA DENSITY 100%

Variance	Translation error	Rotation error	Global error	Local error	Position error
10	0.06	0.02	0.19	0.65	0.71
5	0.04	0.005	0.18	0.43	0.44
1	0.039	0.003	0.117	0.1	0.09
0	0.025	0.002	0.011	0.03	0.014

TABLE II
ERROR ANALYSIS. DATA DENSITY 75%

Variance	Translation error	Rotation error	Global error	Local error	Position error
10	0.062	0.027	0.22	0.67	0.86
5	0.05	0.007	0.19	0.48	0.49
1	0.04	0.004	0.17	0.37	0.29
0	0.039	0.0034	0.15	0.27	0.24

TABLE III
ERROR ANALYSIS. DATA DENSITY 50%

Variance	Translation error	Rotation error	Global error	Local error	Position error
10	0.07	0.03	0.25	0.99	1.07
5	0.06	0.015	0.19	0.66	0.62
1	0.05	0.005	0.18	0.48	0.39
0	0.045	0.0036	0.19	0.43	0.32

TABLE IV
ERROR ANALYSIS. DATA DENSITY 25%

Variance	Translation error	Rotation error	Global error	Local error	Position error
10	0.092	0.036	0.29	1.03	1.24
5	0.09	0.023	0.25	0.7	0.95
1	0.089	0.016	0.2	0.6	0.64
0	0.068	0.013	0.17	0.57	0.53

the formula

$$\mathrm{err}_{\boldsymbol{a}} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(\boldsymbol{a}_{e_i} - \boldsymbol{a}_{T_i})^T(\boldsymbol{a}_{e_i} - \boldsymbol{a}_{T_i})} \tag{40}$$

where $\boldsymbol{a}$ stands for $\boldsymbol{q}_c$, $\boldsymbol{q}_\theta$, $\boldsymbol{q}_s$, $\boldsymbol{q}_d$, or $\boldsymbol{x}$, the subscripts e and T denote estimated and true values, respectively, and n is the number of frames used in the experiment—20 in this case.

Tables I–IV show the error analysis results for the various data percentages and noise variances.

Fig. 7 plots, over the 20 frames, the error in the estimated nodal positions from the 75% data density, noise variance 1 experiment. Given an inaccurate initial model, the error drops sharply as the algorithm rapidly estimates the correct nodal positions. The modest increase in error up to frame 8 and its subsequent decrease is due to the increasing size of the local deformation implied by the data. The filter tunes after eight frames (local deformations are now significant), and we observe a subsequent decrease in the frame error.

The error analysis experiments demonstrate that the recovery of translation, rotation, and global deformations is more robust than the recovery of local deformations in the presence of noise. The Kalman filter is insensitive to some of the noise because it does not force the local deformations to interpolate the datapoints. It produces reasonably accurate shape estimates with small overall position estimation error. We observed that the reliability of the estimated motion parameters began to fail at data densities of around 25% and noise variances around 10.

The following experiment indicates the performance of the method for nonGaussian noise[6] without one-to-one correspondence between model nodes and data points. In Fig.

[6]The estimation of accurate noise models from input data is beyond the scope of this paper.

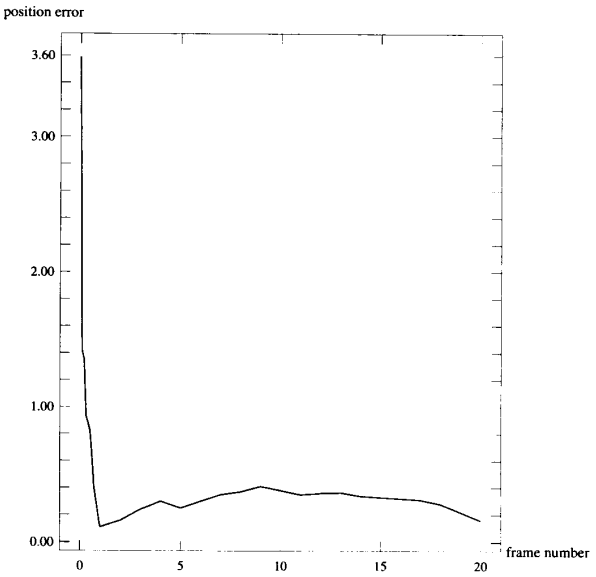


Fig. 7. Position error per frame with 75% data density and noise variance 1.0.

