A Finite Element Model for 3D Shape Reconstruction and Nonrigid Motion Tracking

Tim McInerney and **Demetri Terzopoulos**[1]
Department of Computer Science, University of Toronto, Toronto, ON, Canada M5S 1A4

Abstract

This paper presents a physics-based approach for recovering the 3D shape and tracking the motion of nonrigid objects using a 3D elastically deformable balloon model. The balloon model is based on a thin-plate under tension spline which deforms to fit visual data according to internal forces stemming from the elastic properties of the surface and external forces which are produced from the data. We employ the finite element method to represent the model as a continuous surface. We use a "natural" finite element whose nodal variables comprise the position of the surface plus its first and second partial derivatives, reflecting each of the partial derivatives that occur in the spline's strain energy functional. Hence, the model directly estimates all the information needed to measure the differential geometric properties of the fitted surface. We apply the the balloon model to the reconstruction of 3D objects with irregular shape features and demonstrate its effectiveness in extracting the left ventricular surface and tracking its nonrigid motion in dynamic CT volume images.

1 Introduction

Reconstruction and representation of nonrigid 3D objects requires models with the flexibility to assume a wide variety of shapes and the ability to track complex motions. The models must be able to accurately extract meaningful information from noise-corrupted sensor data while making the weakest possible assumptions about observed shapes. To support subsequent analysis and interpretation of the reconstructed objects, the models should provide explicit information about geometry and motion in a direct and convenient form. These sorts of modeling capabilities are becoming increasingly important, not only in computer vision proper, but also in related fields such as medical image analysis. For example, physicians can relate the stretching of the left ventricle (LV) during the cardiac cycle to heart condition. Powerful models are needed to measure and interpret the motions of dynamically deforming objects such as the LV.

Surface reconstruction techniques based on splines have attracted significant interest for many years because of their versatility. However, the dynamic form of this type of surface reconstruction was introduced fairly recently by Terzopoulos *et al.* [15]. They proposed a dynamic deformable cylinder model constructed from generalized splines and developed force field techniques to fit the model to image data. The dynamic model fitting approach is being pursued by several researchers [7, 10, 3, 4, 16, 6, 17], as it is in this paper.

This paper presents a physics-based approach to the reconstruction of object shape and nonrigid motion tracking using a 3D deformable "balloon" model. The model is dynamic, and its deformation is governed by the laws of nonrigid motion. The formulation of the motion equations includes strain energies, simulated forces, and other physical quantities. The strain energy stems from a thin-plate under tension spline which includes the squared first and second order parametric partial derivatives of the surface. Deformation results from the action of internal forces (which impose surface continuity constraints) and external forces which deform the surface away from its initial shape such that it fits the data. The dynamic formulation of the model makes it suitable both for static shape recovery problems, as well as for problems involving the estimation and tracking of nonrigidly moving objects.

The deformable balloon is formulated as a closed 3D surface that is topologically isomorphic to a sphere. We employ the finite element method to discretize the balloon into a set of connected triangular element domains using a uniform tessellation. The finite element method provides an analytic surface representation and it generally requires fewer discretization points than finite difference techniques. We use a finite element whose nodal degrees of freedom include not only the nodal positions, but also the first and second order parametric derivative terms that occur in the thin-plate under tension energy expression. In this sense, the element is naturally suited to the energy functional. This "rich" representation allows us to easily compute the differential characteristics of the fitted model surface. The model automatically computes all the derivatives needed to calculate the first and second fundamental forms for subsequent surface analysis.

Finite element representations for variational problems in vision were first explored in [13]. Our formulation applies the finite element method to the thin-plate under tension spline proposed in [14] in order to derive discrete nonrigid dynamics equations. Our work is related to that of [2] who also develop a 3D deformable surface model which is based on the thin-plate under tension spline. However, we develop a uniformly tessellated, closed 3D surface model and employ a quintic finite element whose nodal variables comprise the position of the model surface plus its first and second partial derivatives.

2 Dynamic Deformable Surface Model

The balloon model that we develop in this paper is constructed of the simulated thin-plate material under tension. The deformation energy of the material serves as a smoothness constraint which compels the model to vary smoothly almost everywhere. The balloon is implemented using a vector-valued parametric representation $\mathbf{x}(u,v) = [x(u,v), y(u,v), z(u,v)]^\mathsf{T}$ where vector $\mathbf{x}$ represents the positions of material points (u,v) relative to a reference frame in Euclidean 3-space.

