Contents

Deformable Models: Classic, Topology-Adaptive and Generalized Formulations

Demetri Terzopoulos

ABSTRACT
"Deformable models" refers to a class of physics-based modeling methods with an extensive track record in computer vision, medical imaging, computer graphics, geometric design, and related areas. Unlike the Eulerian (fluid) formulations associated with level set methods, deformable models are characterized by Lagrangian (solid) formulations, three variants of which are reviewed herein.

1 Introduction

This chapter reviews *Deformable Models*, a powerful class of physics-based modeling techniques widely employed in image synthesis (computer graphics), image analysis (computer vision), shape design (computer-aided geometric design) and related fields. It may at first seem odd to find a chapter on deformable models, whose governing equations are based on Lagrangian continuum mechanics formulations, in a volume dedicated to the Eulerian formulations associated with level set methods. Upon reflection, however, it becomes self-evident that the two approaches are complementary in precisely the same sense that Lagrangian solid models complement Eulerian fluid models in continuum mechanics. The substantial literature on deformable models and level set methods is a testament to the fact that both approaches are useful in imaging, vision, and graphics.

By numerically simulating their governing equations of motion, typically expressed as PDEs in a continuous setting or ODEs in a discrete setting, deformable models mimic various generic behaviors of natural nonrigid materials in response to applied forces, such as continuity, smoothness, elasticity, plasticity, etc. For the purposes of computer graphics, realistic images and animations of elastic, inelastic, and thermoelastic objects may be synthesized when the applied forces stem from animation controllers and from model interactions within simulated physical environments [17] (see, e.g., Fig. 1). In the related domain of computer-aided geometric design, deformable models have inspired a new approach known as "physics-based geometric design". Here, the parameters of standard geometric primitives become generalized coordinates in Lagrangian formulations that govern their automatic evolution in response to simulated (sculpting) forces, subject to geometric constraints [21].

Among model-based techniques for image analysis, deformable models offer a potent approach that combines geometry, physics, and approximation theory. In

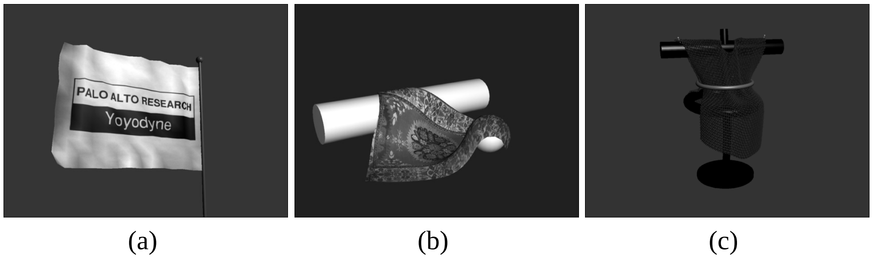

FIGURE 1. Early computer graphics images of physics-based deformable surface model simulations depicting objects with cloth-like behaviors [20, 17]. (a) Flag in wind. (b) Rug falling over geometric obstacles. (c) Draped robe and constraints.

applications to computer vision, deformable models may be used to infer image disparity fields, image flow fields, and to infer the shapes and motions of objects from still or video images [23, 16]. In this context, deformable models are subjected to external forces that impose constraints derived from image data. The forces actively shape and move models to achieve maximal consistency with imaged objects of interest and to maintain consistency over time. There has been a tremendous surge of interest in deformable models in the context of medical image analysis [8]. Here, deformable models have proven useful in segmenting, matching, and tracking anatomic structures by exploiting (bottom-up) constraints derived from the image data in conjunction with (top-down) *a priori* knowledge about the location, size, and shape of these structures. With regard to the latter, deformable models support intuitive interaction mechanisms that enable medical scientists and practitioners to bring their expertise to bear on the model-based image interpretation task.

This remainder of this chapter comprises three main sections. Section 2 covers the mathematical foundations of “classic” deformable models, including energy-minimizing and dynamic *snakes* [4], their discretization, numerical simulation and probabilistic interpretation. The parametric geometry of snakes allows controlled, piecewise continuity and both closed and open curves (note that current level set methods have difficulty representing open curves). The section also covers higher dimensional generalizations of snakes, such as *deformable surfaces*. In image segmentation, the introduction of level set methods was motivated by the success of deformable models and the need for a related technique that makes no prior assumption about the topology of the underlying structure of interest. As other chapters in this volume make evident, Eulerian models of fluids, particularly those implemented via level set methods, seem more appropriate in the face of unknown topology. In Section 3, however, I review *topology-adaptive deformable models*, which provide the topological flexibility of level set methods without sacrificing the explicit geometric substrate upon which the classic deformable model formulations are based. In Section 4, I review deformable models constructed on more sophisticated geometric substrates, including *dynamic non-uniform rational B-splines* for physics-based geometric design and *deformable superquadrics* for computer vision and computer graphics animation. Thus far, such generalized deformable models, in which arbitrary geometric parameters play the role of Lagrangian generalized coordinates, seem to fall outside the scope of level set methods. Finally, Section 5 concludes the chapter.[1]

2 Classic Deformable Models

This section reviews the mathematics of classic deformable models. First, I consider planar *active contour models*, also known as “snakes”, including energy-minimizing snakes and dynamic snakes. Next, I discuss the discretization and numerical simulation of snakes, as well as their probabilistic interpretation. Finally, I review higher-dimensional generalizations of snakes, in particular, deformable surfaces.

2.1 Energy-Minimizing Snakes

Snakes are planar deformable contours that are useful in a variety of image analysis tasks [4]. They are often used to approximate the locations and shapes of object boundaries in images, based on the assumption that boundaries are piecewise continuous or smooth (Fig. 2(a)). In its basic form, the mathematical formulation of snakes draws from the theory of optimal approximation involving functionals.

Geometrically, a snake is an explicit, parametric contour embedded in the image plane $(x, y) \in \Re^2$. The contour is represented as $\mathbf{v}(s) = (x(s), y(s))^\top$, where x and y are the coordinate functions and $s \in [0, 1]$ is the parametric domain (the symbol $^\top$ denotes transposition). The shape of the contour subject to an image $I(x, y)$ is dictated by the functional

$$\mathcal{E}(\mathbf{v}) = \mathcal{S}(\mathbf{v}) + \mathcal{P}(\mathbf{v}), \tag{1.1}$$

which represents the energy of the contour. The final shape of the contour corresponds to the minimum of this energy.

The first term in (1.1),

$$\mathcal{S}(\mathbf{v}) = \frac{1}{2}\int_0^1 w_1(s)\left|\frac{\partial \mathbf{v}}{\partial s}\right|^2 + w_2(s)\left|\frac{\partial^2 \mathbf{v}}{\partial s^2}\right|^2 ds, \tag{1.2}$$

is the internal deformation energy. It characterizes the deformation of a stretchy, flexible contour. Two physical parameter functions, the non-negative functions

[1] This chapter makes no pretense to being a literature survey. In fact, I shall continue herein to cite only my own published work on deformable models. The interested reader may refer to the more recent of these sources for a broader perspective on the extensive deformable models literature, as well as to this volume’s other chapters and integrated bibliography for the associated literature on level set methods.