8, we add uniform ±5% perturbations, with randomly chosen sign, to the positions of the 123 data points. We fit a deformable superquadric with 81 nodes whose parameters and initial position are the same as in the first experiment, except for the initial ellipsoid parameters that were $\boldsymbol{q}_s = (2.8, 0.1, 0.2, 0.8, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.2)^T$, and we set $\boldsymbol{V} = 6.0\boldsymbol{I}$ to account for noise.

Fig. 8 is similar to Fig. 6. Fig. 8(a) and (b) shows two views of the data and the initial model. Fig. 8(c) shows an intermediate step of the fitting process driven by data forces from the first frame of the motion sequence. Fig. 8(d) and (e) shows the model fitted to the initial data with visible tapering and bending global deformations. Fig. 8(f) shows an intermediate frame of the model tracking the nonrigid motion of the object, whereas Fig. 8(g) and (h) shows the final position of the model.

We used our dynamic tracking algorithm in the following two experiments involving constrained multipart models. Fig. 9 illustrates the tracking of an articulated "insect" consisting of five deformable superquadric parts, each having 27 nodes and four point-to-point constraints. We again synthesized datapoints by simulating the motions of constrained deformable superquadrics undergoing global and local deformations. We sampled 27 datapoints from each superquadric through time (36 frames). The nonrigid motions were imparted by three forces: a force applied to each wing tip to make the wings flap and a force applied to the nose to pull the "insect" forward through space. The local deformation stiffness parameters were set to $w_0 = 0.1$ and $w_1 = 0.5$. To demonstrate the performance of our constrained motion algorithm, we initialize the deformable superquadrics models to ellipsoidal shapes ($\boldsymbol{q}_s = (2.0, 0.8, 0.07, 0.07, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.5)^T$) as shown in Fig. 9(a). Fig. 9(b) shows an intermediate step in fitting the models to the datapoints associated with the first

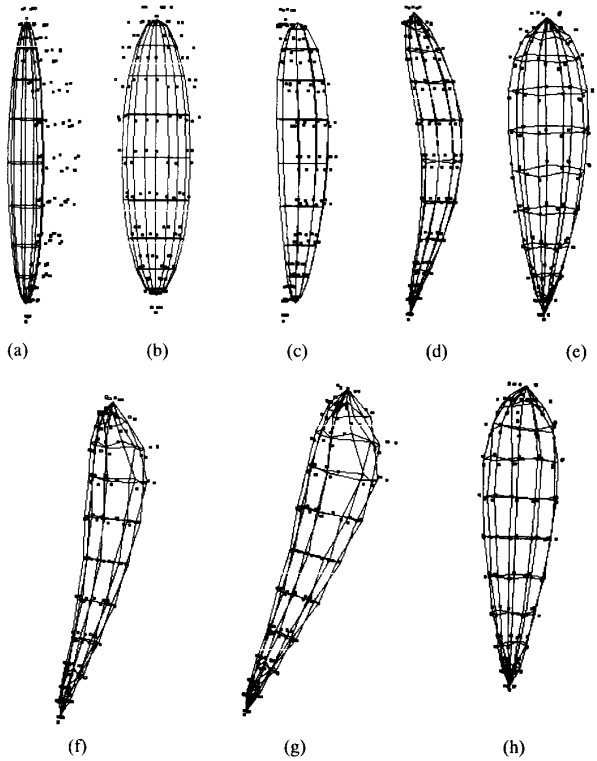


Fig. 8. Tracking of fully deformable object with noise.