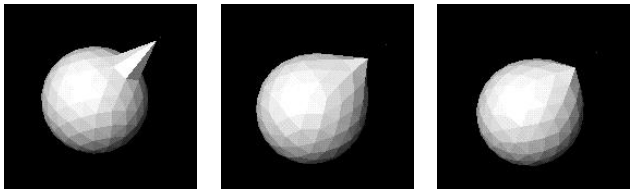
The deformation energy of the thin plate material under tension is given by the functional

$$\mathcal{E}_\mathrm{p}(\mathbf{x}) = \iint \alpha_{10}\left|\mathbf{x}_u\right|^2 + \alpha_{01}\left|\mathbf{x}_v\right|^2 + \beta_{20}\left|\mathbf{x}_{uu}\right|^2 + \beta_{11}\left|\mathbf{x}_{uv}\right|^2 + \beta_{02}\left|\mathbf{x}_{vv}\right|^2 \, du\, dv, \tag{1}$$

where the u and v subscripts denote parametric derivatives. $\mathcal{E}_\mathrm{p}$ is a controlled-continuity spline defined in [14]. The nonnegative weighting functions $\alpha_{ij}(u,v)$ and $\beta_{ij}(u,v)$ control

[1] Fellow, Canadian Institute for Advanced Research

the elasticity of the material. The α_{10} and α_{01} functions control the tensions in the u and v directions, respectively, while the β_{02} and β_{20} functions control the corresponding bending rigidities, and the β_{11} function controls the twisting rigidity. Increasing the α_{ij} has a tendency to decrease the surface area of the material, while increasing the β_{ij} tends to make it less flexible. In general, the weighting functions may be used to introduce depth and orientation discontinuities in the material. In this paper, however, we do not make use of this capability, and set the functions to constant values $\alpha_{ij}(u,v) = \alpha_{ij}$ and $\beta_{ij}(u,v) = \beta_{ij}$. The following figure shows the thin plate under tension balloon being pulled radially by a spring point force (in (a) $\alpha_{ij} = 0.8$ and $\beta_{ij} = 0$, in (b) $\alpha_{ij} = \beta_{ij} = 0.5$, and in (c) $\alpha_{ij} = 0$ and $\beta_{ij} = 0.8$):



(a) (b) (c)

A general and elegant approach to fitting deformable surface models to data, especially when the data are time-varying, is to make the models *dynamic*. A dynamic formulation naturally supports model visualization during the data fitting process, and it allows a user to introduce constraint forces to pull the model out of local minima towards a good solution.

In a dynamic formulation, the positions of material points becomes a time-dependent function $\mathbf{x}(u,v,t)$ and we imbue the simulated material with mass and damping densities. Using Lagrangian dynamics the energies yield forces and $\mathcal{E}_{\mathrm{p}}(\mathbf{x})$ is minimized when the forces equilibrate and the model becomes stationary $\partial\mathbf{x}/\partial t = \partial^2\mathbf{x}/\partial t^2 = \mathbf{0}$.

The dynamic behavior of the models during the fitting process is governed by

$$\mu\frac{\partial^2\mathbf{x}}{\partial t^2} + \gamma\frac{\partial\mathbf{x}}{\partial t} + \delta_{\mathbf{x}}\mathcal{E}_{\mathrm{p}} = \mathbf{f}. \tag{2}$$

where the first term represents the inertial forces due to the mass density $\mu(u,v)$, the second term represents the damping forces due to the damping density $\gamma(u,v)$, the third term represents the elastic force which resist deformation, and finally $\mathbf{f}(u,v,t)$ represents the data forces. The (generally nonlinear) data forces may be thought of as stemming from a data functional

$$\mathcal{E}_{\mathrm{d}}(\mathbf{x}) = -\int\!\!\int \mathbf{x}^{\mathsf{T}}\mathbf{f}\,du\,dv. \tag{3}$$

3 Finite Element Representation

The finite difference method and the finite element method may be applied to compute approximate numerical solutions to the function $\mathbf{x}$. Finite difference solutions approximate the continuous function $\mathbf{x}$ as a set of discrete points in space. A disadvantage of the approach is that the continuity of the solution between points is not made explicit. The finite element method, on the other hand, provides continuous approximations; that is, the method approximates the unknown function $\mathbf{x}$ by local basis functions [5].

To apply the finite element method to our models, we tessellate the continuous material domain (u,v) into a mesh of M element subdomains E_j. We approximate $\mathbf{x}$ as a weighted sum of piecewise polynomial basis functions $\mathbf{N}_i$:

$$\mathbf{x}(u,v,t) \approx \hat{\mathbf{x}}(u,v,t) = \sum_{i=1}^{n}\mathbf{N}_i(u,v)\mathbf{q}_i(t), \tag{4}$$

where $\mathbf{q}_i$ is a vector of nodal variables associated with mesh node i.

Substituting (4) into (2) yields the discrete equations of motion

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{f}_q, \tag{5}$$

with $\mathbf{q} = [\mathbf{q}_1^{\mathsf{T}}, \ldots, \mathbf{q}_i^{\mathsf{T}}, \ldots, \mathbf{q}_n^{\mathsf{T}}]^{\mathsf{T}}$, where the mass matrix $\mathbf{M}$, damping matrix $\mathbf{C}$, and stiffness matrix $\mathbf{K}$ are sparse, symmetric matrices and vector $\mathbf{f}_q$ are nodal data forces. These *global* matrices may be assembled from their associated *local* element matrices by expanding each element matrix appropriately into a $q \times q$ matrix and then summing:

$$\mathbf{M} = \sum_{\mathrm{j}=1}^{\mathrm{M}}\mathbf{M}^{\mathrm{j}}_{\mathrm{q}\times\mathrm{q}};\ \mathbf{C} = \sum_{\mathrm{j}=1}^{\mathrm{M}}\mathbf{C}^{\mathrm{j}}_{\mathrm{q}\times\mathrm{q}};\ \mathbf{K} = \sum_{\mathrm{j}=1}^{\mathrm{M}}\mathbf{K}^{\mathrm{j}}_{\mathrm{q}\times\mathrm{q}};\ \mathbf{f}_{\mathrm{q}} = \sum_{\mathrm{j}=1}^{\mathrm{M}}\mathbf{f}^{\mathrm{j}}_{\mathrm{q}}, \tag{6}$$

where $\mathbf{M}^j$, $\mathbf{C}^j$, $\mathbf{K}^j$, and $\mathbf{f}_q^j$ are element mass, damping, and stiffness matrices, and nodal data forces associated with element E_j, $j = 1, \ldots, M$.

We now derive expressions for $\mathbf{M}^j$, $\mathbf{C}^j$, $\mathbf{K}^j$, and $\mathbf{f}_q^j$ from element kinetic and potential energy functionals. Let $\mathbf{x}^j(u,v,t)$ be the position of material point (u,v) within E_j, and let $\mathbf{q}^j$ denote the concatenation of nodal variables for all the nodes of E_j. Following (4), we write the element trial function

$$\hat{\mathbf{x}}^j(u,v,t) = \mathbf{N}^j(u,v)\mathbf{q}^j(t) \approx \mathbf{x}^j(u,v,t), \tag{7}$$

where $\mathbf{N}^j$ are the element shape functions. Note that, as was stated previously, the basis functions $\mathbf{N}_i$ are obtained by superposing the shape functions associated with node i. The element velocity is $\partial\hat{\mathbf{x}}^j/\partial t = \mathbf{N}^j\dot{\mathbf{q}}^j$, where $\dot{\mathbf{q}}^j(t)$ is the rate of change of the nodal variables.

The kinetic energy associated with element E_j is

$$\frac{1}{2}\int\!\!\int_{\mathrm{E_j}}\mu\frac{\partial\hat{\mathbf{x}}^{\mathrm{j}\,\mathsf{T}}}{\partial\mathrm{t}}\frac{\partial\hat{\mathbf{x}}^{\mathrm{j}}}{\partial\mathrm{t}}\,\mathrm{du\,dv} = \frac{1}{2}\dot{\mathbf{q}}^{\mathrm{j}\,\mathsf{T}}\mathbf{M}^{\mathrm{j}}\dot{\mathbf{q}}^{\mathrm{j}}, \tag{8}$$

where the element mass matrix is given by

$$\mathbf{M}^j = \int\!\!\int_{E_j}\mu\mathbf{N}^{j\,\mathsf{T}}\mathbf{N}^j\,du\,dv. \tag{9}$$

We introduce a simple velocity-proportional kinetic energy dissipation according to the (Raleigh) dissipation functional

$$\frac{1}{2}\int\!\!\int_{E_j}\gamma\frac{\partial\hat{\mathbf{x}}^{j\,\mathsf{T}}}{\partial t}\frac{\partial\hat{\mathbf{x}}^j}{\partial t}\,du\,dv = \frac{1}{2}\dot{\mathbf{q}}^{j\,\mathsf{T}}\mathbf{C}^j\dot{\mathbf{q}}^j. \tag{10}$$

The element damping matrix is proportional to the mass matrix and is given by

$$\mathbf{C}^j = \int\!\!\int_{E_j}\gamma\mathbf{N}^{j\,\mathsf{T}}\mathbf{N}^j\,du\,dv. \tag{11}$$

According to (1) the element deformation matrix may be expressed as

$$\mathcal{E}_{\mathrm{p}}^j(\mathbf{x}) = \int\!\!\int_{E_j}\boldsymbol{\sigma}^{j\,\mathsf{T}}\boldsymbol{\epsilon}^j\,du\,dv, \tag{12}$$

where the strain vector is

$$\boldsymbol{\epsilon}^j = \left[\mathbf{x}_u^{j\,\mathsf{T}}, \mathbf{x}_v^{j\,\mathsf{T}}, \mathbf{x}_{uu}^{j\,\mathsf{T}}, \mathbf{x}_{uv}^{j\,\mathsf{T}}, \mathbf{x}_{vv}^{j\,\mathsf{T}}\right]^{\mathsf{T}} = \mathbf{L}\mathbf{x}^j \tag{13}$$

and the stress vector is

$$\boldsymbol{\sigma}^j = \begin{bmatrix} \alpha_{10}^j \mathbf{I} & 0 & 0 & 0 & 0 \\ 0 & \alpha_{01}^j \mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & \beta_{20}^j \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & \beta_{11}^j \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & \beta_{02}^j \mathbf{I} \end{bmatrix} \boldsymbol{\epsilon}^j = \mathbf{D}^j \boldsymbol{\epsilon}^j, \quad (14)$$

with $\mathbf{I}$ a 3×3 unit matrix. Using (7), we can write

$$\boldsymbol{\epsilon}^j = \mathbf{L}\mathbf{N}^j \mathbf{q}^j = \mathbf{B}^j \mathbf{q}^j, \quad (15)$$

where $\mathbf{B}^j$ is the element strain matrix. Inserting the expressions for $\boldsymbol{\epsilon}^j$ and $\boldsymbol{\sigma}^j$ into (12) yields

$$\mathcal{E}_{\mathrm{p}}^j(\mathbf{x}) = \mathbf{q}^{j\top} \mathbf{K}^j \mathbf{q}^j, \quad (16)$$

where the element stiffness matrix is given by

$$\mathbf{K}^j = \iint_{E_j} \mathbf{B}^{j\top} \mathbf{D}^j \mathbf{B}^j \, du \, dv. \quad (17)$$

