Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics

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

Abstract— **This paper presents a physically based approach to fitting complex 3-D shapes using a new class of dynamic models that can deform both locally and globally. We formulate deformable superquadrics which incorporate the global shape parameters of a conventional superellipsoid with the local degrees of freedom of a spline. The local/global representational power of a deformable superquadric simultaneously satisfies the conflicting requirements of shape reconstruction and shape recognition. The model's (six) global deformational degrees of freedom capture gross shape features from visual data and provide salient part descriptors for efficient indexing into a database of stored models. The local deformation parameters reconstruct the details of complex shapes that the global abstraction misses. The equations of motion which govern the behavior of deformable superquadrics make them responsive to externally applied forces. We fit models to visual data by transforming the data into forces and simulating the equations of motion through time to adjust the translational, rotational, and deformational degrees of freedom of the models. We present model fitting experiments involving 2D monocular image data and 3D range data.**

Index Terms— **Computer vision, physically based modeling, object representation, deformable models, local and global deformations, superquadrics, splines, simulated forces, 3D model fitting, finite element analysis.**

I. Introduction

THE RECONSTRUCTION of shape and the recognition of objects have preoccupied computational vision researchers for several decades. Despite the large body of work on 3D modeling, most models of shape lack the descriptive power to bridge the gap between reconstruction and recognition. The difficulty is one of conflicting requirements.

General-purpose shape reconstruction in low-level visual processing requires models with broad geometric coverage. Reconstruction models must extract meaningful information from noisy sensor data while making the weakest possible assumptions about observed shapes. Generalized spline models that can deform locally subject to generic continuity constraints appear to be well suited to shape reconstruction. By contrast, object recognition is a higher level process that necessitates drastic information reduction and shape abstraction in order to support efficient matching in object databases of manageable size. Volumetric primitives such as spheres, cylinders, and prisms seem appropriate for object recognition since they can decompose composite shapes into natural parts that are compactly expressible using a small set of parameters.

In this paper, we propose an approach to shape modeling that simultaneously satisfies the requirements of reconstruction and recognition and promises a fluent transition between these two aspects of vision. We develop a new family of modeling primitives that have the following features.

Free-Form and Parameterized Geometry: Geometric design makes extensive use of both the free-form and parameterized modeling paradigms. The canonical primitives of these complementary approaches are, respectively, splines with local shape variables and volumetric forms with global shape parameters. One of the goals of our work is to develop a class of hybrid models whose underlying geometric structure intimately combines free-form and parameterized representations. In particular, the present paper combines membrane splines with parameterized superquadric ellipsoids to create a new family of models we dub *deformable superquadrics.*

Local and Global Deformations: Much of the expressive power of modeling primitives stems from their ability to deform into desired shapes. Spline models are free-form because their local shape control variables provide many local degrees of freedom. Consequently, splines have the flexibility to assume diverse shapes, i.e., they have broad geometric coverage. The allowable shapes of parameterized models, on the other hand, are relatively tightly constrained according to a few global parameters such as lengths, radii, and aspect ratios. Our modeling method interprets such parameters as a set of global deformational degrees of freedom. Moreover, we augment our models with the local deformational capabilities of splines. In particular, deformable superquadrics are able to deform both globally like superquadric ellipsoids and locally like membrane splines.

Physics and Dynamics: Concepts from analytic, differential, and computational geometry have fueled much of the shape representation research in vision. Recently, however, some vision researchers have begun to realize that geometry, while adequate for describing the shapes of static objects, is often insufficient when it comes to analyzing the motions and interactions of complex objects. Following our prior work, we remedy the situation by turning to computational physics. In addition to geometry, the formulation of our models includes simulated forces, masses, strain energies, and other physical quantities. Physically based models are fundamentally dynamic, and the behavior of deformable superquadrics is governed by the laws of rigid and nonrigid dynamics expressed through a set of Lagrangian equations of motion.

Manuscript received August 30, 1990; revised November 29, 1990. This work was supported by the Natural Sciences and Engineering Council of Canada and a University of Toronto Open Fellowship.

The authors are with the Department of Computer Science, University of Toronto, Toronto, Canada M5S 1A4.

IEEE Log Number 9100894.

0162-8828/91/0700-0703$01.00 © 1991 IEEE

Forces and Interaction: The numerical simulation of the equations of motion determines the evolution of the degrees of freedom of our models under the action of simulated forces and constraints. Forces provide a general and highly intuitive means for coupling the degrees of freedom of a model to various data sets, such as intensity and range data. In response to forces originating at datapoints, the models position and orient themselves properly in space and deform away from their rest shapes to conform to the dataset. In applications where user control over models is desirable, physically based control offers much more than the option of manually adjusting geometric parameters. The machinery supporting dynamics provides a facile interface to the models through the use of force interaction tools. Hence, another objective of our approach to modeling is to support interactive dynamics through the use of efficient numerical simulation methods.

Detail and Abstraction: The local and global deformation parameters along with rigid transformations comprise the degrees of freedom of our dynamic models. The equations of motion permit external forces to position and orient deformable superquadrics freely in space and shape them through the global degrees of freedom—a translation vector, a quaternion (rotation), a scale, three radial aspects, and two squareness parameters—but the forces can also deform the models like splines via the local degrees of freedom. The local degrees of freedom of deformable superquadrics allow the reconstruction of fine scale structure and the natural irregularities of real world data, whereas the global degrees of freedom capture the salient features of shape that are innate to natural parts and appropriate for matching against object prototypes. Our models are therefore suitable for use in both visual reconstruction and recognition tasks.

As an illustration of some of the above ideas, Fig. 1 shows a snapshot of an interactive 3D world inhabited by deformable superquadrics. Through mouse control, the user can initialize models, change their global deformation parameters, apply forces to them, and move the viewpoint. The figure illustrates four deformable superquadrics with different settings for the global deformation parameters. The model at the left is being pulled by a stretchy spring (displayed as a line) activated and dragged by the mouse (arrow). The spring force causes local and global deformations in the model.

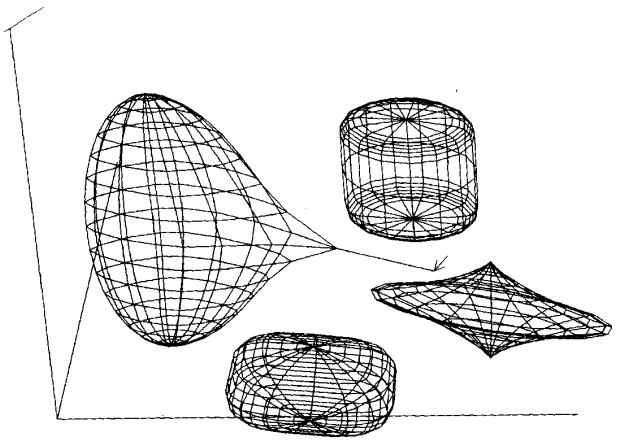


Fig. 1. Interactions with deformable superquadrics.

A. Overview

Section II provides a background for our work in the context of related research. Section III formulates a general set of dynamic equations governing the motion of deformable superquadrics under the action of externally applied forces. Section IV describes a simplified version of these equations that are suitable for vision applications along with their numerical simulation. Section V discusses techniques for converting visual data into forces that can be applied to deformable superquadrics in data fitting scenarios. Section VI presents experimental results demonstrating the fitting of models to 2D monocular image data and 3D range data. Section VII draws conclusions from our work.

II. Background

After more than a decade of research, the notion of early visual reconstruction as a data fitting problem using generalized spline models is now in a highly evolved state of development, most evidently so in the context of the surface reconstruction problem [4], [18], [20]. Generalized spline techniques underly the notion of regularization and its application to a variety of reconstruction problems in early vision [14], [19]. The many degrees of freedom and local deformation properties of generalized splines allow them to conform to low-level visual data with ease.

On another front, much effort has gone into the search for suitable models for the purposes of object recognition. Biederman [3] reports the results of psychophysical experiments, suggesting that the recovery of arrangements of two or three major primitive components or parts results in fast recognition of objects, even when the objects are observed from different viewpoints, are occluded, or are unfamiliar. Parameterized part models capture the structure of the world by describing meaningful chunks of data in terms of a few parameters. Such models are beneficial for object representation since dealing with a manageable number of parameters simplifies the problem of indexing into a database of stored models and verifying match hypotheses.

Throughout the 1970's, the research of Binford and his coworkers on generalized cylinders focused on the problem of recovering parameterized models of objects and led to vision systems such as ACRONYM, which use reasoning to recover parameterized parts [5]. Marr and Nishihara [10] were among the first to propose a hierarchical representation of objects in terms of parts. Their work uses generalized cylinders to describe each part, thereby limiting the scope of the representation to objects adequately describable as collections of generalized cylinders.

Motivated by the generalized cylinder idea and the need to go beyond geometry to exploit computational physics in the modeling process, Terzopoulos *et al.* [22] propose a deformable cylinder constructed from generalized splines.