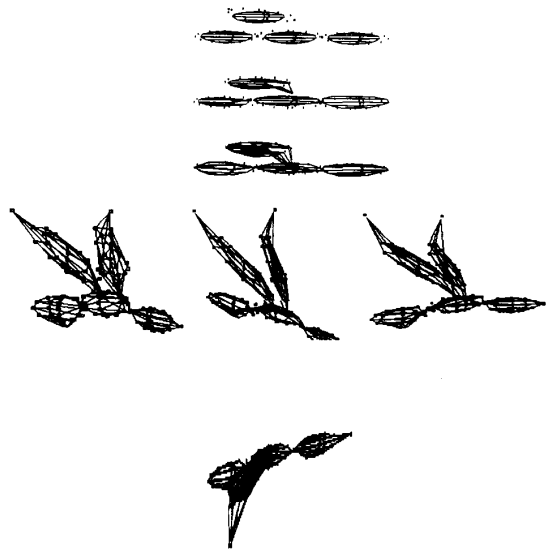


Fig. 9. Tracking of an "insect's" parts.

time frame, whereas Fig. 9(c) shows the final fit to these initial datapoints. The constraint forces help configure the five parts correctly, as is evident from the final deformations of the wings. Fig. 9(d) shows three time frames of the fitted "insect" model tracking the time-varying datapoints with the wings open upwards, whereas Fig. 9(e) shows a frame of the motion with the wings dipping downward.

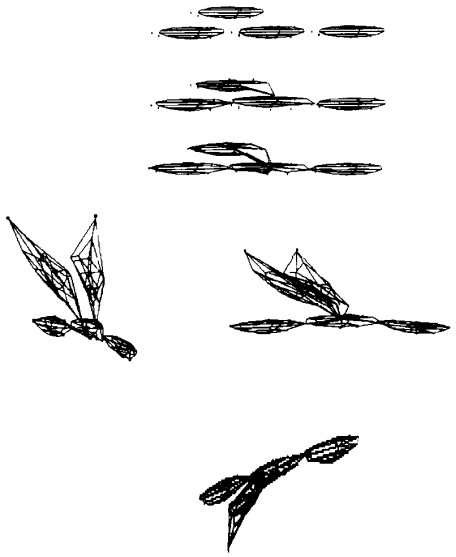


Fig. 10. Tracking of an "insect's" parts using sparse data.

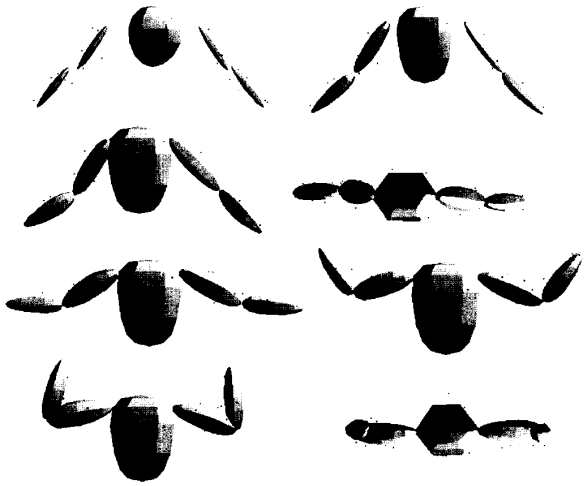


Fig. 11. Tracking of raising and flexing human arm motion.

Fig. 10 illustrates a sparse datapoint version of the previous experiment. We use the same parameter settings, but now, there are only seven datapoints per superquadric. Again, the constraint forces help link the five superquadrics into the correct configuration. The shapes of the wings in Fig. 10(d) differ from the shapes in the previous "insect" experiment. The sparse data do not provide enough constraints for the local deformations to recover the exact shapes.