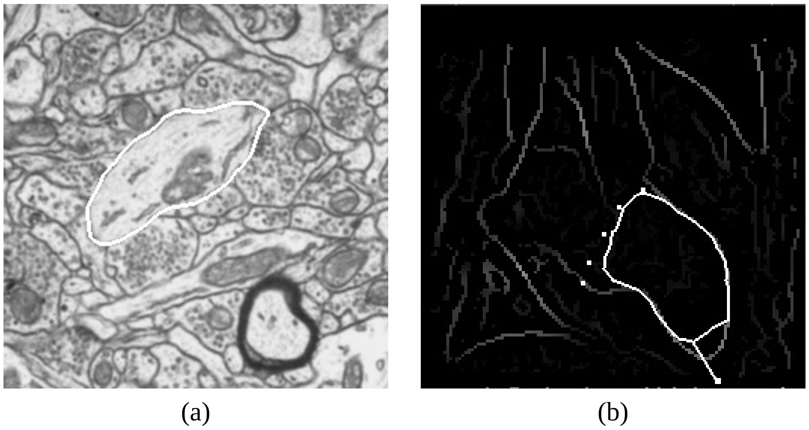

FIGURE 2. Medical image analysis with snakes. (a) Snake (white) segmenting a cell in an EM photomicrograph [2]. The snake is attracted to the dark cell membrane. (b) Snake deforming towards high gradients in a processed cardiac image, influenced by "pin" constraints and an interactive "spring" with which the user pulls the contour towards an edge [6].

$w_1(s)$ and $w_2(s)$, dictate the simulated physical characteristics of the contour at any point s on the snake: $w_1(s)$ controls the "tension" of the contour while $w_2(s)$ controls its "rigidity". For example, increasing the magnitude of $w_1(s)$ tends to eliminate extraneous loops and ripples by reducing the length of the snake. Increasing $w_2(s)$ makes the snake smoother and less flexible. Setting the value of one or both of these functions to zero at a point s permits discontinuities in the contour at s.

The second term in (1.1) couples the snake to the image. Traditionally,

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

where $P(x, y)$ denotes a scalar potential function defined on the image plane. To apply snakes to images, external potentials are designed whose local minima coincide with intensity extrema, edges, and other image features of interest. For example, the contour will be attracted to intensity edges in an image $I(x, y)$ by choosing a potential $P(x, y) = -c|\nabla[G_\sigma * I(x, y)]|$, where c controls the magnitude of the potential, ∇ is the gradient operator, and $G_\sigma * I$ denotes the image convolved with a (Gaussian) smoothing filter whose characteristic width σ controls the spatial extent of the local minima of P.

In accordance with the calculus of variations, the contour $\mathbf{v}(s)$ which minimizes the energy $\mathcal{E}(\mathbf{v})$ must satisfy the Euler-Lagrange equation

$$-\frac{\partial}{\partial s}\left(w_1 \frac{\partial \mathbf{v}}{\partial s}\right) + \frac{\partial^2}{\partial s^2}\left(w_2 \frac{\partial^2 \mathbf{v}}{\partial s^2}\right) + \nabla P(\mathbf{v}(s, t)) = \mathbf{0}. \tag{1.4}$$

This vector-valued partial differential equation (PDE) expresses the balance of internal and external forces when the contour rests at equilibrium. The first two terms represent the internal stretching and bending forces, respectively, while the third term represents the external forces that couple the snake to the image data. The usual approach to solving (1.4) is through the application of numerical algorithms (see Sec. 2.3).

2.2 Dynamic Snakes

While it is natural to view energy minimization as a static problem, a potent approach to computing the local minima of a functional such as (1.1) is to construct a dynamical system that is governed by the functional and allow the system to evolve to equilibrium. The system may be constructed by applying the principles of Lagrangian mechanics. This leads to dynamic deformable models that unify the description of shape and motion, making it possible to quantify not just static shape, but also shape evolution through time. Dynamic models are valuable for, e.g., time-varying medical image analysis, since most anatomical structures are deformable and continually undergo nonrigid motion *in vivo*. Moreover, dynamic models exhibit intuitively meaningful physical behaviors, making their evolution amenable to interactive guidance from a user (Fig. 2(b)).

A simple example is a dynamic snake which can be represented by introducing a time-varying contour $\mathbf{v}(s,t) = (x(s,t), y(s,t))^\top$ along with a mass density $\mu(s)$ and a damping density $\gamma(s)$. The Lagrange equations of motion for a snake with the internal energy given by (1.2) and external energy given by (1.3) is

$$\mu\frac{\partial^2\mathbf{v}}{\partial t^2} + \gamma\frac{\partial\mathbf{v}}{\partial t} - \frac{\partial}{\partial s}\left(w_1\frac{\partial\mathbf{v}}{\partial s}\right) + \frac{\partial^2}{\partial s^2}\left(w_2\frac{\partial^2\mathbf{v}}{\partial s^2}\right) = -\nabla P(\mathbf{v}(s,t)). \tag{1.5}$$

The first two terms on the left hand side of this partial differential equation represent inertial and damping forces. As in (1.4), the remaining terms represent the internal stretching and bending forces, while the right hand side represents the external forces. Equilibrium is achieved when the internal and external forces balance and the contour comes to rest (i.e., $\partial\mathbf{v}/\partial t = \partial^2\mathbf{v}/\partial t^2 = 0$), which yields the equilibrium condition (1.4).

2.3 Discretization and Numerical Simulation

In order to compute numerically a minimum energy solution, it is necessary to discretize the energy $\mathcal{E}(\mathbf{v})$. The usual approach is to represent the continuous geometric model $\mathbf{v}$ in terms of linear combinations of local-support or global-support basis functions. Finite elements, finite differences, and geometric splines are local representation methods, whereas Fourier bases are global representation methods. The continuous model $\mathbf{v}(s)$ is represented in discrete form by a vector $\mathbf{u}$ of shape parameters associated with the basis functions. The discrete form of energies such as $\mathcal{E}(\mathbf{v})$ for the snake may be written as

$$E(\mathbf{u}) = \frac{1}{2}\mathbf{u}^\top\mathbf{K}\mathbf{u} + \mathrm{P}(\mathbf{u}) \tag{1.6}$$

where $\mathbf{K}$ is called the *stiffness matrix*, and $\mathrm{P}(\mathbf{u})$ is the discrete version of the external potential. The minimum energy solution results from setting the gradient of (1.6) to $\mathbf{0}$, which is equivalent to solving the set of algebraic equations

$$\mathbf{K}\mathbf{u} = -\nabla \mathrm{P} = \mathbf{f} \tag{1.7}$$

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

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

$$\mathbf{K} = \begin{bmatrix} a_0 & b_0 & c_0 & & & & c_{N-2} & b_{N-1} \\ b_0 & a_1 & b_1 & c_1 & & & & c_{N-1} \\ c_0 & b_1 & a_2 & b_2 & c_2 & & & \\ & c_1 & b_2 & a_3 & b_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & c_{N-5} & b_{N-4} & a_{N-3} & b_{N-3} & c_{N-3} \\ c_{N-2} & & & & c_{N-4} & b_{N-3} & a_{N-2} & b_{N-2} \\ b_{N-1} & c_{N-1} & & & & c_{N-3} & b_{N-2} & a_{N-1} \end{bmatrix}, \tag{1.8}$$

where

$$a_i = (w_{1\,i-1} + w_{1\,i})/h^2 + (w_{2\,i-1} + 4w_{2\,i} + w_{2\,i+1})/h^4, \tag{1.9}$$

$$b_i = -w_{1\,i}/h^2 - 2(w_{2\,i} + w_{2\,i+1})/h^4, \tag{1.10}$$

$$c_i = w_{2\,i+1}/h^4, \tag{1.11}$$

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

The discretized version of the Lagrangian dynamics equation (1.5) may be written as a set of second order ordinary differential equations (ODEs) for $\mathbf{u}(t)$:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}, \tag{1.12}$$

where $\mathbf{M}$ is the mass matrix and $\mathbf{D}$ is a damping matrix. In a finite difference discretization, the mass and damping matrices are diagonal matrices.