Finally, according to (3), the potential energy in element E_j due to data forces $\mathbf{f}^j(u,v,t)$ is

$$-\iint_{E_j} \hat{\mathbf{x}}^{j\top} \mathbf{f}^j \, du \, dv = -\mathbf{q}^{j\top} \mathbf{f}_q^j, \quad (18)$$

where the nodal data forces are given by

$$\mathbf{f}_q^j = \iint_{E_j} \mathbf{N}^{j\top} \mathbf{f}^j \, du \, dv. \quad (19)$$

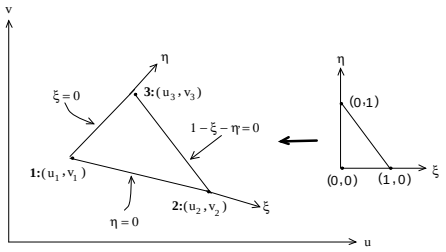
4 Model Structure

The balloon model is a closed surface in Euclidean 3-space which is topologically isomorphic to a sphere. We initially discretize the balloon in the material coordinates (u,v) by tessellating it into a set of 20 triangular elements to form an icosahedron. We chose the icosahedron because it has a simple representation in material coordinates and it has a regular structure in Euclidean 3-space, with each of the 12 nodes connected to five neighboring nodes.

4.1 Triangular C^1 Finite Element

We use a fifth-order triangular finite element to implement the balloon model [5]. In view of the form of the thin-plate under tension energy (1) which leads to the strain vector (13), it is natural to choose as nodal variables $\mathbf{x}$, along with its first and second parametric partial derivatives evaluated at each node i. The nodal variable vector for the balloon is therefore

$$\mathbf{q}_i(t) = \left[\mathbf{x}_i^\top, (\mathbf{x}_u)_i^\top, (\mathbf{x}_v)_i^\top, (\mathbf{x}_{uu})_i^\top, (\mathbf{x}_{uv})_i^\top, (\mathbf{x}_{vv})_i^\top\right]^\top. \quad (20)$$



The figure shows the C^1 continuous element defined locally in the dimensionless oblique coordinates (ξ,η). In this local coordinate system the material coordinates (u,v) can be expressed as

$$\begin{aligned} u &= (1-\xi-\eta)u_3 + \xi u_1 + \eta u_2, \\ v &= (1-\xi-\eta)v_3 + \xi v_1 + \eta v_2, \end{aligned} \quad (21)$$

where (u_i, v_i) are the material coordinates at the nodes (as numbered in the figure), and the local nodal variable vector becomes

$$\mathbf{q}_{i_\xi}(t) = \left[\mathbf{x}_i^\top, (\mathbf{x}_\xi)_i^\top, (\mathbf{x}_\eta)_i^\top, (\mathbf{x}_{\xi\xi})_i^\top, (\mathbf{x}_{\xi\eta})_i^\top, (\mathbf{x}_{\eta\eta})_i^\top\right]^\top. \quad (22)$$

The transformation from global to local coordinates is

$$\mathbf{q}_i = \mathbf{T}_i \mathbf{q}_{i_\xi} \quad (23)$$

where the transformation matrix $\mathbf{T}_i$ is specified in [5] (pp. 100–101).

Concatenating the $\mathbf{q}_{i_\xi}$ at each of the three nodes of element j, we obtain the 18-dimensional element nodal vector $\mathbf{q}_\xi^j = [\mathbf{q}_{1_\xi}^\top, \mathbf{q}_{2_\xi}^\top, \mathbf{q}_{3_\xi}^\top]^\top$. According to (7), we can write the local trial function as $\hat{\mathbf{x}}^j(\xi,\eta,t) = \mathbf{N}^j(\xi,\eta)\mathbf{q}_\xi^j(t)$. The nodal shape functions $\mathbf{N}_i(\xi,\eta)$ which are contained in the 18×18 matrix $\mathbf{N}^j$ are specified in [5] (pp. 100–101). Note that the polynomial basis of the element is complete up to fourth-order terms and contains three fifth-order terms. The trial functions are C^∞ within elements and they ensure C^1 continuity between elements. Since (1) contains up to second order derivatives, the element is conforming.