They develop force field techniques for fitting their model to monocular, binocular, and dynamic image data. The distributed nature of this deformable model enhances its descriptive power and allows the representation of natural objects with asymmetries and fine detail. However, the generalized spline components of the model do not explicitly provide an abstract representation of object shape in terms of a few parameters.

The generalized cylinder representation requires the specification of an axis, generally a space curve, and the cross-section function. Pentland [11], [12] proposes the use of a simpler part model with scalar parameters: the superquadric ellipsoid with parameterized deformations [1] (the superquadrics were discovered by Hein; see [6]). Pentland's proposal has spawned a flurry of efforts to reconstruct superquadric ellipsoids with global geometric deformations from 3D data, and these have met with some success [7], [8], [17].

Pentland [13], following the physically based approach of [22], proposes an alternative method for fitting deformable part models based on superquadric ellipsoids. Inspired by modal analysis (a technique for analyzing the vibrations of linear mechanical systems under periodic forcing conditions [2]), he applies polynomial deformation "modes" to superquadrics. Pentland's modeling primitives are not fully dynamic in that the underlying superquadric parameters do not respond to forces and are not fitted to data through force interactions. The deformation modes may make the method efficient for the recovery of smooth, symmetrically deformed parts. On the down side, global deformation modes lack an obvious physical meaning, and they make it difficult to deal with nonlinearities and boundary conditions. Moreover, the representation of complex shapes requires many modes, rendering Pentland's scheme no more efficient than a nodal finite element solution [2].

The present paper develops a new family of primitives with fully dynamic global and local deformations. Our formulation is similar to that of Terzopoulos and Witkin [21] with regard to the dynamics of free rigid-body motions and local deformations, but it includes additional global deformational degrees of freedom that may be inherited from any parameterized family of geometric primitives (in this paper, superquadrics). Our treatment of global deformation dynamics shares similar features with Witkin and Welch's [23] formulation of linearly deformable primitives, but we must deal with the nonlinear deformations of superquadrics. Our treatment of local deformation dynamics is general in that it permits the use of local or global support basis functions (it reduces to modal analysis [2] if we choose a sinusoidal eigenfunction basis). In this paper, we employ local finite element basis functions since they provide greater shape flexibility and are better suited to the purpose of local deformations, i.e., the representation of local detail.

III. Formulation of Deformable Superquadrics

In this section, we provide a general formulation of deformable superquadrics. To arrive at the equations of motion that govern these models, we extend some results from [16] and [21].

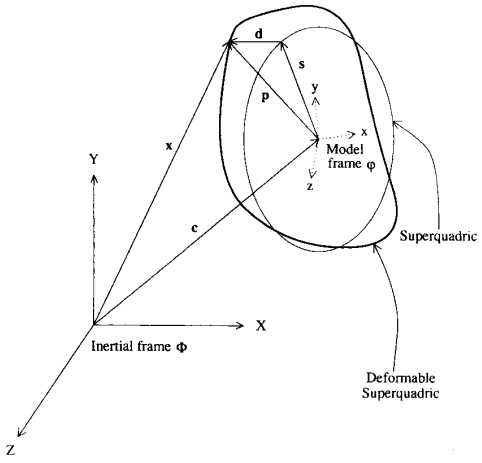


Fig. 2. Geometry of deformable superquadric.

A. Geometry

Geometrically, the models developed in this paper are closed surfaces in space whose intrinsic (material) coordinates are $\mathrm{u} = (u, v)$, defined on 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 of u:

$$\boldsymbol{x}(\mathrm{u}, t) = (x_1(\mathrm{u}, t), x_2(\mathrm{u}, t), x_3(\mathrm{u}, t))^T \tag{1}$$

where ^{T} is the transpose operator. We set up a noninertial, model-centered reference frame ϕ and express these positions as

$$\boldsymbol{x} = \boldsymbol{c} + \boldsymbol{R}\boldsymbol{p} \tag{2}$$

where $\boldsymbol{c}(t)$ is the origin of ϕ at the center of the model, and the orientation of ϕ is given by the rotation matrix $\boldsymbol{R}(t)$. Thus, $\boldsymbol{p}(\mathrm{u}, t)$ denotes the canonical positions of points on the model relative to the model frame. We further express $\boldsymbol{p}$ as the sum of a reference shape $\boldsymbol{s}(\mathrm{u}, t)$ and a displacement function $\boldsymbol{d}(\mathrm{u}, t)$:

$$\boldsymbol{p} = \boldsymbol{s} + \boldsymbol{d}. \tag{3}$$

Fig. 2 illustrates the model geometry.

The ensuing formulation can be carried out for any reference shape given as a parameterized function of u. For concreteness, however, we consider the case of superquadric ellipsoids [1], which yield the reference shape

$$\boldsymbol{s} = a \begin{pmatrix} a_1 C_u^{\epsilon_1} C_v^{\epsilon_2} \\ a_2 C_u^{\epsilon_1} S_v^{\epsilon_2} \\ a_3 S_u^{\epsilon_1} \end{pmatrix} \tag{4}$$

where $-\pi/2 \leq u \leq \pi/2$ and $-\pi \leq v < \pi$, and where $S_w^{\epsilon} = \mathrm{sgn}(\sin w)|\sin w|^{\epsilon}$ and $C_w^{\epsilon} = \mathrm{sgn}(\cos w)|\cos w|^{\epsilon}$, respectively. Here, $a \geq 0$ is a scale parameter, $0 \leq a_1, a_2, a_3 \leq 1$ are aspect ratio parameters, and $\epsilon_1, \epsilon_2 \geq 0$ are "squareness" parameters. We collect the superquadric parameters into the parameter vector

$$\boldsymbol{q}_s = (a, a_1, a_2, a_3, \epsilon_1, \epsilon_2)^T. \tag{5}$$

In general, we can express the displacement $\boldsymbol{d}$ as a linear combination of basis functions $\boldsymbol{b}_i(u)$

$$\boldsymbol{d} = \sum_i \mathrm{diag}(\boldsymbol{b}_i)\boldsymbol{q}_i \tag{6}$$

where $\mathrm{diag}(\boldsymbol{b}_i)$ is a diagonal matrix formed from the basis functions and where $\boldsymbol{q}_i$ depend only on time and are known as degrees of freedom or generalized coordinates. The basis functions must be admissible, i.e., they must satisfy the kinematic boundary conditions of the model. We will give specific basis functions shortly; however, if we approximate the displacement field using a finite number of basis functions and collect the generalized coordinates into a vector of degrees of freedom $\boldsymbol{q}_d = (\cdots, \boldsymbol{q}_i, \cdots)^T$, we can write

$$\boldsymbol{d} = \boldsymbol{S}\boldsymbol{q}_d \tag{7}$$

where $\boldsymbol{S}$ is the shape matrix whose entries are the basis functions.

B. Kinematics

The velocity of points on the model is given by

$$\begin{aligned}\dot{\boldsymbol{x}} &= \dot{\boldsymbol{c}} + \dot{\boldsymbol{R}}\boldsymbol{p} + \boldsymbol{R}\dot{\boldsymbol{p}} \\ &= \dot{\boldsymbol{c}} + \boldsymbol{B}\dot{\boldsymbol{\theta}} + \boldsymbol{R}\dot{\boldsymbol{s}} + \boldsymbol{R}\boldsymbol{S}\dot{\boldsymbol{q}}_d\end{aligned} \tag{8}$$

where $\boldsymbol{\theta} = (\cdots, \theta_i, \cdots)^T$ is the vector of rotational coordinates of the model, and $\boldsymbol{B} = [\cdots \partial(\boldsymbol{R}\boldsymbol{p})/\partial\theta_i \cdots]$. Furthermore

$$\dot{\boldsymbol{s}} = \left[\frac{\partial \boldsymbol{s}}{\partial \boldsymbol{q}_s}\right]\dot{\boldsymbol{q}}_s = \boldsymbol{J}\dot{\boldsymbol{q}}_s \tag{9}$$

where $J$is the Jacobian of the superquadric ellipsoid function (see (10) at the bottome of the page). We can therefore write

$$\dot{\boldsymbol{x}} = [\boldsymbol{I}\ \boldsymbol{B}\ \boldsymbol{R}\boldsymbol{J}\ \boldsymbol{R}\boldsymbol{S}]\dot{\boldsymbol{q}} = \boldsymbol{L}\dot{\boldsymbol{q}} \tag{11}$$

where $\boldsymbol{q} = \left(\boldsymbol{q}_c^T, \boldsymbol{q}_\theta^T, \boldsymbol{q}_s^T, \boldsymbol{q}_d^T\right)^T$, with $\boldsymbol{q}_c = \boldsymbol{c}$ and $\boldsymbol{q}_\theta = \boldsymbol{\theta}$.