To simulate the snake dynamics, the system of ordinary differential equations (1.12) in the shape parameters $\mathbf{u}$ must be integrated forward through time. The finite element literature offers several suitable explicit and implicit direct integration methods, including the central difference, Houbolt, Newmark, or Wilson methods [1]. We can illustrate the basic idea with a semi-implicit Euler method that takes time steps Δt. We replace the time derivatives of u with the backward finite differences $\ddot{\mathrm{u}} \approx (\mathrm{u}^{(t+\Delta t)} - 2\mathrm{u}^{(t)} + \mathrm{u}^{(t-\Delta t)})/(\Delta t)^2$, and $\dot{\mathrm{u}} \approx (\mathrm{u}^{(t+\Delta t)} - \mathrm{u}^{(t-\Delta t)})/2\Delta t$, where the superscripts denote the quantity evaluated at the time given in parentheses. This yields the update formula

$$\mathbf{A}\mathrm{u}^{(t+\Delta t)} = \mathbf{b}^{(t)}, \tag{1.13}$$

where $\mathbf{A} = \mathbf{M}/(\Delta t)^2 + \mathbf{D}/2\Delta t + \mathbf{K}$ is a pentadiagonal matrix and $\mathbf{b}^{(t)} = (2\mathbf{M}/(\Delta t)^2)\mathrm{u}^{(t)} - (\mathbf{M}/(\Delta t)^2 - \mathbf{D}/2\Delta t)\mathrm{u}^{(t-1)} + \mathbf{f}^{(t)}$. The pentadiagonal system can be solved efficiently ($O(N)$ complexity) by factorizing $\mathbf{A}$ into lower and upper triangular matrices, then solving the two resulting sparse triangular systems. We compute the unique normalized factorization $\mathbf{A} = \mathbf{LYU}$ where $\mathbf{L}$ is a lower triangular matrix, $\mathbf{Y}$ is a diagonal matrix, and $\mathbf{U} = \mathbf{L}^\top$ is an upper triangular matrix [1]. The solution $\mathrm{u}^{(t+\Delta t)}$ to (1.13) is obtained by first solving $\mathbf{Ls} = \mathbf{b}^{(t)}$ by forward substitution, then $\mathbf{Uu} = \mathbf{Y}^{-1}\mathbf{s}$ by backward substitution. For the linear snakes described above, only a single factorization is necessary, since $\mathbf{A}$ is constant. Note that the factorization and forward/backward substitutions are inherently sequential, recursive operations.

Researchers have investigated alternative approaches to numerically simulating snake models, including dynamic programming and greedy algorithms (see [8] for a survey in the context of medical image analysis).

2.4 *Probabilistic Interpretation*

An alternative view of deformable models emerges from casting the model fitting process in a probabilistic framework. This permits the incorporation of prior model and sensor model characteristics in terms of probability distributions. The probabilistic framework also provides a measure of the uncertainty of the estimated shape parameters after the model is fitted to the image data.

Let $\mathbf{u}$ represent the deformable model shape parameters with a prior probability $p(\mathbf{u})$ on the parameters. Let $p(I|\mathbf{u})$ be the imaging (sensor) model—the probability of producing an image I given a model $\mathbf{u}$. Bayes' theorem

$$p(\mathbf{u}|I) = \frac{p(I|\mathbf{u})p(\mathbf{u})}{p(I)} \tag{1.14}$$

expresses the posterior probability $p(\mathbf{u}|I)$ of a model given the image, in terms of the imaging model and the prior probabilities of model and image.

It is easy to convert the internal energy measure (1.2) of the deformable model into a prior distribution over expected shapes, with lower energy shapes being the more likely. This is achieved using a Boltzmann (or Gibbs) distribution of the form

$$p(\mathbf{u}) = \frac{1}{Z_s}\exp(-S(\mathbf{u})), \tag{1.15}$$

where $S(\mathbf{u})$ is the discretized version of $\mathcal{S}(\mathbf{v})$ in (1.2) and Z_s is a normalizing constant (called the partition function). This prior model is then combined with a simple sensor model based on linear measurements with Gaussian noise

$$p(I|\mathbf{u}) = \frac{1}{Z_I}\exp(-\mathrm{P}(\mathbf{u})), \tag{1.16}$$

where $\mathrm{P}(\mathbf{u})$ is a discrete version of the potential $\mathcal{P}(\mathbf{v})$ in (1.3), which is a function of the image $I(x, y)$.

Models may be fitted by finding $\mathbf{u}$ which locally maximize $p(\mathbf{u}|I)$ in (1.14). This is known as the maximum a posteriori solution. With the above construction, it yields the same result as minimizing (1.1), the energy configuration of the deformable model given the image.

The probabilistic framework can be extended by assuming a time-varying prior model, or system model, in conjunction with the sensor model, resulting in a Kalman filter. The system model describes the expected evolution of the shape parameters $\mathbf{u}$ over time. If the equations of motion of the physical snakes model (1.12) are employed as the system model, the result is a sequential estimation algorithm known as “Kalman snakes” [22], which is useful for tracking objects in video.

2.5 Higher-Dimensional Generalizations

Snakes are a special case within the general framework of continuous (multidimensional) deformable models in a Lagrangian dynamics setting that is based on deformation energies in the form of (controlled-continuity) generalized splines [15].

In a p-dimensional domain $\mathbf{x} = (x_1, \ldots, x_p) \in \Re^p$, the natural generalization of the smoothness functional (1.2) defined on a q-dimensional vector of coordinate functions $\mathbf{v}(\mathbf{x}) = [v_1(\mathbf{x}), \ldots, v_q(\mathbf{x})]$ is

$$\mathcal{S}_n(\mathbf{v}) = \frac{1}{2}\int_{\Re^p} \sum_{m=1}^{n} \sum_{|j|=m} \frac{m!}{j_1! \ldots j_p!} w_j(\mathbf{x}) \left| \frac{\partial^m \mathbf{v}(\mathbf{x})}{\partial x_1^{j_1} \ldots \partial x_p^{j_p}} \right|^2 d\mathbf{x}. \tag{1.17}$$

Here, $j = (j_1, \ldots, j_p)$ is a multi-index with $|j| = j_1 + \ldots + j_p$. Note that this functional offers higher-order smoothness by generalizing (1.2) beyond second-order derivatives, to derivatives of order n. Analogous to equation (1.4), assuming that the control functions $\mathbf{w}(\mathbf{x})$ are differentiable to order p, the Euler–Lagrange equation is:

$$\sum_{m=1}^{n} (-1)^m \Delta_w^m \mathbf{v}(\mathbf{x}) + \nabla P(\mathbf{v}(\mathbf{x})) = \mathbf{0}, \tag{1.18}$$

where

$$\Delta_w^m = \sum_{|j|=m} \frac{m!}{j_1! \ldots j_p!} \frac{\partial^m}{\partial x_1^{j_1} \ldots \partial x_p^{j_p}} \left(w_j(\mathbf{x}) \frac{\partial^m}{\partial x_1^{j_1} \ldots \partial x_p^{j_p}} \right) \tag{1.19}$$

is a spatially weighted mth-order iterated Laplacian operator.