The shape functions are expressed in terms of the local coordinates (ξ,η) and it is convenient to work with these coordinates. Thus, the required derivatives of the shape functions in the strain matrix $\mathbf{B}$ are computed using repeated applications of the chain rule and equation (22). Also, a function $f(u,v)$ may be integrated over E_j by transforming to the local coordinate system as follows:

$$\iint_{E_j} f(u,v)\, du\, dv = \iint_{E_j} f(u(\xi,\eta), v(\xi,\eta)) \det \mathbf{J} \, d\xi \, d\eta, \quad (24)$$

where

$$\mathbf{J} = \begin{bmatrix} \frac{\partial u}{\partial \xi} & \frac{\partial v}{\partial \xi} \\ \frac{\partial u}{\partial \eta} & \frac{\partial v}{\partial \eta} \end{bmatrix} \quad (25)$$

is the Jacobian matrix. These integrals are approximated using Gauss-Legendre quadrature rules.

4.2 Model Refinement

Our implementation allows the balloon model to be refined during the fitting process by subdividing the triangular elements. Each element spawns 4 child elements by connecting the midpoints of its 3 edges. This process may be applied recursively to each child element. The connectivity of all new vertices formed in this fashion is six, while the original 12 vertices of the icosahedron remain five-connected. The refinement process enables a low resolution model to initially fit the data, reconstructing the rough overall shape, then creates a finer resolution model to capture the detail. This approach greatly reduces the computation time required for the reconstruction process.

Since a global subdivision of the balloon model increases the number of element nodes by approximately fourfold, the usefulness of this procedure is limited. A better scheme is to locally subdivide the model in areas where the shape implied by the data varies considerably. Local subdivision is not pursued in this paper.

5 Applied Forces

Our dynamic surface reconstruction paradigm applies data constraints to the model as external force distributions $\mathbf{f}(u,v,t)$. The contribution of the force distribution to each element E_j is converted through (19) to generalized forces $\mathbf{f}_q^j$ associated with the nodal variables of the element.

5.1 3D Image Forces

When extracting and reconstructing surfaces from 3D image data, we design forces that localize salient image features. For example, to attract our model towards significant 3D "edges" (intensity gradients) in some region of an image function $I(x,y,z)$ we construct a 3D potential function

$$P(x,y,z) = \lambda_1 \left\|\nabla(G_\sigma * I)\right\| + \lambda_2 \left\|O_{\mathrm{CDM}} * I\right\| \qquad (26)$$

whose potential "valleys" (minima) coincide with the object surface [15]. In the first term on the right hand side of (26), G_σ denotes a 3D Gaussian smoothing filter of characteristic width σ. This filter broadens or narrows the potential valleys of this term thus determining the extent of the region of attraction of the intensity gradient. Typically, the attraction has a relatively short range. In the second term, a 3D edge detector, the 3D Canny-Deriche-Monga (CDM) operator [9], is applied to the image data to produce a set of 3D "edge" elements. The potential valleys of this term tend to be narrow and deep, complementing (and coinciding with) the wider but more shallow valleys produced by the first term. A weighted combination of these terms is formed so the model will "slide" down the shallow valleys and then drop into the deeper valleys thus "locking" onto image edges.

The potential function produces a force distribution

$$\mathbf{f}(\mathbf{x}) = \kappa \frac{\nabla P(\mathbf{x})}{\left\|\nabla P(\mathbf{x})\right\|} \qquad (27)$$

on the model, where κ controls the strength of the force. We normalize the image force as for better numerical stability [3]. Consequently, all 3D "edge" elements, including spurious edges, will attract the model equally. However, once the model converges towards the true 3D "edges" of the object, the regularization effect of the model will give it a tendency to ignore spurious 3D "edges".

Note that to compute ∇P at any model point $\mathbf{x}(u,v)$ from a discrete 3D image data set $I(i,j,k)$, we tri-linearly interpolate $I(\mathbf{x})$ using values at the eight surrounding pixels.

5.2 Balloon Inflation Force

When extracting object surfaces from 3D image data, the balloon model must first be placed within the object. If the model is not close enough to the surface of the object, the short range image forces defined previously may not be sufficient to attract it. For this reason, an internal pressure force is used to "inflate" the balloon model towards the object surface [15] [3]. The force takes the form $\mathbf{f} = \kappa_1 \mathbf{n}(u,v)$, where $\mathbf{n}(u,v)$ is the unit normal vector to the model surface at the point $\mathbf{x}(u,v)$, and κ_1 is the amplitude of this force. If κ_1 is negative, the force will deflate the balloon. The image force scale parameters κ and κ_1 are usually chosen to be of the same order, with κ slightly larger than κ_1 so that a significant 3D "edge" will stop the inflation, but with κ_1 large enough so that the model will pass through weak or spurious edges.

5.3 Range Data Forces

When reconstructing surfaces from laser range data we define long-range spring-like "point" forces

$$\mathbf{f}(u,v) = \kappa \left\|\mathbf{p} - \mathbf{x}(u_p, v_p)\right\| \qquad (28)$$

proportional to the separation between a data point $\mathbf{p}$ in space and the point of influence (u_p, v_p) of the force on the model's surface.

We approximate (u_p, v_p) as the model node with minimal distance to the point $\mathbf{p}$, using a heuristic local neighborhood search to find the nearest model node.