C. Dynamics

When fitting the model to visual data, our goal is to recover $\boldsymbol{q}$, which is the vector of degrees of freedom of the model. The components $\boldsymbol{q}_c$ and $\boldsymbol{q}_\theta$ are the global rigid motion coordinates, $\boldsymbol{q}_s$ are the global deformation coordinates, and $\boldsymbol{q}_d$ are the local deformation coordinates of the model. Our approach carries out the coordinate fitting procedure in a physically based way. We make our model dynamic in $\boldsymbol{q}$ by introducing mass, damping, and a deformation strain energy. This allows us, through the apparatus of Lagrangian dynamics, to arrive at a set of equations of motion governing the behavior of our model under the action of externally applied forces.

The derivation in Appendix A shows that the equations of motion take the form

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{g}_q + \boldsymbol{f}_q \tag{12}$$

where $\boldsymbol{M}$, $\boldsymbol{C}$, and $\boldsymbol{K}$ are the mass, damping, and stiffness matrices, respectively, where $\boldsymbol{g}_q$ are inertial (centrifugal and Coriolis) forces arising from the dynamic coupling between the local and global degrees of freedom, and where $\boldsymbol{f}_q(\mathrm{u}, t)$ are the generalized external forces associated with the degrees of freedom of the model. The appendix provides formulas for the above matrices and vectors.

D. Elasticity

The stiffness matrix $\boldsymbol{K}$ given by (30) determines the elastic properties of the model. We derive $\boldsymbol{K}$ from a deformation energy.

For the applications in this paper, we want the global deformation parameters $\boldsymbol{q}_s$ that stem from the superquadric to freely account for as much of the data as possible. Consequently, we impose no deformation energy on $\boldsymbol{q}_s$, i.e., we set $\boldsymbol{K}_{ss} = \boldsymbol{K}_{sd} = 0$ in (30). The local deformation parameters $\boldsymbol{q}_d$, however, must be constrained to yield a small and continuous deformation function.

We impose on $\boldsymbol{q}_d$ a special case of the generalized spline deformation energy proposed in [21], which is a membrane energy given in continuous form by

$$\mathcal{E}(\boldsymbol{d}) = \int w_1 \left(\left(\frac{\partial \boldsymbol{d}}{\partial u}\right)^2 + \left(\frac{\partial \boldsymbol{d}}{\partial v}\right)^2\right) + w_0 \boldsymbol{d}^2 d\mathrm{u} \tag{13}$$

where the function $w_0(\mathrm{u})$ controls the local magnitude, and $w_1(\mathrm{u})$ controls the local variation of the deformation. In our implementation, we reduce these functions to scalar stiffness parameters w_0 and w_1.

IV. Simplified Numerical Simulation

Equations (12) along with the expressions in Appendix A give the general equations of motion for a dynamic model with local and global deformations. A full implementation and simulation of the general equations would be appropriate for physically based animation where realistic motion is important [21]. However, in computer vision and geometric design applications involving the fitting of models to data, models governed by simplified equations of motion suffice, as the experiments in Section VI will demonstrate.

We can simplify the equations while preserving useful dynamics by setting the mass density $\mu(\mathrm{u})$ (see (24)) to zero to obtain

$$\boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{f}_q. \tag{14}$$

$$\boldsymbol{J} = \begin{pmatrix} a_1 C_u^{\epsilon_1} C_v^{\epsilon_2} & a C_u^{\epsilon_1} C_v^{\epsilon_2} & 0 & 0 & a a_1 \epsilon_1 C_u^{\epsilon_1 - 1} C_v^{\epsilon_2} & a a_1 \epsilon_2 C_u^{\epsilon_1} C_v^{\epsilon_2 - 1} \\ a_2 C_u^{\epsilon_1} S_v^{\epsilon_2} & 0 & a C_u^{\epsilon_1} S_v^{\epsilon_2} & 0 & a a_2 \epsilon_1 C_u^{\epsilon_1 - 1} S_v^{\epsilon_2} & a a_2 \epsilon_2 C_u^{\epsilon_1} S_v^{\epsilon_2 - 1} \\ a_3 S_u^{\epsilon_1} & 0 & 0 & a S_u^{\epsilon_1} & a a_3 \epsilon_1 S_u^{\epsilon_1 - 1} & 0 \end{pmatrix} \tag{10}$$

These equations yield a model that has no inertia and comes to rest as soon as all the applied forces vanish or equilibrate.

Equation (14) is discretized in material coordinates u using nodal finite element basis functions. We carry out the discretization by tessellating the surface of the model into bilinear quadrilateral elements, except at the polar caps, where we use linear triangular elements (see Appendix B). The local (nodal) basis functions associated with these elements lead to a sparse $\boldsymbol{K}_{dd}$ [2].

The formulation of our model yields numerically stable equations of motion that may be integrated forward through time using explicit procedures. For fast interactive response, we employ a first-order Euler method to integrate (14). The Euler procedure updates the degrees of freedom $\boldsymbol{q}$ of the model at time $t + \Delta t$ according to the formula

$$\boldsymbol{q}^{(t+\Delta t)} = \boldsymbol{q}^{(t)} + \Delta t \Big(\boldsymbol{C}^{(t)}\Big)^{-1} \Big(\boldsymbol{f}_q^{(t)} - \boldsymbol{K}\boldsymbol{q}^{(t)}\Big) \tag{15}$$

where Δt is the time step size. Note that we need never assemble the large $\boldsymbol{K}_{dd}$ submatrix of $\boldsymbol{K}$. Instead, we compute $\Sigma_j (k_{dd})_{ij} \boldsymbol{q}_{dj}$ for each node i in an "element-by-element" fashion.

Taking time steps in $\boldsymbol{q}$ is straightforward, but the rotation component $\boldsymbol{q}_\theta$ is a little delicate. We present $\boldsymbol{q}_\theta$ using quaternions. Updating quaternions at each time step is easier than directly updating a rotation matrix and ensuring that it remains orthogonal. Quaternions also avoid problems with "gimbal lock" that may arise when Euler angles are used to represent rotations.

A quaternion $[s, \boldsymbol{v}]$ with unit magnitude $\| [s, \boldsymbol{v}] \| = s^2 + \boldsymbol{v}^T\boldsymbol{v} = 1$ specifies a rotation of the model from its reference position through an angle $\theta = 2\cos^{-1} s$ around an axis aligned with vector $\boldsymbol{v} = [v_1, v_2, v_3]^T$. The rotation matrix corresponding to $[s, \boldsymbol{v}]$ is

$$\boldsymbol{R} = \begin{bmatrix} 1 - 2\big(v_2^2 + v_3^2\big) & 2(v_1v_2 - sv_3) & 2(v_1v_3 + sv_2) \\ 2(v_1v_2 + sv_3) & 1 - 2\big(v_1^2 + v_3^2\big) & 2(v_2v_3 - sv_1) \\ 2(v_1v_3 - sv_2) & 2(v_2v_3 + sv_1) & 1 - 2\big(v_1^2 + v_2^2\big) \end{bmatrix}. \tag{16}$$

To obtain $\boldsymbol{q}_\theta$ from (15), we use the generalized torque at time t given by $\boldsymbol{f}_\theta^T = \int \boldsymbol{f}^T \boldsymbol{B} d\mathrm{u}$ (see (33)), with $\boldsymbol{B}$

$$\boldsymbol{B}(\mathrm{u}) = -\boldsymbol{R}\tilde{\boldsymbol{p}}(\mathrm{u})\,\boldsymbol{G} \tag{17}$$

[16], where $\boldsymbol{R}$ represents the rotation matrix at time t, where $\tilde{\boldsymbol{p}}(\mathrm{u})$ is the dual 3×3 matrix of the position vector $\boldsymbol{p}(\mathrm{u}) = (p_1, p_2, p_3)^T$ (see (3)) defined as

$$\tilde{\boldsymbol{p}}(\mathrm{u}) = \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix} \tag{18}$$

and where $\boldsymbol{G}$ is a 3×4 matrix whose definition is based on the value of the quaternion $\boldsymbol{q}_\theta = [s, \boldsymbol{v}]$ representing the rotation at time t:

$$\boldsymbol{G} = 2 \begin{bmatrix} -v_1 & s & v_3 & -v_2 \\ -v_2 & -v_3 & s & v_1 \\ -v_3 & v_2 & -v_1 & s \end{bmatrix}. \tag{19}$$

V. Applied Forces

In the dynamic model fitting process, the data are transformed to an externally applied force distribution $\boldsymbol{f}(\mathrm{u}, t)$. Using (33), we convert the external forces to generalized forces $\boldsymbol{f}_q$, which act on the degrees of freedom of the model. We employ two types of forces based on the structure of the input data—short-range forces obtained through gradients of potential functions and long-range forces based on distances between data points and the model's surface.