Deformable Surfaces:

In the special case $n = 2$, $q = 3$, and $p = 2$, where we define $\mathbf{x} = (x_1, x_2) = (x, y)$ for notational convenience and restrict the 2-dimensional domain to the unit square, (1.17) can be written as

$$\mathcal{S}(\mathbf{v}) = \frac{1}{2} \int_0^1 \int_0^1 w_{10} \left| \frac{\partial \mathbf{v}}{\partial x} \right|^2 + w_{01} \left| \frac{\partial \mathbf{v}}{\partial y} \right|^2 + \\ w_{20} \left| \frac{\partial^2 \mathbf{v}}{\partial x^2} \right|^2 + 2w_{11} \left| \frac{\partial^2 \mathbf{v}}{\partial x \partial y} \right|^2 + w_{02} \left| \frac{\partial^2 \mathbf{v}}{\partial y^2} \right|^2 \, dx \, dy, \quad (1.20)$$

where the subscripts on $\mathbf{v}$ denote its partial derivatives with respect to x and y. This functional, the natural, two-dimensional generalization of the snake energy (1.2), pertains to the problem of deformable surfaces. The physical parameter functions $w_{10}(x, y)$ and $w_{01}(x, y)$ control the tension of the surface, while $w_{20}(x, y)$, $w_{11}(x, y)$, and $w_{02}(x, y)$ control its “rigidity”.

This *thin plate under tension* functional has seen considerable application, notably to visible-surface reconstruction in computer vision [16].

2.6 Connections to Curve Evolution

There is a well-known relationship between classic deformable models and level set methods. The typical curve evolution equations that are computed as level sets correspond to a reduced version of the equations of motion (1.5) that characterize a massless snake $\mu(s) = 0$ with no rigidity $w_2(s) = 0$. This special case results in snakes that, like conventional level set curves, minimize arc length in the metric induced by the image.

The analogous special case in 3D is surfaces that minimize surface area in the metric induced by a volume image. This is equivalent to a massless deformable surface governed by the functional $\mathcal{S}_n$ in (1.20) with $w_{20}(x, y) = w_{11}(x, y) = w_{02}(x, y) = 0$; i.e., the membrane functional.

3 Topology-Adaptive Deformable Models

As physics-based models of nonrigid solids, deformable models have had an enormous impact in medical image analysis [8]. The complexity of human anatomy, comprising numerous nonrigid organs with intricate substructures at multiple scales of resolution, means that deformable models must generally be able to deal with non-simple and even non-constant topologies; for example, in serial reconstruction the topology of a segmented cross-section can vary dramatically as the image data are sliced in different ways. Unfortunately, without additional machinery, classic, Lagrangian deformable models cannot alter their prescribed topology.

Level set image segmentation methods were motivated by the need for a related technique that makes no prior assumption about the topology of the underlying object of interest. To this end, level-set methods formulate boundary curve or surface estimation as an Eulerian problem defined over the entire image domain. The dimensionality of the Eulerian problem typically is one greater than the dimensionality of the associated Lagrangian problem for deformable models. The primary feature of this approach is that the higher-dimensional hypersurface remains a simple function, even as the level set changes topology (or ceases to be simply connected). Hence, topological changes are handled naturally.

Topology-adaptive deformable models are an alternative approach to non-fixed topology [10, 9]. They circumvent the increased dimensionality of the level set methods while retaining the strengths of standard parametric deformable models, including the explicit geometric representation, the natural user interaction mechanisms, and the constraint mechanisms implemented through energy or force functions. Topology-adaptive deformable models can segment and reconstruct some of the most complex biological structures from 2D and 3D images. In this section, I will first review *topology-adaptive snakes*, or T-snakes, followed by *topology-adaptive deformable surfaces*, or T-surfaces.

3.1 Topology-Adaptive Snakes

T-snakes are hybrid models that combine Lagrangian, parametric snakes with aspects of the Eulerian, level set approach. They employ an Affine Cell Image Decomposition (ACID) of the image domain. The ACID extends the abilities of conventional snakes, enabling topological flexibility, among other features [10]. In particular, the ACID framework enables a novel snake reparameterization mechanism, which enables snakes to "flow" into geometrically complex objects, conforming to the object boundaries (Fig. 3(a)). One or more T-snakes can be dynamically created or destroyed and can seamlessly split or merge as necessary in order to adapt to object topology (Fig. 3(b)–(e)). See references [7] for numerous additional examples of T-snakes applied to images.

As a T-snake deforms under the influence of external and internal forces, it is systematically reparameterized with a new set of nodes and elements. This is done by efficiently computing the intersection points of the model with the superposed affine cell grid; for example, the Coxeter-Freudenthal decomposition (Fig. 4(a)). At the end of each deformation step, the nodes have moved relative to the grid cell edges (Fig. 4(b)–(d)). In phase I of the reparameterization algorithm, the intersection points between the T-snake elements and the grid cell edges are computed. These intersection points will become the nodes of the new T-snake. In phase II, grid cell vertices that have moved from the exterior to the interior of the T-snake are marked as "on"; in this manner, the interior of a T-snake is continuously tracked.

A closed T-snake defines a region. When a T-snake bounded region collides with itself or with another T-snake region, or splits into two or more subregions, (or shrinks and disappears,) a topological transformation must take place. Topology changes are performed automatically via the ACID (Fig. 3(b)–(e)). The boundary can always be determined unambiguously by keeping track of the inside grid

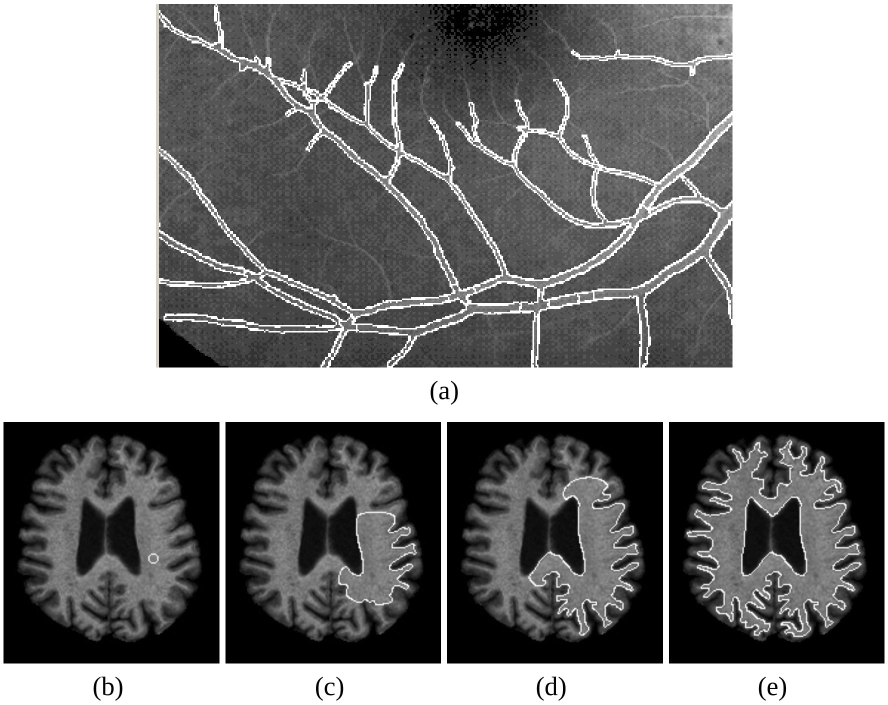

FIGURE 3. Segmentation with T-snakes. (a) T-snakes segmenting blood vessels in a retinal angiogram. Geometric flexibility allows the T-snakes to grow into the complex vessel shapes. (b–e) T-snake segmenting gray-matter/white-matter interface and ventricles in an MR brain image slice. The initially circular T-snake (b) changes its topology to a highly deformed annular region (e).