We conducted an experiment involving real 3-D data acquired using the WATSMART system from arm and torso motions of a human subject. Fig. 11 shows shape and motion estimation results involving an articulated multipart model composed of five deformable superquadrics connected by four point-to-point constraints. We used 49-node deformable superquadric models for the arms and a 36-node model for the torso. The input data for the estimation were collected from the raising and flexing motion of the two arms of a human subject. Approximately 120 data frames were used in this experiment. The incoming data are "segmented," inasmuch as individual datapoints are associated with the correct body parts. Fig. 11(a) shows a frontal view of the data and the initial models. The models were automatically initialized to the center of mass of their associated data, and their initial orientations were computed using the matrix of central moments of the data. The initial size along the longest axis was set by simply calculating the distance between the furthermost data points along the initial model-centered coordinate system [24] (we initialized the model along the other two axes arbitrarily). The initial Kalman filter covariance matrix was $\boldsymbol{P}(0) = \boldsymbol{I}$, and we used $\boldsymbol{Q} = 0.1\boldsymbol{I}$ and $\boldsymbol{V} = 4.0\boldsymbol{I}$. Fig. 11(b) shows an intermediate step of the fitting process driven by stabilized constraint forces (with $a = b = 6$) and data forces from the first frame of the motion sequence. Fig. 11(c) and (d) shows frontal and top views of the models fitted to the initial data. Fig. 11(e) and (f) shows intermediate frames of the models tracking the nonrigid motion of the arms, whereas Fig. 11(g) and (h) shows frontal and top views of the final position of the models.

The WATSMART experiment demonstrates that incorporating our constraint algorithm into the Kalman filter produces satisfactory fits to the sparse and noise corrupted data. Local deformations were not permitted in the experiment since the available data were very sparse, i.e., about 6 to 13 points per deformable superquadric part.

The above experiments indicate the tradeoffs between local and global deformations when our models must deal with sparse data. Global deformations require relatively few datapoints to abstract the rough shapes of objects. By contrast, local deformations can provide a more accurate approximation to the exact shape of a complex object, but their recovery generally requires more data. The symbiosis of local and global deformations within our dynamic models appears to offer the best of both worlds.

VII. Conclusion

This paper presented a physics-based framework for 3-D shape modeling and nonrigid motion estimation from incomplete, noise-corrupted, time-varying observations. Our approach applies Lagrangian mechanics to systematically convert parameterized geometric parts, deformations, and constraints into dynamic models. Assimilating these dynamic models within continuous nonlinear Kalman filtering theory, we derived recursive algorithms to estimate translation, rotation, and deformation parameters of nonrigid objects. Our estimators are much more sophisticated and powerful than previous Kalman filter-based estimators described in the vision literature (including our earlier dynamic estimators that may be viewed as degenerate "Kalman filters" with unit error covariance matrices). The formulations in this paper were purposefully general since the general models show promise for present off-line and future real-time analysis tasks. Nonetheless, we showed that it is also possible to significantly ease the com-

putational burden of our recursive estimation algorithms with minimal loss of accuracy through state decoupling. We are confident that with this and other reasonable simplifications, our framework can support fast, on-line estimation of nonrigid motion parameters on currently available computers for real-time vision applications.

APPENDIX
JACOBIAN MATRIX

Defining $r = \frac{e_3+b_2}{a_0 w a_3}\pi b_3$, the Jacobian matrix $\boldsymbol{J}$ is a 3×11 matrix whose nonzero entries are