5.4 User and Constraint Forces

In applications where user control over models is desirable, dynamic models offer much more than the option of manually adjusting geometric parameters. The dynamic modeling approach provides a "facile" interface to the models through the use of force interaction tools. For example, as the model is deforming, the user may use the mouse to specify spring forces to pull the model towards significant image features, or to specify "pins" which force the model to interpolate data features marked by a pin.

6 Parallel Numerical Integration

In computer vision and geometric design applications involving the fitting of models to data, we can simplify the equations of motion (5) while preserving useful dynamics by setting the mass density $\mu(u,v)$ to zero to obtain $\mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{f}_q$. This first-order dynamic system governs a model which has no inertia and comes to rest as soon as all the applied forces vanish or equilibrate.

We integrate equation (6) forward through time using an explicit first-order Euler method. This method approximates the temporal derivatives with forward finite differences. It updates the degrees of freedom $\mathbf{q}$ of the model from time t to time $t + \Delta t$ according to the formula

$$\mathbf{q}^{(t+\Delta t)} = \mathbf{q}^{(t)} + \Delta t (\mathbf{C}^{(t)})^{-1} \left(\mathbf{f}_q^{(t)} - \mathbf{K}\mathbf{q}^{(t)}\right). \qquad (29)$$

In our implementation, we do not explicitly assemble and factorize a global stiffness matrix $\mathbf{K}$ as is common practice in applied finite element analysis. Instead, we update the nodal vectors $\mathbf{q}_i^{(t+\Delta t)}$ iteratively by computing the product $\mathbf{K}^j\mathbf{q}^j$ on an element-by-element basis using the element stiffness matrices $\mathbf{K}^j$. This approach makes the model fitting process easily parallelizable.

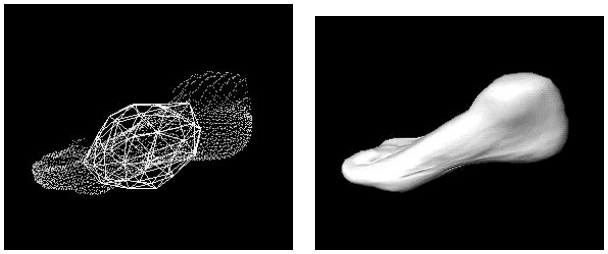
The deformable model is implemented as lists of finite elements and nodes. Furthermore, during each time step of the model fitting process actions such as the following occur:

- For each data point, compute data forces $\mathbf{f}_q$.
- For each element, accumulate the internal forces on the element nodal vectors $\mathbf{q}^j$ by computing the product $\mathbf{K}^j\mathbf{q}^j$.
- For each model node, update the position based on the applied and internal forces on the node using Euler integration.

These loops can be readily parallelized on a shared memory multiprocessor, such as our 4 processor Silicon Graphics Iris 4D/340VGX workstation, by partitioning the node, element, and data lists into equal sized sublists based on the number of processors available. Each processor then executes the loops using its assigned lists of nodes, elements, and data points.

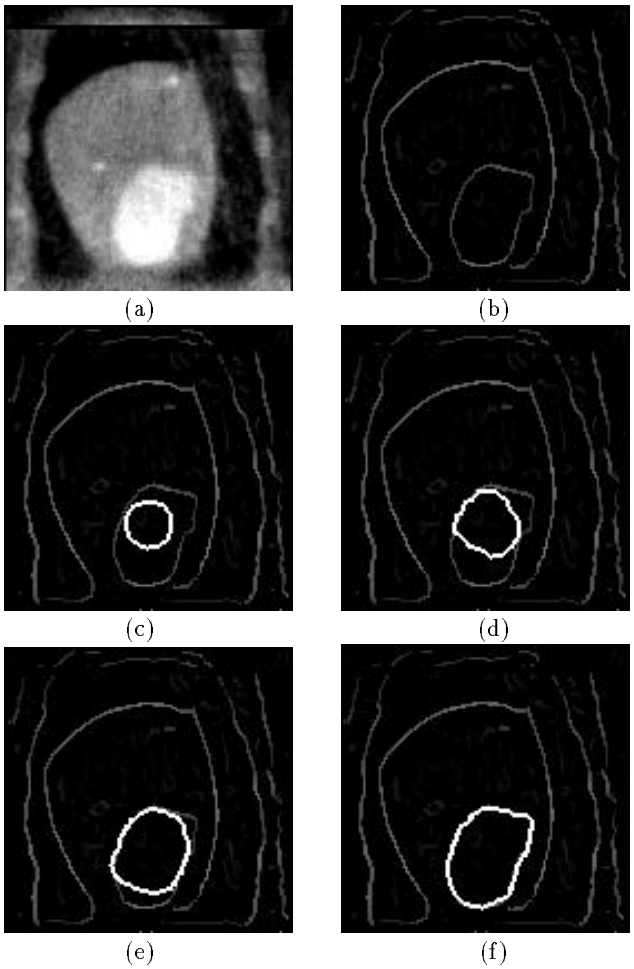
7 Surface Analysis

Once a surface has been extracted and reconstructed from the data, a logical next step is to characterize the surface structure for use in visualization, recognition, and nonrigid motion estimation and tracking. For example, surface curvature extrema often correspond to significant intrinsic features of an object [8]. Also, Gaussian curvature is independent of



(a) (b)

Figure 1: (a) Balloon model deforming to foot data. (b) Balloon model reconstruction.