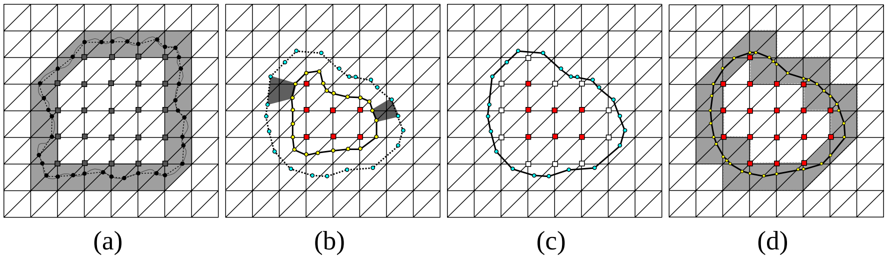

FIGURE 4. (a) Simplicial approximation (dashed-line) of an object contour (solid-line) using a Freudenthal triangulation. The model nodes (intersection points) are marked and the boundary triangles are shaded. (b-d) Illustration of the T-snake reparameterization process. (b) Shaded regions show examples of grid vertices that are turned on by the expanding contour, (c) new inside grid vertices (white) added to current inside vertices (dark), (d) new contour after one deformation step showing new grid intersections, inside grid vertices, and boundary grid cells (gray shaded).

vertices (and hence the boundary grid cells) and re-establishing the correspondence of the T-snake boundary with the grid after every deformation step. New elements are constructed based on the “signs” (i.e. inside or outside) of the grid vertices in each boundary cell and from the intersection points computed in phase I, such that the inside and outside grid vertices in these cells are separated by a single line.

Compared to conventional snakes, the T-snake is relatively insensitive to its initial placement within regions of interest in the image. It flows into complex shapes, modifying its topology as necessary in order to fit the relevant image data. The ACID provides a principled, computational geometry framework for topological transformations, but disabling the ACID reduces the T-snakes model to a conventional parametric snake model. Consequently, T-snakes also incorporate shape constraints in the form of energy functionals and applied forces. An important advantage of the latter is user control, which caters to medical image analysis, where it is often essential for an expert user to be able to control and refine the segmentation process in an interactive manner.

3.2 Topology-Adaptive Deformable Surfaces

The main components of the three-dimensional T-surfaces formulation (see [9] for the details) are analogous to those for the two-dimensional T-snakes. The first component is a discrete form of the conventional parametric deformable surfaces [6]. The second component is the extension of the ACID framework to three dimensions using simplicial (tetrahedral) cells or nonsimplicial (e.g. hexahedral) cells.[2] The third component of T-surfaces is a reparameterization process analogous to the one for T-snakes. To determine if a T-surface triangular element intersects a grid cell edge, a standard ray-triangle intersection algorithm is used and local neighborhood searches are employed to speed up the process. The user can interact with a T-surface by applying 3D interaction forces, by applying 2D interaction forces to cross-sections of the surface that are overlaid on image slices thorough the 3D dataset.

Fig. 5 demonstrates the topological adaptability of a T-surface when applied to a $120 \times 128 \times 52$ CT volume image of a human vertebra phantom. This example uses a $32 \times 30 \times 13$ cell grid (where each cubical cell is divided into 6 tetrahedra).

4 Generalized Deformable Models

Next, I will review the systematic Lagrangian formulation of physics-based deformable models that are built upon the standard geometric substrates of computer

[2] Most nonsimplicial methods employ a rectangular tessellation of space. The formulation of T-surfaces using a non-simplicial grid is essentially identical to its simplicial counterpart except for the addition of the disambiguation scheme.

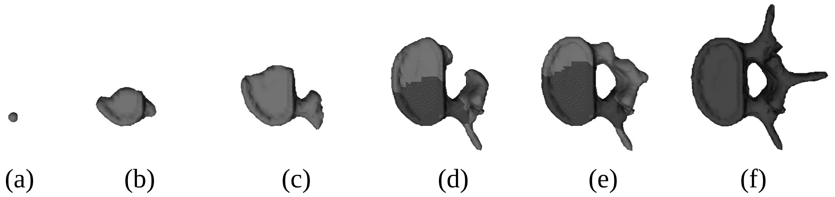

FIGURE 5. T-surface segmenting vertebra phantom from CT volume image [9].

graphics and geometric modeling. These explicit geometric substrates are by no means trivial; certainly more complicated than those encountered in the previous two sections. Nonetheless they yield generalized varieties of deformable models possessing geometric degrees of freedom and novel physical behaviors that are beneficial in certain applications to computer-aided geometric design, computer graphics animation, and computer vision.

The first part of this section reviews *Dynamic Non-Uniform Rational B-Splines* (D-NURBS). The second part reviews *Deformable Superquadrics*. There exist other interesting generalized deformable model formulations that I will not review because of lack of space, among them "united snakes"—a hybrid between snakes and "livewire" segmentation tools based on dynamic programming [5]—and dynamic free-form deformations (FFDs) [3].

4.1 Dynamic NURBS

Deformable models have given impetus to a new, physics-based paradigm for computer-aided geometric design. The natural behavior of deformable models suggests a "computational modeling clay" metaphor which is particularly intuitive in the sculpting of free-form shapes [17]. An important goal, however, is to formulate physics-based modeling primitives that generalize the purely geometric free-form shape primitives employed in conventional shape design systems. To this end, we have developed a physics-based generalization of industry-standard, non-uniform rational B-splines (NURBS) and associated constraint methods for physics-based geometric design and interactive sculpting [21]. The shape parameters of conventional, geometric NURBS play the role of generalized (physical) coordinates in "Dynamic NURBS" (D-NURBS). We introduce time, mass, and deformation energy into the standard NURBS formulation and employ Lagrangian dynamics to arrive at the system of nonlinear ordinary differential equations that govern the shape and motion of D-NURBS.

D-NURBS Curves:

A kinematic NURBS curve extends the geometric NURBS definition by explicitly incorporating time. The kinematic curve is a function of both the parametric variable u and time t:

$$\mathbf{c}(u,t) = \frac{\sum_{i=0}^{n} \mathbf{p}_i(t) w_i(t) B_{i,k}(u)}{\sum_{i=0}^{n} w_i(t) B_{i,k}(u)}, \tag{1.21}$$

where the $B_{i,k}(u)$ denote the usual recursively defined piecewise basis functions, $\mathbf{p}_i(t)$ are the $n+1$ control points, and $w_i(t)$ are associated non-negative weights. Assuming basis functions of degree $k-1$, the curve has $n+k+1$ knots t_i in non-decreasing sequence: $t_0 \leq t_1 \leq \ldots \leq t_{n+k}$.

To simplify notation, we define the vector of generalized coordinates $\mathbf{q}_i(t)$ and weights $w_i(t)$ as

$$\mathbf{q}(t) = \left[\begin{array}{cccccc} \mathbf{p}_0^\top & w_0 & \cdots & \mathbf{p}_n^\top & w_n \end{array}\right]^\top .$$

We then express the spline curve (1.21) as $\mathbf{c}(u, \mathbf{q})$.