$$
\begin{aligned}
\boldsymbol{J}_{11} &= (t_1 {S_u}^{\epsilon_1} + 1) w a_1 {C_u}^{\epsilon_1} {C_v}^{\epsilon_2} + \frac{b_1 b_2 b_3}{a_0^2 w a_3}\pi \sin(r) \\
\boldsymbol{J}_{21} &= (t_2 {S_u}^{\epsilon_1} + 1) w a_2 {C_u}^{\epsilon_1} {S_v}^{\epsilon_2} \\
\boldsymbol{J}_{31} &= w a_3 {S_u}^{\epsilon_1} \\
\boldsymbol{J}_{12} &= (t_1 {S_u}^{\epsilon_1} + 1) a_0 w {C_u}^{\epsilon_1} {C_v}^{\epsilon_2} \\
\boldsymbol{J}_{23} &= (t_2 {S_u}^{\epsilon_1} + 1) a_0 w {C_u}^{\epsilon_1} {S_v}^{\epsilon_2} \\
\boldsymbol{J}_{14} &= \frac{b_1 b_2 b_3}{a_0 w {a_3}^2}\pi \sin(r) \\
\boldsymbol{J}_{34} &= a_0 w {S_u}^{\epsilon_1} \\
\boldsymbol{J}_{15} &= t_1 \ln(|\sin u|) {S_u}^{\epsilon_1} a_0 w a_1 {C_u}^{\epsilon_1} {C_v}^{\epsilon_2} + \\
&\quad (t_1 {S_u}^{\epsilon_1} + 1) a_0 w a_1 \ln(|\cos u|) {C_u}^{\epsilon_1} {C_v}^{\epsilon_2} \\
&\quad - b_1 b_3 \pi \ln(|\sin u|) {S_u}^{\epsilon_1} \sin(r) \\
\boldsymbol{J}_{25} &= t_2 \ln(|\sin u|) {S_u}^{\epsilon_1} a_0 w a_2 {C_u}^{\epsilon_1} {S_v}^{\epsilon_2} + \\
&\quad (t_2 {S_u}^{\epsilon_1} + 1) a_0 w a_2 \ln(|\cos u|) {C_u}^{\epsilon_1} {S_v}^{\epsilon_2} \\
\boldsymbol{J}_{35} &= a_0 w a_3 \ln(|\sin u|) {S_u}^{\epsilon_1} \\
\boldsymbol{J}_{16} &= (t_1 {S_u}^{\epsilon_1} + 1) a_0 w a_1 \ln(|\cos v|) {C_u}^{\epsilon_1} {C_v}^{\epsilon_2} \\
\boldsymbol{J}_{26} &= (t_2 {S_u}^{\epsilon_1} + 1) a_0 w a_2 \ln(|\sin v|) {C_u}^{\epsilon_1} {S_v}^{\epsilon_2} \\
\boldsymbol{J}_{17} &= {S_u}^{\epsilon_1} a_0 w a_1 {C_u}^{\epsilon_1} {C_v}^{\epsilon_2} \\
\boldsymbol{J}_{28} &= {S_u}^{\epsilon_1} a_0 w a_2 {C_u}^{\epsilon_1} {S_v}^{\epsilon_2} \\
\boldsymbol{J}_{19} &= \cos(r) \\
\boldsymbol{J}_{1\,10} &= -\frac{b_1 b_3}{a_0 w a_3}\pi \sin(r) \\
\boldsymbol{J}_{1\,11} &= -b_1 \pi \sin(r) r,
\end{aligned} \tag{41}
$$

where ${S_\theta}^{\epsilon} = \operatorname{sgn}(\sin\theta)|\sin\theta|^{\epsilon}$ and ${C_\theta}^{\epsilon} = \operatorname{sgn}(\cos\theta)|\cos\theta|^{\epsilon}$.

CALCULATION OF ACCELERATION AND INERTIAL FORCES

The acceleration of a point $\boldsymbol{x}$ on the deformable model is given by [23]

$$\ddot{\boldsymbol{x}} = \boldsymbol{L}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{L}}\dot{\boldsymbol{q}} \tag{42}$$

where

$$\dot{\boldsymbol{L}}\dot{\boldsymbol{q}} = \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{R}\boldsymbol{p}) + 2\boldsymbol{\omega} \times \boldsymbol{R}\dot{\boldsymbol{p}}. \tag{43}$$

Here, $\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{R}\boldsymbol{p})$ and $2\boldsymbol{\omega} \times \boldsymbol{R}\dot{\boldsymbol{p}}$ are the centrifugal and the Coriolis accelerations, respectively. We obtain

$$\dot{\boldsymbol{p}} = \dot{\boldsymbol{s}} + \dot{\boldsymbol{d}} = \boldsymbol{J}\dot{\boldsymbol{q}}_s + \boldsymbol{S}\dot{\boldsymbol{q}}_d \tag{44}$$

and the angular velocity of the deformable model with respect to the world coordinate system

$$\boldsymbol{\omega} = \boldsymbol{Q}\dot{\boldsymbol{\theta}} \tag{45}$$

where $\boldsymbol{Q}$ is a 3×4 matrix whose definition is based on the value of the quaternion $\boldsymbol{\theta} = \boldsymbol{q}_\theta = [s, (v_1, v_2, v_3)^T]$ representing the rotation at time t:

$$\boldsymbol{Q} = 2\begin{bmatrix} -v_1 & s & -v_3 & v_2 \\ -v_2 & v_3 & s & -v_1 \\ -v_3 & -v_2 & v_1 & s \end{bmatrix}. \tag{46}$$