Techniques for generating suitable potential functions from monocular, binocular, and dynamic image sequences are described in [22]. For example, to attract a 3D model towards significant intensity gradients in a continuous image $I(x, y)$, we construct the potential function

$$P(x, y) = \| \nabla (G_\sigma * I) \| \tag{20}$$

where G_σ denotes a Gaussian smoothing filter of characteristic width σ, which determines the extent of the region of attraction of the intensity gradient. Typically, the attraction has a relatively short range. The potential function applies a force

$$\boldsymbol{f} = \beta \nabla P(\Pi \boldsymbol{x}) \tag{21}$$

to the model, where β controls the strength of the force, and Π is a suitable projection of points on the model into the image plane.

To compute the potential function in practice, we begin with a digital image $I(i, j)$, convolve it with a discrete filter G_σ, and compute at each pixel (i, j) the magnitude of the discrete gradient operator calculated from central finite differences of neighboring pixel values. To evaluate (21) at the location of a projected model point $\Pi \boldsymbol{x} = (x, y)$, we first calculate using central finite differences the discrete gradients ∇P_k at the four pixels $k = 1, \cdots, 4$ that surround (x, y). We then consider these pixels as the nodes of the quadrilateral finite element of Fig. 10, with $a = b = 1$, in order to define a bilinear interpolant in the region between the pixels, i.e., using (48) and (49), the interpolant is given by $\nabla P(x, y) = \Sigma_{k=1}^4 N_k(2(x - x_c), 2(x - y_c)) \nabla P_k$, where (x_c, y_c) denotes the centroid of the four pixels.

Alternatively, we may define long-range forces

$$\boldsymbol{f}(\mathrm{u}_r) = \beta \, \| \boldsymbol{r} - \boldsymbol{x}(\mathrm{u}_r) \| \tag{22}$$

based on the separation between a datapoint $\boldsymbol{r}$ in space and the force's point of influence u_r on the model's surface. In general, $\mathrm{u}_r = (u_r, v_r)$ will fall somewhere within an element on the surface of the model. We can compute $\boldsymbol{x}(\mathrm{u}_r)$ in the domain of a quadrilateral element, for instance, according to its bilinear local interpolant $\boldsymbol{x}(\mathrm{u}_r) = \Sigma_{i=1}^4 N_i(2(u_r - u_c)/a, 2(v_r - v_c)/b)\boldsymbol{x}_i$, where the $\boldsymbol{x}_i$ are the nodal positions (see (48) and (49)). The equivalent forces on each of the four nodes is $\boldsymbol{f}_i = N_i(2(u_r - u_c)/a, 2(v_r - v_c)/b)\boldsymbol{f}(\mathrm{u}_r)$. When u_r falls within the domain of a triangular element at the polar caps of the model, the computations proceed in an analogous fashion using the corresponding formulas given in Appendix B-2.

Usually, we want u_r to minimize the distance $d(\mathrm{u}) = \|\boldsymbol{r} - \boldsymbol{x}(\mathrm{u})\|$. A closed-form analytic formula for u_r is unavailable for a discrete deformable superquadric. A brute-force

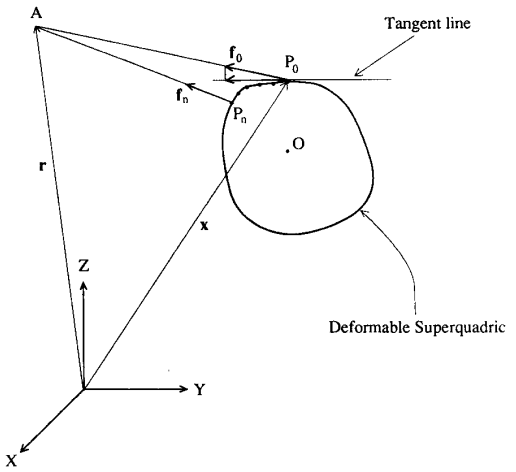


Fig. 3. Migration of data force point of influence over model surface.

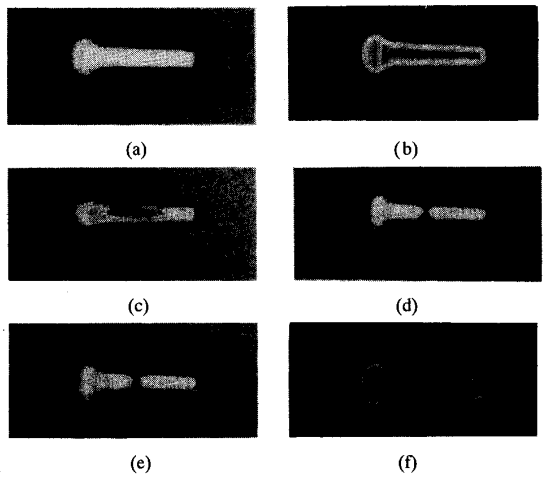


Fig. 4. Fitting deformable superquadric to pestle image (see text for (a)–(f)).

approach that works well in our experiments is to place u_r at a node of the model, which minimizes d. The complexity of this operation is high—$O(mn)$, where m is the number of datapoints and n is the number of nodes—but it need not be carried out at every time step. A more efficient approach (not used in the experiments) is to use a nonlinear optimization procedure. With a good starting point (from the previous time step), conjugate gradient or quasi-Newton methods will converge to a solution in a few steps, making such an approach linear in m. The minimization method must be applied within element domains using the nodal shape functions as an analytic representation of the model's surface. The bookkeeping in the minimization is complicated by the need to optimize across element boundaries.

We have experimented with an alternative approach that involves a dynamic procedure for migrating points of influence over the model's surface until they arrive at locations of minimal distance from the given datapoints. Starting from initial points of influence not necessarily at minimal distance (Fig. 3), we project the force at each time step onto the unit tangent vectors $(\partial \boldsymbol{x}/\partial u)/\|\partial \boldsymbol{x}/\partial u\|$ and $(\partial \boldsymbol{x}/\partial v)/\|\partial \boldsymbol{x}/\partial v\|$ to the surface at the current point of influence $P_0 = u_0$, and we migrate the point in the u plane by taking an Euler step in u and v proportional to the magnitude of the respective projections. Thus, the point of influence migrates to a point P_n of minimal distance, where the tangential components of the force vanish. The scheme works well, unless the surface is highly convoluted.

VI. Experiments

We have evaluated our approach in simulations involving image and range data. Our experiments run at interactive rates on a Silicon Graphics Personal Iris 4D-25TG workstation including the real-time graphics.

A. 2D Image Data

Fig. 4 shows the various steps of fitting a deformable superquadric to a 120 x 128, 256-intensity monocular image Fig. 4(a) of a 3D object—a pestle. The size of the image is rescaled to fit within the unit square on the $x - y$ plane. Fig. 4(b) shows the potential function $P(x, y)$ generated from the image by computing the magnitude of the intensity gradient.

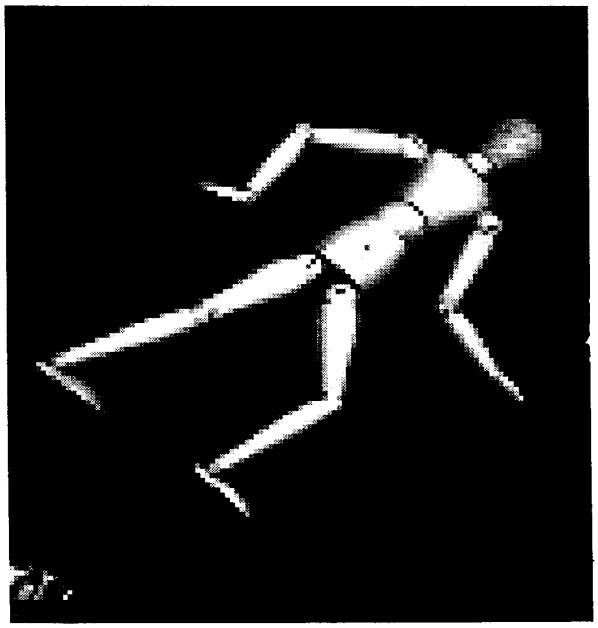
Fig. 4(c) shows the initial state of the deformable superquadric displayed in wireframe projected onto the image. The surface of the model is discretized into 5043 nodes. The initialization consists of specifying the center of the model $\boldsymbol{c}$, along with the major and minor axes, $a \cdot a_1$ and $a \cdot a_2$, by picking four points with the mouse. This initializes the translation $\boldsymbol{q}_c$ and rotation $\boldsymbol{q}_\theta$ of the model. We also fix $\epsilon_1 = \epsilon_2 = 1.0$. In this and subsequent experiments, the local deformation $\boldsymbol{q}_d$ is initially set to zero. Note that the initialization step produces a very crude first approximation to the pestle.