The velocity of the kinematic spline is

$$\dot{\mathbf{c}}(u, \mathbf{q}) = \mathbf{J}\dot{\mathbf{q}}, \tag{1.22}$$

where the overstruck dot denotes a time derivative and $\mathbf{J}(u, \mathbf{q})$ is the Jacobian matrix. Because $\mathbf{c}$ is a 3-component vector-valued function and $\mathbf{q}$ is a $4(n+1)$ dimensional vector, $\mathbf{J}$ is the $3 \times 4(n+1)$ matrix

$$\mathbf{J} = \left[\begin{array}{cccc} \cdots & \left[\begin{array}{ccc} \frac{\partial \mathbf{c}_x}{\partial \mathbf{p}_{i,x}} & 0 & 0 \\ 0 & \frac{\partial \mathbf{c}_y}{\partial \mathbf{p}_{i,y}} & 0 \\ 0 & 0 & \frac{\partial \mathbf{c}_z}{\partial \mathbf{p}_{i,z}} \end{array}\right] & \frac{\partial \mathbf{c}}{\partial w_i} & \cdots \end{array}\right], \tag{1.23}$$

where

$$\frac{\partial \mathbf{c}_x}{\partial \mathbf{p}_{i,x}} = \frac{\partial \mathbf{c}_y}{\partial \mathbf{p}_{i,y}} = \frac{\partial \mathbf{c}_z}{\partial \mathbf{p}_{i,z}} = \frac{w_i B_{i,k}}{\sum_{j=0}^{n} w_j B_{j,k}};$$

$$\frac{\partial \mathbf{c}}{\partial w_i} = \frac{\sum_{j=0}^{n} (\mathbf{p}_i - \mathbf{p}_j) w_j B_{i,k} B_{j,k}}{(\sum_{j=0}^{n} w_j B_{j,k})^2}.$$

The subscripts x, y, and z denote the components of a 3-vector.

The equations of motion of our D-NURBS are derived from Lagrangian dynamics. Applying the Lagrangian formulation to D-NURBS curves, we obtain the second-order nonlinear equations of motion

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{f}_q + \mathbf{g}_q, \tag{1.24}$$

where $\mathbf{M}(\mathbf{q})$ is the mass matrix, $\mathbf{D}(\mathbf{q})$ is the damping matrix, and $\mathbf{K}(\mathbf{q})$ is the stiffness matrix. The $N \times N$ mass and damping matrices are given by

$$\mathbf{M}(\mathbf{q}) = \int \mu \mathbf{J}^\top \mathbf{J}\, du; \qquad \mathbf{D}(\mathbf{q}) = \int \gamma \mathbf{J}^\top \mathbf{J}\, du, \tag{1.25}$$

where $\mu(u, v)$ is the prescribed mass density function over the parametric domain of the surface and $\gamma(u, v)$ is the prescribed damping density function. To define an elastic potential energy for the D-NURBS curve, we use the snake energy (1.2). This yields the $N \times N$ stiffness matrix

$$\mathbf{K}(\mathbf{q}) = \int w_1(u) \mathbf{J}_u^\top \mathbf{J}_u + w_2(u) \mathbf{J}_{uu}^\top \mathbf{J}_{uu}\, du, \tag{1.26}$$

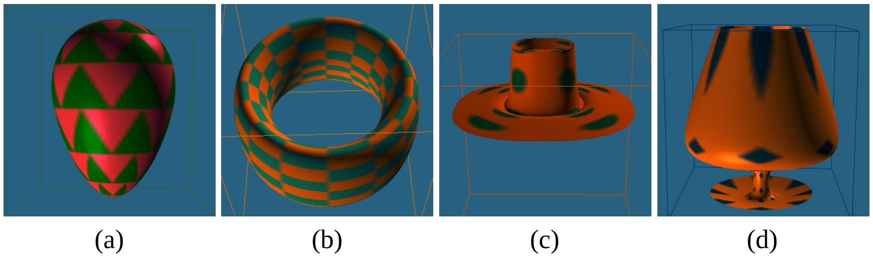

FIGURE 6. Interactive Sculpting of D-NURBS Swung Surfaces. Open and closed surfaces shown were sculpted interactively from prototype shapes noted in parentheses (a) Egg shape (sphere). (b) Deformed toroid (torus). (c) Hat (open surface). (d) Wine glass (cylinder) (from [13]).

where the subscripts on $\mathbf{J}$ denote parametric partial derivatives. As in snakes, the elasticity functions $w_1(u)$ and $w_2(u)$ control tension and rigidity, respectively. The generalized force $\mathbf{f}_p(\mathbf{q}) = \int \mathbf{J}^\top \mathbf{f}(u,t)\,du$, where $\mathbf{f}(u,t)$ is the applied force distribution. Because of the geometric nonlinearity, generalized inertial forces $\mathbf{g}_q(\mathbf{q})$ are also associated with the models (for the details, see [21]).

D-NURBS Surfaces:

Beyond D-NURBS curves, we have formulated three varieties of D-NURBS surfaces: tensor-product D-NURBS surfaces [21], swung D-NURBS surfaces [13], and triangular D-NURBS surfaces [14]. Examples of topologically different shapes interactively sculpted using D-NURBS swung surfaces are shown in Fig. 6.

For example, in analogy to the kinematic curve of (1.21), a tensor-product D-NURBS surface

$$\mathbf{s}(u,v,t) = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n} \mathbf{p}_{i,j}(t) w_{i,j}(t) B_{i,k}(u) B_{j,l}(v)}{\sum_{i=0}^{m}\sum_{j=0}^{n} w_{i,j}(t) B_{i,k}(u) B_{j,l}(v)} \tag{1.27}$$

generalizes the standard, geometric NURBS surface. The $(m+1)(n+1)$ control points $\mathbf{p}_{i,j}(t)$ and weights $w_{i,j}(t)$, which are functions of time, comprise the tensor-product D-NURBS surface generalized coordinates. We concatenate these $N = 4(m+1)(n+1)$ coordinates into the vector $\mathbf{p}(t)$. Analogous to (1.22), we have $\dot{\mathbf{s}}(u,v,\mathbf{p}) = \mathbf{J}\dot{\mathbf{p}}$, where $\mathbf{J}(u,v,\mathbf{p})$ is the $3 \times N$ Jacobian matrix of the D-NURBS surface with respect to $\mathbf{p}$. Note that $\mathbf{J}$ is now a $3 \times 4(m+1)(n+1)$ matrix. Refer to [21] for the mathematical details and examples.

4.2 *Deformable Superquadrics*

We will next review a systematic Lagrangian approach for creating dynamic solid models capable of realistic physical behaviors, starting from the common, globally parameterized solid primitives, such as spheres, cylinders, cones, or superquadrics. Such primitives can deform kinematically in simple ways; for example, a geometric cylinder deforms as its radius or length is changed. To gain additional

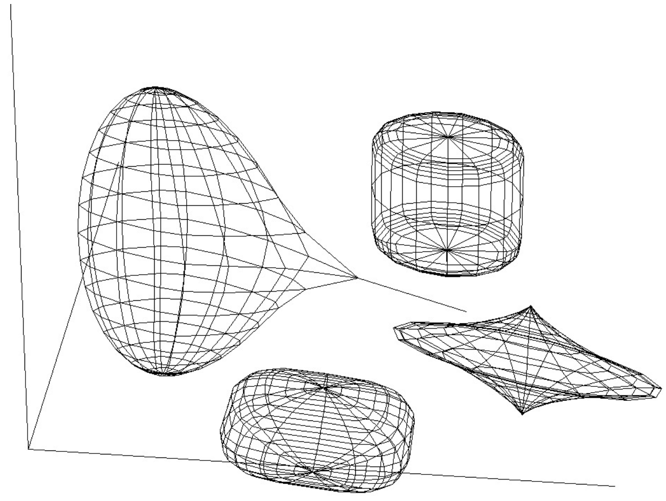

FIGURE 7. Interaction with deformable superquadrics.

modeling power we allow the solid primitives to undergo parameterized global deformations (bends, tapers, twists, shears, etc.). To further enhance the geometric flexibility, we permit local free-form deformations. All of the geometric parameters are once again collected into a vector of generalized coordinates, along with the six degrees of freedom of rigid-body motion. As usual, Lagrange equations of motion govern the dynamics of these parameterized deformable models, dictating the evolution of the generalized coordinates in response to applied forces, so as to exhibit correct mechanical behavior subject to prescribed mass distributions, elasticities, and energy dissipation rates. Such models are useful in physics-based computer animation.