(a) (b) (c) (d) (e) (f)

Figure 2: Intensity and edge detected CT image slice of left ventricle. (a) Intensity image XZ plane slice 91. (b) Edge detected image. (c) Cross section of initial balloon model. (d)-(f) Cross section of balloon deforming to left ventricle.

Figure 3: Reconstruction of Left Ventricle.

rigid motion and parameterization. Changes in this quantity over time can provide information about the local stretching or shrinking of an object's surface. Consequently, it is desirable for a surface model to not only extract accurate and meaningful information from the data, but to provide this information in a convenient and useful form.

From differential geometry, 3D smooth surfaces are uniquely characterized by their first and second fundamental forms. The parametric surface form of the balloon model used in our implementation, along with the finite element approximation, provides us with an analytical description of the reconstructed surface that is directly amenable to a differential analysis. Furthermore, since the nodal variables of our finite element contain not only the nodal position but the parametric derivatives $(\mathbf{x}_u, \mathbf{x}_v, \mathbf{x}_{uu}, \mathbf{x}_{uv}, \mathbf{x}_{uv})$ as well, all the information needed to compute the first and second fundamental forms of the surface is generated automatically. The intrinsic differential characteristics of the surface, such as the unit normal and the principal curvatures, can be conveniently computed from this information, as can the enclosed volume through the use of Green's theorem.

8 Experiments

We have applied our surface fitting techniques to a variety of data sets, including sampled laser range data and 3D medical images. Our algorithms run at interactive rates on a Silicon Graphics Iris 4D-340VGX workstation.

8.1 Surface Reconstruction from 3D Range Data

In the first experiment, we fit the deformable balloon to scattered 3D data sampled from a mannequin foot using the Cyberware, Inc., 3D digitizer. The 6000 range data points are shown in Fig. 1(a), along with the initial balloon model consisting of 80 triangular elements. Two global subdivisions of the model were performed during the fitting process to increase the number of elements to 1280. Fig. 1(b) shows the final fitted model rendered as a shaded surface.

Experiments with the parallel balloon model surface fitting algorithm indicate a consistent speedup vs the serial version which improves with increasing size of the data sets or model resolution. This result is expected since the model data structures are cleanly divided (i.e. there is little data sharing) among the processors. In fitting the balloon to the foot data, for example, a speedup of 3.6 was obtained using 4 processors on the SGI workstation. The fitting time was 59.0 seconds excluding the rendering time, which is a consistently small percentage of the total time. Due to restrictions of the VGX hardware, rendering cannot be parallelized.

8.2 Surface Reconstruction from 3D Medical Data

In this experiment we use the deformable balloon to extract and reconstruct the left ventricle (LV) from 3D medical im-

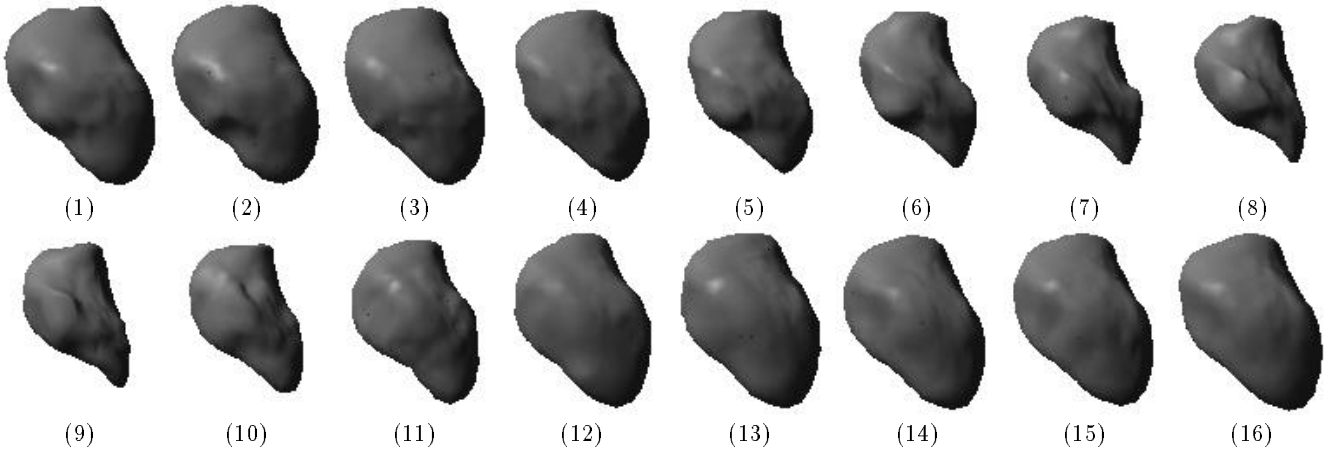


Figure 4: Tracking of the LV motion during one cardiac cycle (1–16).

ages. The data consists of 16 volume (118 slices of 128×128 pixels) images of a canine heart reconstructed within a cardiac cycle. The heart data comes from the dynamic spatial reconstructor (DSR), a high speed volumetric X-ray CT scanner, which can generate high resolution images of moving organs largely free of motion blur [12].