The virtual work due to inertia on the deformable model is computed as follows [23]:

$$
\begin{aligned}
\delta \boldsymbol{W}_I &= \int \mu \delta \boldsymbol{x}^T \ddot{\boldsymbol{x}}\, \mathrm{du} = \int \delta \boldsymbol{q}^T \boldsymbol{L}^T (\boldsymbol{L}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{L}}\dot{\boldsymbol{q}})\, \mathrm{du} \\
&= \delta \boldsymbol{q}^T (\boldsymbol{M}\ddot{\boldsymbol{q}} - \boldsymbol{g}_q)
\end{aligned} \tag{47}
$$

where the generalized mass matrix is

$$\boldsymbol{M} = \int \mu \boldsymbol{L}^T \boldsymbol{L}\, \mathrm{du} \tag{48}$$

and the generalized inertial forces are

$$\boldsymbol{g}_q = -\int \mu \boldsymbol{L}^T \dot{\boldsymbol{L}}\dot{\boldsymbol{q}}\, \mathrm{du}. \tag{49}$$

ACKNOWLEDGMENT

We would like to thank R. Szeliski, W. Enright, and R. Kwong for discussions on Kalman filtering and numerical analysis, A. Shabana for discussions on dynamics, and S. Tupling for providing access to the WATSMART system.

REFERENCES

[1] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Comput. Methods Applied Mechanics Eng.*, vol. 1, pp. 1–16, 1972.
[2] A. Blake and A. Yuille, Eds., *Active Vision*. Cambridge, MA: MIT Press, 1992.
[3] A. Barr, "Global and local deformations of solid primitives," *Comput. Graphics*, vol. 18, pp. 21–30, 1984.
[4] R. Barzel and A. Barr, "A modeling system based on dynamic constraints," *Comput. Graphics*, vol. 22, no. 4, pp. 179–188, 1988.
[5] T. Broida and R. Chellappa, "Estimation of object motion parameters from noisy images," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 8, no. 1, pp 90–98, Jan. 1986.
[6] T. J. Broida, S. Chandrashekhar, and R. Chellappa, "Recursive 3-D motion estimation from a monocular image sequence," *IEEE Trans. Aerospace Electron. Syst.*, vol. 26, no. 4, pp. 639–656, July 1990.
[7] C. W. Chen and T. S. Huang, "Epicardial motion and deformation estimation from coronary artery bifurcation points," in *Proc. IEEE Third Int. Conf. Comput. Vision (ICCV'90)* (Osaka), Dec. 1990, pp. 456–459.
[8] R. Deriche and O. Faugeras, "Tracking line segments," *Image Vision Comput.*, vol. 8, no. 4, pp. 261–270, Nov. 1990.
[9] E. D. Dickmanns and V. Graefe, "Dynamic monocular machine vision," *Machine Vision Applications*, vol. 1, pp. 223–240, 1988.
[10] ——, "Applications of dynamic monocular machine vision," *Machine Vision Applications*, vol. 1, pp. 241–261, 1988.
[11] J. S. Duncan, R. L. Owen, and P. Anandan, "Measurement of nonrigid motion using contour shape descriptors," *IEEE Comput. Vision Patt. Recogn. Conf. (CVPR'91)* (Hawaii), 1991, pp. 318–324.
[12] A. Gelb, *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1974.
[13] D. B. Goldgof, H. Lee, and T. S. Huang, "Motion analysis of nonrigid structures," in *Proc. IEEE Computer Vision and Pattern Recognition Conference (CVPR'88)*, 1988, pp. 375–380.
[14] J. Heel, "Temporally integrated surface reconstruction," in *Proc. IEEE Third Int. Conf. Comput. Vision (ICCV'90)* (Osaka), Dec. 1990, pp. 292–295.
[15] T. S. Huang, "Modeling, analysis and visualization of nonrigid object motion," in *Proc. IEEE 10th Int. Conf. Pattern Recogn.* (Atlantic City, NJ), 1990, pp. 361–364.