For example, Fig. 7 shows several deformable superquadrics. A superellipsoid is deforming in response to the traction from a linear spring attached to its surface and pulled interactively. In general, the models are abstract viscoelastic solids. It is possible, for instance, to mold a supersphere into any of the deformable superquadrics shown in the figure by applying appropriate forces.

Geometric Formulation:

A superquadric is a solid model whose intrinsic (material) coordinates are $\mathrm{u} = (u, v, w)$. Referring to Fig. 8, $\mathbf{x}(\mathrm{u}, t) = (x_1(\mathrm{u}, t), x_2(\mathrm{u}, t), x_3(\mathrm{u}, t))^\top$ gives the positions of points on the model relative to the fixed reference frame Φ. We can write

$$\mathbf{x} = \mathbf{c} + \mathbf{R}\mathbf{p}, \tag{1.28}$$

where $\mathbf{p}(\mathrm{u}, t)$ denotes the positions of points relative to the noninertial, model-centered frame ϕ whose instantaneous position is $\mathbf{c}(t)$ and orientation relative to

Φ is given by the rotation matrix $\mathbf{R}(t)$. We further express

$$\mathbf{p} = \mathbf{s} + \mathbf{d}, \tag{1.29}$$

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

$$\mathbf{s} = \mathbf{T}(\mathbf{e}(\mathrm{u};\ a_1, a_2, \ldots);\ b_1, b_2, \ldots). \tag{1.30}$$

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

$$\mathbf{q}_s = (a_1, a_2, \ldots, b_1, b_2, \ldots)^\top. \tag{1.31}$$

Next, we express the displacement as a linear combination of basis functions $\mathbf{b}_i(\mathrm{u})$. The basis functions can be local or global; however, finite element shape functions are the natural choice for representing *local deformations*

$$\mathbf{d} = \mathbf{S}\mathbf{q}_d. \tag{1.32}$$

Here $\mathbf{S}$ is a shape matrix whose entries are the shape functions and

$$\mathbf{q}_d = (\ldots, \mathbf{d}_i^\top, \ldots)^\top \tag{1.33}$$

is the vector of local deformation parameters. Typically, finite elements have nodes at their vertices, and the parameter $\mathbf{q}_i$ denotes a displacement vector associated with node i of the model.

In [19, 11] we provide the formulas for a superquadric ellipsoid $\mathbf{e}$ with tapering, bending, shearing, and twisting deformations.

Kinematics and Dynamics:

To convert the geometric representation into a physical model that responds dynamically to forces, we first consider the kinematics implied by the geometry and then introduce mass, damping, and elasticity into the model to derive its mechanics.

The velocity of points on the model is given by,

$$\dot{\mathbf{x}} = \dot{\mathbf{c}} + \dot{\mathbf{R}}\mathbf{p} + \mathbf{R}\dot{\mathbf{p}} = \dot{\mathbf{c}} + \mathbf{B}\dot{\boldsymbol{\theta}} + \mathbf{R}\dot{\mathbf{s}} + \mathbf{R}\mathbf{S}\dot{\mathbf{q}}_d, \tag{1.34}$$

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

$$\mathbf{x} = \mathbf{c} + \mathbf{R}(\mathbf{s} + \mathbf{d}) = \mathbf{h}(\mathbf{q}), \tag{1.35}$$

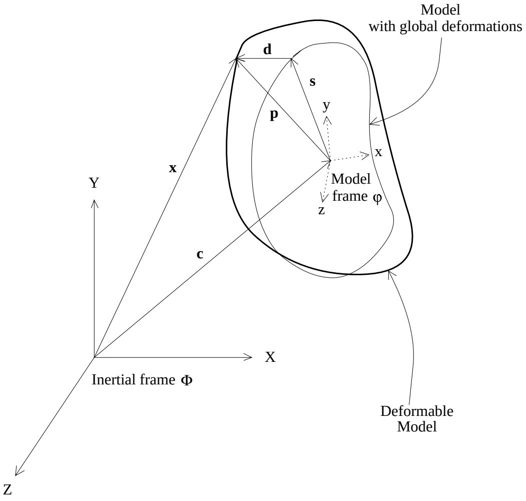

FIGURE 8. Geometric structure.

$$\dot{\mathbf{x}} = [\mathbf{I}\ \mathbf{B}\ \mathbf{RJ}\ \mathbf{RS}]\dot{\mathbf{q}} = \mathbf{L}\dot{\mathbf{q}}, \tag{1.36}$$

where

$$\mathbf{q} = (\mathbf{q}_c^\top, \mathbf{q}_\theta^\top, \mathbf{q}_s^\top, \mathbf{q}_d^\top)^\top, \tag{1.37}$$

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

To specify the dynamics, we introduce a mass distribution $\mu(\mathrm{u})$ over the model and assume that the material is subject to frictional damping. We also assume that the material may deform elastically or viscoelastically [18]. Applying the Lagrangian mechanics approach, we obtain second-order equations of motion which take the form

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{g}_{\mathbf{q}} + \mathbf{f}_{\mathbf{q}}. \tag{1.38}$$

The mass matrix $\mathbf{M} = \int \mu \mathbf{L}^\top \mathbf{L}\, d\mathrm{u}$. The stiffness matrix $\mathbf{K}$ may be obtained from a deformation strain energy $(\mathbf{q}^\top \mathbf{K}\mathbf{q})/2$. The Raleigh damping matrix $\mathbf{D} = \alpha\mathbf{M} + \beta\mathbf{K}$. The generalized inertial forces $\mathbf{g}_{\mathbf{q}} = -\int \mu \mathbf{L}^\top \dot{\mathbf{L}}\dot{\mathbf{q}}\, d\mathrm{u}$ include generalized centrifugal, Coriolis, and transverse forces due to the dynamic coupling between $\mathbf{q}_\theta$, $\mathbf{q}_s$, and $\mathbf{q}_d$. Finally, $\mathbf{f}_{\mathbf{q}} = \int \mathbf{L}^\top \mathbf{f}\, d\mathrm{u}$ are generalized external forces associated with the components of $\mathbf{q}$, where $\mathbf{f}(\mathrm{u}, t)$ is the force distribution applied to the model. See [19, 11] for explicit formulas for the above matrices and vectors.

Multibody Constraints:

Deformable superquadric models raise interesting challenges related to the application of constraints in order to construct composite models and control animation. We describe a method for computing generalized constraint forces between our models which is based on Baumgarte's constraint stabilization technique. Our algorithm may be used to assemble complex objects satisfying constraints from initially mispositioned and misshaped parts, and it enables us to construct and animate articulated objects composed of rigid or nonrigid components. We have applied these multibody models to computer animation [11] and computer vision [12].