Fig. 4(d) shows an intermediate step in the fitting process that simulates the equations of motion using stiffness parameters $w_0 = 1.0 \times 10^{-6}$ and $w_1 = 4.0 \times 10^{-2}$. Using an orthogonal projection Π, nodes of the model whose positions $\boldsymbol{x}$ in space lie near the image plane ($|x_3| < 0.2$) are subject to a force directed parallel to the image plane:

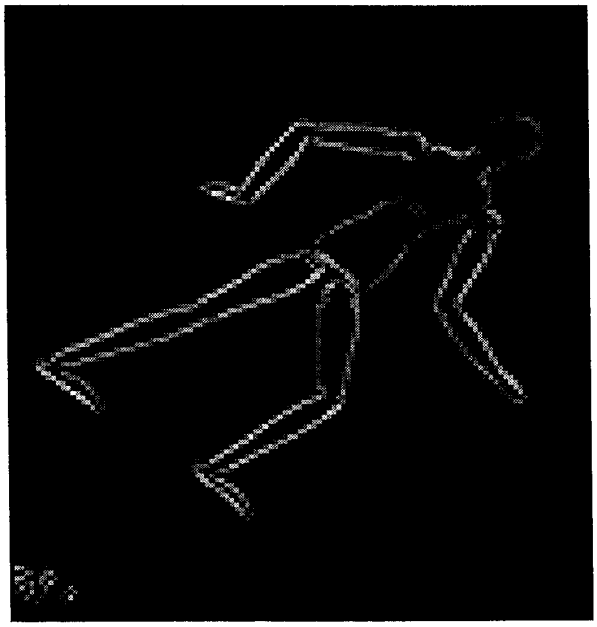
$$\boldsymbol{f} = \beta\left(\frac{\partial P}{\partial x}, \frac{\partial P}{\partial y}, 0\right)^T \tag{23}$$

where the force strength factor is $\beta = 4.0 \times 10^{-6}$. The forces deform the model, and Figs. 4(e) and (f) show the final state of the model at equilibrium, superimposed on the image and the potential, respectively.

In the second experiment, we use the image of a doll shown in Fig. 5(a), whose potential is shown in Fig. 5(b). The specifics of this experiment are identical to those of the previous one, except that the discrete models consisted of 963 nodes, and their stiffness parameters were $w_0 = 0.001$ and $w_1 = 0.1$. Fig. 6(a) illustrates the results of the initialization phase for the doll image, which was carried out as described above, showing 11 crude approximations to the major body parts of the figure. The image forces deform the part models into the final shapes shown in Fig. 6(b).



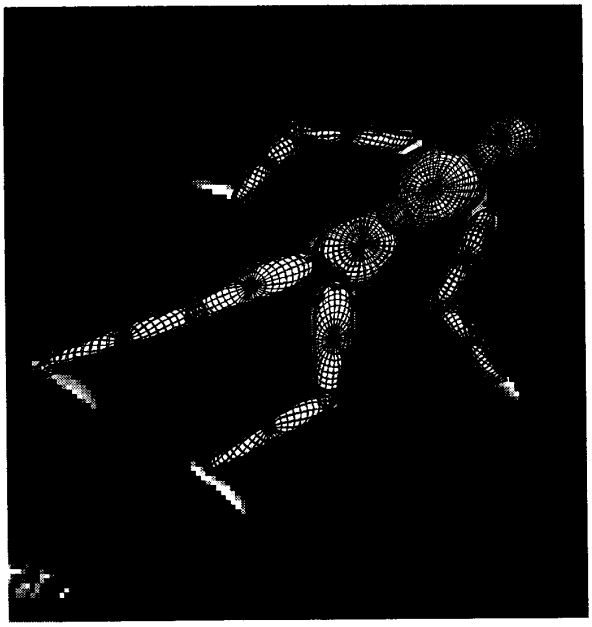
(a)



(b)

Fig. 5. (a) Doll image and (b) potential.

(a)



(b)

Fig. 6. Fitting deformable superquadrics to doll image (see text for (a)–(b)).

The above simulations did not require user intervention beyond the initialization phase. It is important to realize that initialization requires only a rough segmentation of the image into blobs corresponding to parts. We therefore expect that the initialization can be automated using available image segmentation techniques. The reaction-diffusion space segmentation process of Kimia *et al.* [9] appears promising for this purpose.

B. 3D Range Data

3D data generally provide greater constraint in the model fitting process than do 2D image projections. The following experiments utilize range data from the NRCC 3D image database [15]. We have experimented with the two force techniques for fitting the model to 3D data described in the

previous section. In the following simulations, we have applied forces to the model using the brute-force nearest-node search method, updating the nodes of attachment for each datapoint every 200 iterations.

In the third experiment, we fit a deformable superquadric model with 2603 nodes to 3D data sparsely sampled from the upper "hemisphere" of an egg (from range map EGG 1 CAT # 233). Fig. 7(a) shows the 499 range datapoints. The stiffness parameters of the model were $w_0 = 1.0 \times 10^{-6}$ and $w_1 = 0.1$. We initialized the model to a sphere located at the center of gravity of the data ($a = 1.2, a_1 = a_2 = a_3 = 0.5$, $\epsilon_1 = \epsilon_2 = 1.0$). Fig. 7(b) shows the fitted deformable superquadric at equilibrium, and Fig. 7(c) shows a top view of the fitted model. Evidently, the fit is accurate over the portions of the surface covered by datapoints, but it begins to deteriorate at the boundary of the data near the "equator" because of the influence of the underside of the model, which remains too spherical due to the lack of datapoints.

In the fourth experiment, we fit a model with 1683 nodes to 3D data sparsely sampled from the upper part of a mug with a pitted surface (from range map MUG 1 CAT # 251). Fig. 8(a) shows the 651 range datapoints. The stiffness parameters of the model were $w_0 = 0.01$ and $w_1 = 0.1$. We initialized the model to a "tubular" shape ($a = 1.5, a_1 = a_2 = 0.3$, $a_3 = 0.8, \epsilon_1 = 0.7, \epsilon_2 = 1.0$). Fig. 8(b) shows the fitted deformable superquadric at equilibrium. The underside of the model is smooth due to the lack of data, but the pitted texture of the top surface has been accurately reconstructed by the local deformational degrees of freedom of the deformable superquadric.

VII. Concluding Remarks

We have developed deformable superquadrics, which are dynamic models with global and local deformation properties inherited from superquadric ellipsoids and membrane splines. These physically based models are governed by equations of motion. We are able to simulate and render simplified versions of the equations at interactive rates on a graphics workstation. The dynamic equations make deformable superquadrics responsive to forces derived from image or range data, which compels the model to conform to the data. The model is useful for reconstructing 3D objects or parts of objects with irregular, local shape features from such data. It also promises to be useful for abstracting global shape features of objects for the purposes of recognition.

In addition to specifying the stiffness parameters (w_0 and w_1) of the model along with a few parameters in the numerical simulation procedures, we must currently provide reasonable initial estimates for the translation, rotation, and global deformation variables of our model, especially when fitting them to 2D image data. Initial values may be estimated by rough segmentation and calculation of central moments of the data [17].

We are currently implementing the full, second-order equations of motion (12) and extending our formulation to accommodate other global deformations such as bends, shears, and tapers. Additional deformations of this sort may provide useful degrees of freedom to deformable superquadrics or to related

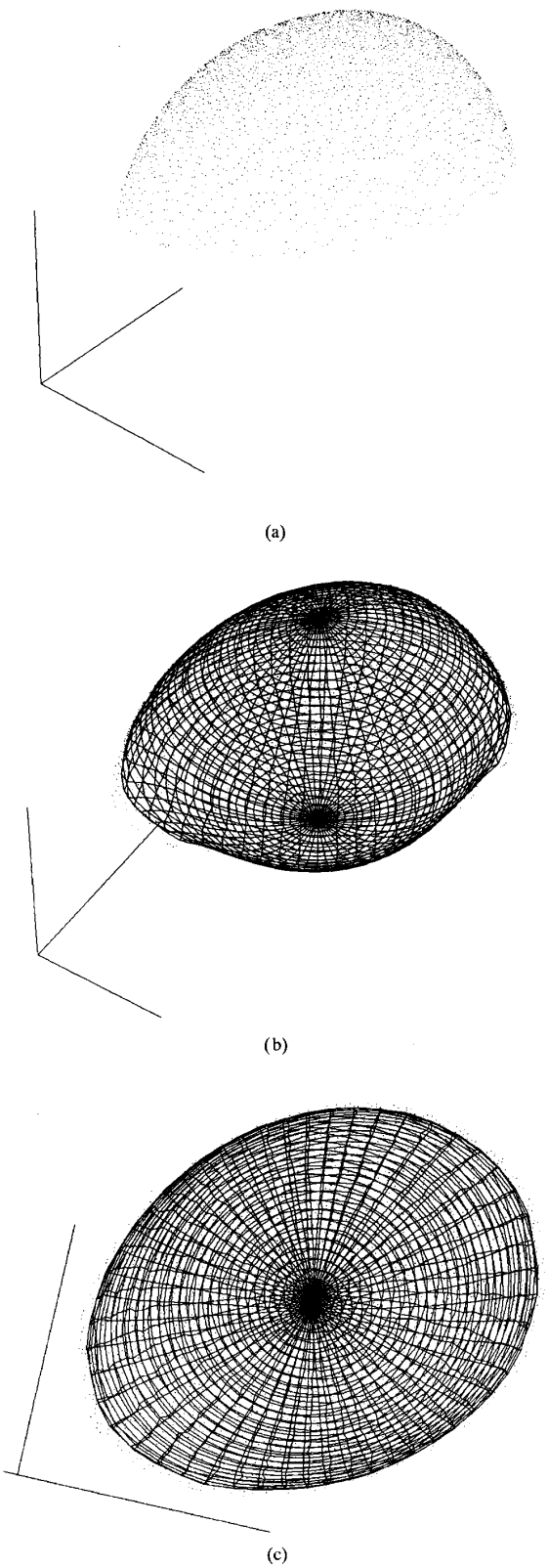


Fig. 7. Fitting deformable superquadric ((b),(c)) to egg range data (d).