[16] H. Kardestuncer, *Finite Element Handbook*. New York: McGraw-Hill, 1987.
[17] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vision*, vol. 1, no. 4, pp. 321–331, 1988.
[18] L. Matthies, T. Kanade, and R. Szeliski, "Kalman filter-based algorithms for estimating depth from image sequences," *Int. J. Comput. Vision*, vol. 3, pp. 209–236, 1989.
[19] D. Metaxas and D. Terzopoulos, "Constrained deformable superquadrics and nonrigid motion tracking," in *Proc. IEEE Comput. Vision Patt. Recogn. Conf. (CVPR'91)* (Hawaii), June 1991, pp. 337–343.
[20] ——, "Dynamic deformation of solid primitives with constraints," in *Proc. Comput. Graphics ACM Siggraph'92* (Chicago, IL), July 1992, pp. 309–312, vol. 26, no. 2.
[21] D. Metaxas, "Physics-based modeling of nonrigid objects for vision and graphics," Ph.D thesis, Dept. Comput. Sci., Univ. of Toronto, Toronto, Canada, 1992.
[22] A. Pentland and B. Horowitz, "Recovery of nonrigid motion and structure," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 13, no. 7, pp. 730–742, 1991.
[23] A. A. Shabana, *Dynamics of Multibody Systems*. New York: Wiley, 1989.
[24] F. Solina and R. Bajcsy, "Recovery of parametric models from range images: The case for superquadrics with global deformations," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, no. 2, pp. 131–146, 1990.
[25] R. Szeliski and D. Terzopoulos, "Physically-based and probabilistic modeling for computer vision," in *Proc. SPIE vol. 1570 Geometric Methods Comput. Vision* (San Diego), July 1991, pp. 140–152.
[26] D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on deformable models: Recovering 3D shape and nonrigid motion," *Artificial Intell.*, vol. 36, no. 1, pp. 91–123, 1988.
[27] D. Terzopoulos and D. Metaxas, "Dynamic 3D models with local and global deformations: Deformable superquadrics," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 13, no. 7, pp. 703–714, 1991; see also *Proc. Third Int. Conf. Comput. Vision (ICCV'90)* (Osaka), Dec. 1990, pp. 606-615.
[28] Y. F. Wang and J. -F. Wang, "Surface reconstruction using deformable models with interior and boundary constraints," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 14, no. 5, pp. 572–579, May 1992.
[29] J. Wittenberg, *Dynamics of Systems of Rigid Bodies*. Stuttgart: Tubner, 1977.

Dimitri Metaxas (S'91–M'93) received the Diploma in electrical engineering with highest honours from the National Technical University of Athens, Athens, Greece, in 1986, the M.Sc. degree in computer science from the University of Maryland, College Park, in 1988, and the Ph.D. degree in computer science from the University of Toronto, Canada, in 1992.

Since September 1992, he has been an Assistant Professor in the Department of Computer and Information Science at the University of Pennsylvania, Philadelphia. During the summer of 1985, he did research in artificial intelligence at NEC's research laboratories in Tokyo, Japan, and from June 1987 to June 1988, he worked as a research assistant at the University of Maryland Center for Automation Research.

Dr. Metaxas has published papers in computer vision, computer graphics, computer systems, databases, and artificial intelligence. He has received several scholarships including a Fulbright scholarship and a University of Toronto Open Doctoral Fellowship. He is a member of ACM and the Technical Chamber of Greece.

Demetri Terzopoulos (M'78), For a photograph and biography, see this issue, p. 579.