Figure 2 shows a cross section of the balloon model in an image slice deforming to fit the edge of the ventricle. In our system, the user can apply interactive forces to a cross-section contour as if it were a deformable contour (snakes). The final ventricle reconstruction is shown in Fig. 3. The initial model consisted of 20 triangular elements. Four global subdivisions of the model were performed during the fitting process to increase the accuracy of the reconstruction. The final model contains 5120 elements. As the balloon model deforms in 3D, it can potentially reconstruct a globally more consistent surface than can reconstruction by serial sections using deformable contours [1, 11]. It is also a more robust technique—missing slices do not seriously reduce the quality of the fit.

8.3 Tracking the LV Motion

Left ventricular motion is a well-known example of nonrigid motion. From the medical standpoint, measuring and analyzing the motion of the LV provides quantitative data for the diagnosis and treatment of heart disease. From the computer vision standpoint, measuring and tracking the complex nonrigid LV motion throughout the cardiac cycle is a challenging problem.

In this experiment, we use the balloon model to track the motion of the LV over a cardiac cycle. We begin by fitting the model to the first volume in the sequence and subsequently use this fitted model as the starting point for the reconstruction of the LV in the next volume. We continue this process for all 16 volumes making up a complete cardiac cycle.

Figure 4 shows the reconstructed LV sequence. Each fitted model contains 1280 elements. Once the initial model has been fitted to the first volume, relatively little model deformation is required to fit subsequent volumes. Furthermore, the fitting time per volume image will be less if images are generated at higher frequencies because the interframe motion will be smaller. This should lead to greater reductions in effort when the technique is applied to future image scanners.

By recovering a dynamic surface model of the LV over the whole cardiac cycle, the deformable model technique enables a direct analysis of nonrigid LV motion.

Acknowledgements

We thank the following people for their cooperation: The cardiac CT images were made available by Eric Hoffman of the University of Pennsylvania Medical School and were redistributed to us courtesy of Dmitry Goldgof, University of South Florida. The Canny-Deriche-Monga edge detector was provided courtesy of Nicholas Ayache and Gregoire Malandain of INRIA, France.

References

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

[2] I. Cohen, L.D. Cohen, and N. Ayache. Introducing New Deformable Surfaces to Segment 3D Images. In *Proc. IEEE Conf. Comp. Vis. Pat. Rec.*, pages 738–739, June 1991.

[3] L.D. Cohen. On Active Contour Models and Balloons. In *CVGIP: Image Understanding*, volume 53(2), pages 211–218, March 1991.

[4] H. Delingette, M. Hebert, and K. Ikeuchi. Shape Representation and Image Segmentation Using Deformable Surfaces. In *Proc. IEEE Conf. Comp. Vis. Pat. Rec.*, pages 467–472, June 1991.

[5] G. Dhatt and G. Touzot. *The Finite Element Method Displayed.* Wiley, New York, 1984.

[6] W.C. Huang and D.B. Goldgof. Adaptive-Size Physically-Based Models for Nonrigid Motion Analysis. In *Proc. IEEE Conf. Comp. Vis. Pat. Rec.*, pages 833–835, 1992.

[7] D. Metaxas and D. Terzopoulos. Recursive Estimation of Shape and Nonrigid Motion. In *IEEE Workshop on Visual Motion*, pages 306–311, Oct. 1991.

[8] O. Monga, N. Ayache, and P. Sander. From Voxel to Curvature. In *Proc. IEEE Conf. Comp. Vis. Pat. Rec.*, pages 644–649, June 1991.

[9] O. Monga and R. Deriche. 3D Edge Detection Using Recursive Filtering. In *Proc. IEEE Conf. Comp. Vis. Pat. Rec.*, June 1989.

[10] A. Pentland and B. Horowitz. Recovery of Nonrigid Motion and Structure. *IEEE Trans. Pat. Anal. Mach. Intel.*, 13(7):730–742, July 1991.

[11] S. Ranganath. Contour extraction using snakes: An application to cardiac MRI. Technical Report TR-91-043, Philips Laboratories, Briarcliff Manor, NY, 1991.

[12] E.L. Ritman, R.A. Robb, and L.D. Harris. *Imaging Physiological Functions: Experience with the Dynamic Spatial Reconstruction.* Praeger, New York, 1985.

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

[14] D. Terzopoulos. Regularization of Inverse Visual Problems Involving Discontinuities. *IEEE Trans. Pat. Anal. Mach. Intel.*, 8(4):413–424, 1986.

[15] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on Deformable Models: Recovering 3D Shape and Nonrigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.

[16] Y.F. Wang and J.F. Wang. Surface Reconstruction using Deformable Models with Interior and Boundary Constraints. *IEEE Trans. Pat. Anal. Mach. Intel.*, 14(5):572–579, May 1992.

[17] A. Young and L. Axel. Non-Rigid Heart Wall Motion using MR Tagging. In *Proc. IEEE Conf. Comp. Vis. Pat. Rec.*, pages 399–404, 1992.