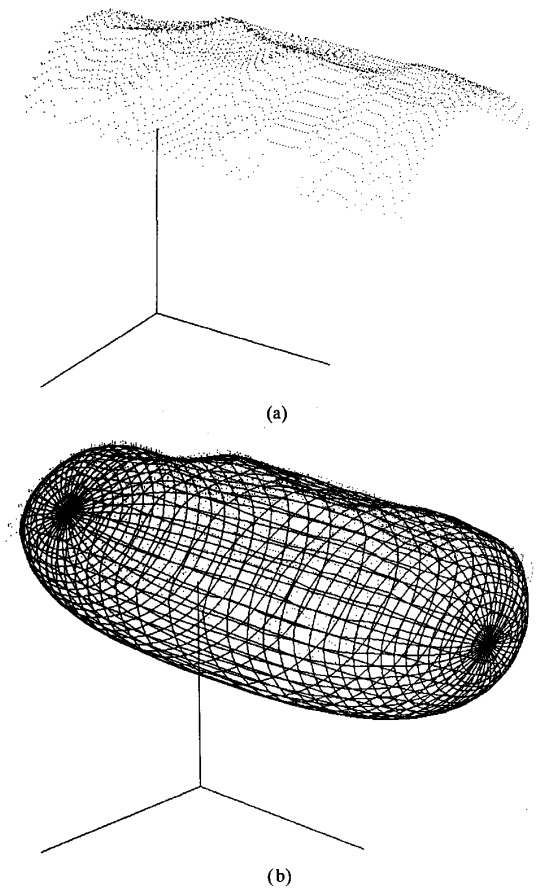


Fig. 8. Fitting deformable superquadric (b) to mug range data (a).

models based on other parameterized volumetric primitives. Additionally, the formulation of constraints among dynamic models with local and global deformations is an important topic currently under investigation.

Appendix A
Derivation of the Equations of Motion

1. Kinetic Energy: Mass Matrix

The kinetic energy of the model is given by

$$\mathcal{T} = \frac{1}{2}\int \mu \dot{\boldsymbol{x}}^T \dot{\boldsymbol{x}}\, \mathrm{d}u = \frac{1}{2}\dot{\boldsymbol{q}}^T \left[\int \mu \boldsymbol{L}^T \boldsymbol{L}\, \mathrm{d}u\right] \dot{\boldsymbol{q}} = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M} \dot{\boldsymbol{q}} \tag{24}$$

where $\boldsymbol{M} = \int \mu \boldsymbol{L}^T \boldsymbol{L}\, \mathrm{d}u$ is the symmetric mass matrix of the object, and $\mu(u)$ is the mass density of the object. Using the expression for $\boldsymbol{L}$ from (11), we can rewrite $\boldsymbol{M}$ as follows:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_{cc} & \boldsymbol{M}_{c\theta} & \boldsymbol{M}_{cs} & \boldsymbol{M}_{cd} \\ & \boldsymbol{M}_{\theta\theta} & \boldsymbol{M}_{\theta s} & \boldsymbol{M}_{\theta d} \\ & & \boldsymbol{M}_{ss} & \boldsymbol{M}_{sd} \\ \text{symmetric} & & & \boldsymbol{M}_{dd} \end{bmatrix} \tag{25}$$

where

$$\begin{aligned} \boldsymbol{M}_{cc} &= \int \mu \boldsymbol{I}\, \mathrm{d}u & \boldsymbol{M}_{\theta s} &= \int \mu \boldsymbol{B}^T \boldsymbol{R} \boldsymbol{J}\, \mathrm{d}u \\ \boldsymbol{M}_{c\theta} &= \int \mu \boldsymbol{B}\, \mathrm{d}u & \boldsymbol{M}_{\theta d} &= \int \mu \boldsymbol{B}^T \boldsymbol{R} \boldsymbol{S}\, \mathrm{d}u \\ \boldsymbol{M}_{cs} &= \boldsymbol{R}\int \mu \boldsymbol{J}\, \mathrm{d}u & \boldsymbol{M}_{ss} &= \int \mu \boldsymbol{J}^T \boldsymbol{J}\, \mathrm{d}u \\ \boldsymbol{M}_{cd} &= \boldsymbol{R}\int \mu \boldsymbol{S}\, \mathrm{d}u & \boldsymbol{M}_{sd} &= \int \mu \boldsymbol{J}^T \boldsymbol{S}\, \mathrm{d}u \\ \boldsymbol{M}_{\theta\theta} &= \int \boldsymbol{B}^T \boldsymbol{B}\, \mathrm{d}u & \boldsymbol{M}_{dd} &= \int \mu \boldsymbol{S}^T \boldsymbol{S}\, \mathrm{d}u. \end{aligned} \tag{26}$$

2. Energy Dissipation: Damping Matrix

We assume velocity dependent kinetic energy dissipation, which can be expressed in terms of the (Raleigh) dissipation functional:

$$\mathcal{F} = \frac{1}{2}\int \gamma \dot{\boldsymbol{x}}^T \dot{\boldsymbol{x}}\, \mathrm{d}u \tag{27}$$

where $\gamma(u)$ is a damping density. Since it has the same form as (24), we can rewrite (27) as follows:

$$\mathcal{F} = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{C} \dot{\boldsymbol{q}} \tag{28}$$

where the damping matrix $\boldsymbol{C}$ has the same form as $\boldsymbol{M}$, except that γ replaces μ.

3. Strain Energy: Stiffness Matrix

We define the deformation characteristics of the model in terms of a deformation strain energy. We impose a strain energy of the general form

$$\mathcal{E}(x) = \frac{1}{2}\boldsymbol{q}^T \boldsymbol{K} \boldsymbol{q} \tag{29}$$

where

$$\boldsymbol{K} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 \\ & & \boldsymbol{K}_{ss} & \boldsymbol{K}_{sd} \\ \text{symmetric} & & & \boldsymbol{K}_{dd} \end{pmatrix} \tag{30}$$

is the stiffness matrix. The zero submatrices indicate that only the global $\boldsymbol{q}_s$ and local $\boldsymbol{q}_d$ deformational degrees of freedom can contribute to the strain energy.

4. External Forces and Virtual Work

The external forces $\boldsymbol{f}(\mathrm{u}, t)$ applied to the model do virtual work, which can be written as

$$\delta \boldsymbol{W}_F = \int \boldsymbol{f}^T \boldsymbol{L} \delta \boldsymbol{q}\, \mathrm{d}u = \int \boldsymbol{f}_q \delta \boldsymbol{q}\, \mathrm{d}u \tag{31}$$

where

$$\boldsymbol{f}_q = \boldsymbol{f}^T \boldsymbol{L} = (\boldsymbol{f}_c, \boldsymbol{f}_\theta, \boldsymbol{f}_s, \boldsymbol{f}_d) \tag{32}$$

with

$$\begin{aligned} \boldsymbol{f}_c^T &= \int \boldsymbol{f}^T\, \mathrm{d}u, & \boldsymbol{f}_s^T &= \int \boldsymbol{f}^T \boldsymbol{R} \boldsymbol{J}\, \mathrm{d}u \\ \boldsymbol{f}_\theta^T &= \int \boldsymbol{f}^T \boldsymbol{B}\, \mathrm{d}u, & \boldsymbol{f}_d^T &= \int \boldsymbol{f}^T \boldsymbol{R} \boldsymbol{S}\, \mathrm{d}u \end{aligned} \tag{33}$$

is the vector of generalized external forces associated with the degrees of freedom of the model.