We can extend (1.38) to account for the motions of composite models with interconnected deformable parts by forming a composite generalized coordinate vector $\mathbf{q}$, as well as force vectors $\mathbf{g}_\mathbf{q}$ and $\mathbf{f}_\mathbf{q}$ for an n-part model by concatenating the $\mathbf{q}_i$, $\mathbf{g}_{\mathbf{q}_i}$, and $\mathbf{f}_{\mathbf{q}_i}$ associated with each part $i = 1, \ldots, n$. Similarly, the composite matrices $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ for the n-part model are block diagonal matrices with submatrices $\mathbf{M}_i$, $\mathbf{D}_i$, and $\mathbf{K}_i$, respectively, for each part i. The method solves the composite equations of motion $\mathbf{M}\ddot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{g}_\mathbf{q} + \mathbf{f}_\mathbf{q} - \mathbf{C}_\mathbf{q}^\top \boldsymbol{\lambda}$. The generalized constraint forces $\mathbf{f}_{g_c} = -\mathbf{C}_\mathbf{q}^\top \boldsymbol{\lambda}$ acting on the parts stem from the holonomic constraint equations

$$\mathbf{C}(\mathbf{q}, t) = \mathbf{0}; \tag{1.39}$$

i.e., $\mathbf{C} = [\mathbf{C}_1^\top, \mathbf{C}_2^\top, \ldots, \mathbf{C}_k^\top]^\top$ expresses k constraints among the n parts of the model. The term $\mathbf{C}_\mathbf{q}^\top$ is the transpose of the constraint Jacobian matrix and $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1^\top, \ldots, \boldsymbol{\lambda}_n^\top)^\top$ is a vector of Lagrange multipliers that must be determined.

To obtain an equal number of equations and unknowns, we differentiate (1.39) twice with respect to time, yielding $\boldsymbol{\sigma} = \mathbf{C}_\mathbf{q}\ddot{\mathbf{q}} = -\mathbf{C}_{tt} - (\mathbf{C}_\mathbf{q}\dot{\mathbf{q}})_\mathbf{q}\dot{\mathbf{q}} - 2\mathbf{C}_{\mathbf{q}_t}\dot{\mathbf{q}}$. The augmented equations of motion are

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}_\mathbf{q}^\top \\ \mathbf{C}_\mathbf{q} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{D}\dot{\mathbf{q}} - \mathbf{K}\mathbf{q} + \mathbf{g}_\mathbf{q} + \mathbf{f}_\mathbf{q} \\ \boldsymbol{\sigma} - 2\alpha\dot{\mathbf{C}} - \beta^2\mathbf{C} \end{bmatrix}. \tag{1.40}$$

Fast constraint stabilization means choosing $\beta = \alpha$ to obtain the critically damped solution $\mathbf{C}(\mathbf{q}, 0)e^{-\alpha t}$ which, for given α, has the quickest asymptotic decay towards constraint satisfaction $\mathbf{C} = \mathbf{0}$.

5 Conclusion

Deformable models are a rich family of physics-based modeling primitives that have seen extensive use in computer vision and graphics. They have also been applied heavily in the associated areas of medical image analysis and geometric design. In this chapter, I have reviewed

1. the formulation of classic deformable models, including snakes and higher-dimensional models;
2. topology-adaptive deformable models that offer the topological flexibility associated with level set methods, but retain the parametric, Lagrangian formulation;
3. formulations of generalized Lagrangian deformable models that are built on nontrivial geometric foundations, such as non-uniform rational B-splines and superquadrics.

Lagrangian models of solids complement Eulerian models of fluids in continuum mechanics. Deformable models and level set methods are complementary techniques in precisely the same sense. Together, these methods have given impetus to a voluminous and rapidly growing literature. They continue to have a promising future in multiple application areas.

Acknowledgments: I thank Nikos Paragios and Stan Osher for encouraging me to prepare this chapter. I am grateful to many colleagues who have collaborated with me in the development of deformable model theory and applications; their names appear in the citations: Carlbom, Faloutsos, Fleischer, Kass, Liang, McInerney, Metaxas, Qin, Szeliski, Witkin. In view of the material presented in this chapter, I would like especially to acknowledge the key contributions of three former PhD students: Tim McInerney, whose dissertation developed topology-adaptive deformable models, Dimitri Metaxas, whose dissertation developed deformable superquadrics, and Hong Qin, whose dissertation developed D-NURBS.

6 REFERENCES

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

[2] I. Carlbom, D. Terzopoulos, and K. Harris. Computer-assisted registration, segmentation, and 3D reconstruction from images of neuronal tissue sections. *IEEE Transactions on Medical Imaging*, 13(2):351–362, 1994.

[3] P. Faloutsos, M. van de Panne, and D. Terzopoulos. Dynamic free-form deformations for animation synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 3(3):201–214, September 1997.

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

[5] J. Liang, T. McInerney, and D. Terzopoulos. Medical image analysis with topologically adaptive snakes. In *Proc. Seventh International Conf. on Computer Vision (ICCV'99)*, pages 933–940, Kerkyra, Greece, September 1999.

[6] T. McInerney and D. Terzopoulos. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis. *Computerized Medical Imaging and Graphics*, 19(1):69–83, January 1995.

[7] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. In *Proc. Fifth International Conf. on Computer Vision (ICCV'95)*, pages 840–845, Cambridge, MA, June 1995.

[8] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1(2):91–108, 1996.

[9] T. McInerney and D. Terzopoulos. Topology adaptive deformable surfaces for medical image volume segmentation. *IEEE Transactions on Medical Imaging*, 18(10):840–850, October 1999.

[10] T. McInerney and D. Terzopoulos. T-snakes: Topology adaptive snakes. *Medical Image Analysis*, 4(2):73–91, June 2000.

[11] D. Metaxas and D. Terzopoulos. Dynamic deformation of solid primitives with constraints. *Computer Graphics (Proc. SIGGRAPH'92)*, 26(2):309–312, 1992.

[12] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993.

[13] H. Qin and D. Terzopoulos. Dynamic NURBS swung surfaces for physics-based shape design. *Computer-Aided Design*, 27(2):111–127, 1995.

[14] H. Qin and D. Terzopoulos. Triangular NURBS and their dynamic generalizations. *Computer-Aided Geometric Design*, 14:325–347, 1997.

[15] D. Terzopoulos. On matching deformable models to images. Technical Report 60, Schlumberger Palo Alto Research, 1986. Reprinted in *Topical Meeting on Machine Vision*, Technical Digest Series, Vol. 12 (Optical Society of America, Washington, DC) 1987, 160-167.

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

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

[18] D. Terzopoulos and K. Fleischer. Modeling inelastic deformation: Viscoelasticity, plasticity, fracture. *Computer Graphics (Proc. SIGGRAPH'88)*, 22(4):269–278, 1988.

[19] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, 1991.

[20] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics (Proc. SIGGRAPH'87)*, 21(4):205–214, 1987.

[21] D. Terzopoulos and H. Qin. Dynamic NURBS with geometric constraints for interactive sculpting. *ACM Transactions on Graphics*, 13(2):103–136, 1994.

[22] D. Terzopoulos and R. Szeliski. Tracking with Kalman snakes. In *Active Vision*, pages 3–20. MIT Press, Cambridge, MA, 1992.

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