5. Lagrange Equations of Motion

The Lagrange equations of motion for the model take the form

$$\frac{d}{dt}\left(\frac{\partial \mathcal{T}}{\partial \dot{\boldsymbol{q}}}\right)^T - \left(\frac{\partial \mathcal{T}}{\partial \boldsymbol{q}}\right)^T + \left(\frac{\partial \mathcal{F}}{\partial \dot{\boldsymbol{q}}}\right)^T + \delta_x \mathcal{E} = \boldsymbol{f}_q. \quad (34)$$

The first two terms of (34) express inertial forces and can be written as

$$\frac{d}{dt}\left(\frac{\partial \mathcal{T}}{\partial \dot{\boldsymbol{q}}}\right)^T - \left(\frac{\partial \mathcal{T}}{\partial \boldsymbol{q}}\right)^T = \boldsymbol{M}\ddot{\boldsymbol{q}} - \boldsymbol{g}_q \quad (35)$$

where

$$\boldsymbol{g}_q = -\dot{\boldsymbol{M}}\dot{\boldsymbol{q}} + \frac{1}{2}\left[\frac{\partial}{\partial \boldsymbol{q}}\left(\dot{\boldsymbol{q}}^T \boldsymbol{M}\dot{\boldsymbol{q}}\right)\right]^T \quad (36)$$

gives the centrifugal and Coriolis forces [16]. The third term expresses the friction forces and takes the form

$$\frac{\partial \mathcal{F}}{\partial \dot{\boldsymbol{q}}} = \boldsymbol{C}\dot{\boldsymbol{q}}. \quad (37)$$

The fourth term, the variational derivative of $\mathcal{E}$ with respect to $\boldsymbol{x}$, expresses the elastic forces

$$\delta_{\boldsymbol{x}} \mathcal{E} = \boldsymbol{K}\boldsymbol{q}. \quad (38)$$

Substituting the above expressions into (34) yields the equations of motion (12).

Appendix B
Derivation of K_{dd}

We discretize the model in material coordinates u using finite elements. We can derive $\boldsymbol{K}_{dd}$ as an assembly of the local stiffness matrices $\boldsymbol{K}_{dd}^j$ associated with each element domain $E_j \subset \mathrm{u}$. Since $\boldsymbol{d}(\mathrm{u}, t) = \left[d_1(\mathrm{u}, t), d_2(\mathrm{u}, t), d_3(\mathrm{u}, t)\right]^T$, we can rewrite the membrane spline deformation energy (13) on E_j as the sum of component energies

$$\mathcal{E}^j(\boldsymbol{d}) = \mathcal{E}^j(d_1) + \mathcal{E}^j(d_2) + \mathcal{E}^j(d_3) \quad (39)$$

where for $k = 1, 2, 3$

$$\mathcal{E}^j(d_k) = \int_{E_j} w_1^j \left(\left(\frac{\partial d_k}{\partial u}\right)^2 + \left(\frac{\partial d_k}{\partial v}\right)^2\right) + w_0^j d_k^2 \, \mathrm{du}. \quad (40)$$

In accordance to the theory of elasticity, (40) can be written in the form

$$\mathcal{E}^j(d_k) = \int_{E_j} {\boldsymbol{\sigma}_k^j}^T \boldsymbol{\epsilon}_k^j \, \mathrm{du} \quad (41)$$

where

$$\boldsymbol{\epsilon}_k^j = \left[\frac{\partial d_k}{\partial u}, \frac{\partial d_k}{\partial v}, d_k\right]^T \quad (42)$$

is the strain vector and

$$\boldsymbol{\sigma}_k^j = \boldsymbol{D}_k^j \boldsymbol{\epsilon}_k^j = \begin{pmatrix} w_1^j & 0 & 0 \\ 0 & w_1^j & 0 \\ 0 & 0 & w_0^j \end{pmatrix} \boldsymbol{\epsilon}_k^j \quad (43)$$

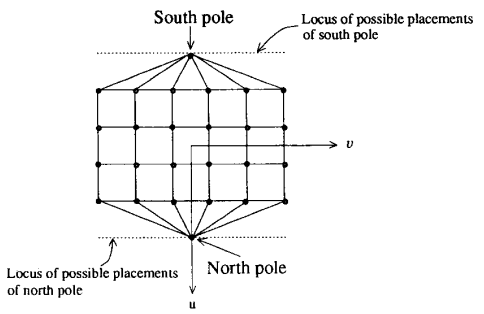


Fig. 9. Model tesselation in material coordinates.

is the stress vector associated with component k of $\boldsymbol{d}$. Therefore, the element stress vector is

$$\boldsymbol{\sigma}^j = \boldsymbol{D}^j \boldsymbol{\epsilon}^j = \begin{pmatrix} \boldsymbol{D}_1^j & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}_2^j & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{D}_3^j \end{pmatrix} \begin{pmatrix} \boldsymbol{\epsilon}_1^j \\ \boldsymbol{\epsilon}_2^j \\ \boldsymbol{\epsilon}_3^j \end{pmatrix} \quad (44)$$

where $\boldsymbol{D}_1^j = \boldsymbol{D}_2^j = \boldsymbol{D}_3^j$.

We denote the finite element nodal shape functions by N_i^j, $i = 1, \cdots, n$, where n is the number of nodes associated with element E_j. Hence, we can write (42) as

$$\boldsymbol{\epsilon}_k^j = \sum_{i=1}^{n} \boldsymbol{\gamma}_i^j \left(q_{d_k}^j\right)_i = \boldsymbol{\Gamma}_k^j \boldsymbol{q}_{d_k}^j \quad (45)$$

where $\boldsymbol{\gamma}_i^j = \left[\frac{\partial N_i^j}{\partial u}, \frac{\partial N_i^j}{\partial v}, 1\right]^T$, $\boldsymbol{\Gamma}_k^j = \left(\boldsymbol{\gamma}_1^j \boldsymbol{\gamma}_2^j \cdots \boldsymbol{\gamma}_n^j\right)$, and $\boldsymbol{q}_{d_k}^j = \left[\left(q_{d_k}^j\right)_1, \left(q_{d_k}^j\right)_2, \cdots, \left(q_{d_k}^j\right)_n\right]^T$. We can write the element strain vector $\boldsymbol{\epsilon}^j$ as

$$\boldsymbol{\epsilon}^j = \begin{pmatrix} \boldsymbol{\epsilon}_1^j \\ \boldsymbol{\epsilon}_2^j \\ \boldsymbol{\epsilon}_3^j \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Gamma}_1^j & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Gamma}_2^j & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Gamma}_3^j \end{pmatrix} \begin{pmatrix} \boldsymbol{q}_{d_1}^j \\ \boldsymbol{q}_{d_2}^j \\ \boldsymbol{q}_{d_3}^j \end{pmatrix} = \boldsymbol{\Gamma}^j \boldsymbol{q}_d^j \quad (46)$$

where $\boldsymbol{\Gamma}_1^j = \boldsymbol{\Gamma}_2^j = \boldsymbol{\Gamma}_3^j$. Thus, the element stiffness matrix is

$$\boldsymbol{K}_{dd}^j = \int_{E_j} {\boldsymbol{\Gamma}^j}^T \boldsymbol{D}^j \boldsymbol{\Gamma}^j \, \mathrm{du} = \mathrm{diag}\left(\int_{E_j} {\boldsymbol{\Gamma}_k^j}^T \boldsymbol{D}_k^j \boldsymbol{\Gamma}_k^j \, \mathrm{du}\right). \quad (47)$$

Fig. 9 illustrates the tessellation of the material coordinate system $\mathrm{u} = (u, v)$ into finite element domains. The need for quadrilateral and triangular elements is evident. Equation (4) implies that the v material coordinate of both north ($u = \pi/2$) and south ($u = -\pi/2$) poles may be arbitrary. This is illustrated in the figure by the dotted lines. The next two sections describe the bilinear quadrilateral and linear triangular elements that we employ in our implementation.

1. Bilinear Quadrilateral Elements

The nodal shape functions of the bilinear quadrilateral element (Fig. 10) are

$$N_i(\xi, \eta) = \frac{1}{4}(1 + \xi\xi_i)(1 + \eta\eta_i) \quad (48)$$

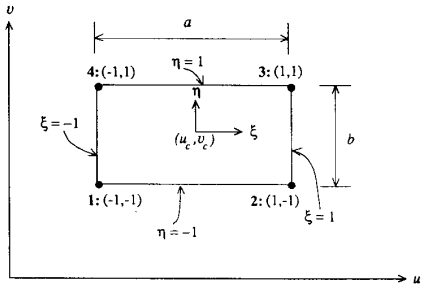


Fig. 10. Bilinear quadrilateral element. The four nodes are numbered.

where (ξ_i, η_i) are the reference coordinates of node i shown in the figure. The relationship between the reference coordinates and material coordinates u = (u, v) is given by

$$\xi = \frac{2}{a}(u - u_c), \qquad \eta = \frac{2}{b}(v - v_c) \tag{49}$$

where (u_c, v_c) are the coordinates of the element center. The required derivatives of the shape functions may be computed as follows:

$$\frac{\partial N_i}{\partial u} = \frac{\partial N_i}{\partial \xi}\frac{\partial \xi}{\partial u} + \frac{\partial N_i}{\partial \eta}\frac{\partial \eta}{\partial u} = \frac{1}{2a}\xi_i(1 + \eta\eta_i) \tag{50}$$

$$\frac{\partial N_i}{\partial v} = \frac{\partial N_i}{\partial \xi}\frac{\partial \xi}{\partial v} + \frac{\partial N_i}{\partial \eta}\frac{\partial \eta}{\partial v} = \frac{1}{2b}(1 + \xi\xi_i)\eta_i \tag{51}$$

and we may integrate a function $f(u, v)$ over E_j by transforming to the reference coordinate system:

$$\int\int_{E_j} f(u, v)\, du\, dv = \int_{-1}^{1}\int_{-1}^{1} f(\xi, \eta)\frac{ab}{4}\, d\xi\, d\eta. \tag{52}$$

We approximate such integrals using Gauss-Legendre quadrature rules.

Using the above formulas, we can compute the matrix $\int \boldsymbol{\Gamma}_k^{j^T} \boldsymbol{D}_k^j \boldsymbol{\Gamma}_k^j\, du$ and, hence, the element stiffness matrix $\boldsymbol{K}_{dd}^j$ for the bilinear quadrilateral element.

2. Linear Triangular Elements

As we stated above, the v coordinates of the poles are arbitrary. Hence, we can use right triangular elements in which the v coordinate of the pole is set equal to the v coordinate of one of the other nodes in each triangle.

a) North Pole Linear Triangular Elements: The nodal shape functions for the north pole linear triangular element (Fig. 11) are

$$N_1(\xi, \eta) = 1 - \xi - \eta \tag{53}$$

$$N_2(\xi, \eta) = \xi \tag{54}$$

$$N_3(\xi, \eta) = \eta. \tag{55}$$

The relationship between the uv and $\xi\eta$ coordinates is

$$\xi = \frac{1}{a}(u - u_1) \tag{56}$$

$$\eta = \frac{1}{b}(v - v_1) \tag{57}$$

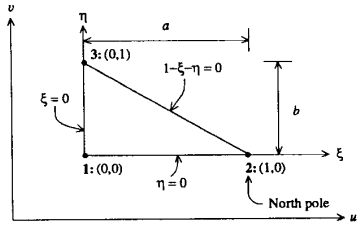


Fig. 11. North pole linear triangular element. The three nodes are numbered.

where (u_1, v_1) are the coordinates of node 1 at which $(\xi_1, \eta_1) = (0, 0)$. Computing the derivatives of the shape functions as in (50) and (51) yields

$$\frac{\partial N_1}{\partial u} = -\frac{1}{a}; \quad \frac{\partial N_2}{\partial u} = \frac{1}{a}; \quad \frac{\partial N_3}{\partial u} = 0 \tag{58}$$

$$\frac{\partial N_1}{\partial v} = -\frac{1}{b}; \quad \frac{\partial N_2}{\partial v} = 0; \quad \frac{\partial N_3}{\partial v} = \frac{1}{b} \tag{59}$$

and we may integrate a function $f(u, v)$ over the E_j using

$$\int\int_{E_j} f(u, v)\, du\, dv = \int_0^1 \int_0^{(1-\eta)} f(\xi, \eta) ab\, d\xi\, d\eta. \tag{60}$$

We approximate such integrals using Radau quadrature rules.

Using the above formulas, we can compute the matrix $\int \boldsymbol{\Gamma}_k^{j^T} \boldsymbol{D}_k^j \boldsymbol{\Gamma}_k^j\, du$ and, hence, $\boldsymbol{K}_{dd}^j$ for the north pole linear triangular element.

b) South Pole Linear Triangular Elements: The nodal shape functions for the south pole linear triangular element (Fig. 12) are

$$N_1(\xi, \eta) = 1 - \xi \tag{61}$$

$$N_2(\xi, \eta) = \xi - \eta \tag{62}$$

$$N_3(\xi, \eta) = \eta. \tag{63}$$

The relationship between the uv and $\xi\eta$ coordinates is

$$\xi = \frac{1}{a}(u - u_1) \tag{64}$$

$$\eta = \frac{1}{b}(v - v_1) \tag{65}$$

where (u_1, v_1) are the coordinates of node 1 at which $(\xi_1, \eta_1) = (0, 0)$. Computing the derivatives of the shape functions as in (50) and (51) yields

$$\frac{\partial N_1}{\partial u} = -\frac{1}{a}; \quad \frac{\partial N_2}{\partial u} = \frac{1}{a}; \quad \frac{\partial N_3}{\partial u} = 0 \tag{66}$$

$$\frac{\partial N_1}{\partial v} = 0; \quad \frac{\partial N_2}{\partial v} = -\frac{1}{b}; \quad \frac{\partial N_3}{\partial v} = \frac{1}{b} \tag{67}$$

and we may integrate a function $f(u, v)$ over the E_j using

$$\int\int_{E_j} f(u, v)\, du\, dv = \int_0^1 \int_0^{(1-\xi)} f(\xi, \eta) ab\, d\eta\, d\xi. \tag{68}$$

We approximate such integrals using Radau quadrature rules.

Using the above formulas, we can compute the matrix $\int \boldsymbol{\Gamma}_k^{j^T} \boldsymbol{D}_k^j \boldsymbol{\Gamma}_k^j\, du$ and, hence, $\boldsymbol{K}_{dd}^j$ for the south pole linear triangular element.

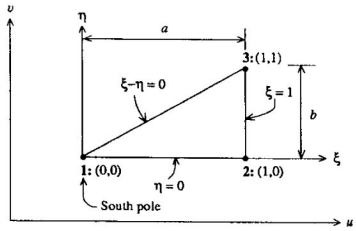


Fig. 12. South pole linear triangular element. The three nodes are numbered.

REFERENCES

[1] A. Barr, "Superquadrics and angle-preserving transformations," *IEEE Comput. Graphics Applications,* vol. 18, pp. 21–30, 1981.

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

[3] I. Biederman, "Human image understanding: Recent research and theory," *Comput. Vision Graphics Image Processing,* vol. 32, pp. 29–73, 1985.

[4] A. Blake and A. Zisserman, *Visual Reconstruction.* Cambridge, MA: MIT Press, 1987.

[5] R. Brooks, "Symbolic reasoning among 3D models and 2D images," *Art. Intell.,* vol. 17, pp. 285–348, 1981.

[6] M. Gardiner, "The superellipse: A curve between the ellipse and the rectangle," *Scientific Amer.,* vol. 213, pp. 222–234, 1965.

[7] A. D. Gross and T. E. Boult, "Error of fit measures for recovering parametric solids," in *Proc. Sec. Int. Conf. Comput. Vision,* 1988, pp. 690–694.

[8] A. Gupta and R. Bajcsy, "Part description and segmentation using contour, surface, and volumetric primitives," in *Sensing and Reconstruction of Three-Dimensional Objects and Scenes, Proc. SPIE 1260,* B. Girod (Ed.), 1990, pp. 203–214.

[9] B. B. Kimia, A. Tannenbaum, and S. W. Zucker, "Toward a computational theory of shape: An overview," Tech. Rep. TR-CIM-89-13, Comput. Vision Robotics Lab., 1989.

[10] D. Marr and H. K. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," *Proc. Roy. Soc. London B,* vol. 200, pp. 269–294, 1978.

[11] A. Pentland, "Perceptual organization and the representation of natural form," *Art. Intell.,* vol. 28, pp. 293–331, 1986.

[12] ——, "Automatic extraction of deformable part models," Tech. Rep. Vision-Sciences 104, MIT Media Lab., 1988.

[13] ——, "Canonical fitting of deformable part models," in *Sensing and Reconstruction of Three-Dimensional Objects and Scenes, Proc. SPIE 1260,* B. Girod (Ed.), 1990, pp. 216–228.

[14] T. Poggio. V. Torre, and C. Koch, "Computational Vision and regularization theory," *Nature,* vol. 317, pp. 314–319, 1985.

[15] M. Rioux and L. Cournoyer, "The NRCC three-dimensional image data files," Tech. Rep. CNRC 29077, Nat. Res. Council Canada, 1988.

[16] A. A. Shabana, *Dynamics of Multibody Systems.* New York: Wiley, 1989.

[17] F. Solina and R. Bajcsy, "Recovery of parametric models from range images: The case for superquadrics with global deformations," *IEEE Trans Patt. Anal. Mach. Intell.,* vol. 12, pp. 131–146, 1990.

[18] R. Szeliski, *Bayesian Modeling of Uncertainty in Low-Level Vision.* Boston: Kluwer, 1989.

[19] D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities," *IEEE Trans. Patt. Anal. Mach. Intell.,* vol. PAMI-8, pp. 413–424, 1986.

[20] ——, "The computation of visible-surface representations," *IEEE Trans. Patt. Anal. Mach. Intell.,* vol. 10, pp. 417–438, 1988.

[21] D. Terzopoulos and A. Witkin, "Physically based models with rigid and deformable components," *IEEE Comput. Graphics Applications,* vol. 8, pp. 41–51, 1988.

[22] D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on deformable models: Recovering 3D shape and nonrigid motion," *Art. Intell.,* vol. 36, pp. 91–123, 1988.

[23] A. Witkin and W. Welch, "Fast animation and control of nonrigid structures," *Comput. Graphics,* vol. 24, pp. 243–252, 1990.