International Journal of Computer Vision 25(1), 49–61 (1997)
© 1997 Kluwer Academic Publishers. Manufactured in The Netherlands.

Multi-Level Shape Representation Using Global Deformations and Locally Adaptive Finite Elements*

DIMITRIS METAXAS, EUNYOUNG KOH AND NORMAN I. BADLER
Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 19104-6389
dnm@central.cis.upenn.edu
badler@central.cis.upenn.edu

Received July 24, 1994; Revised February 24, 1996; Accepted May 28, 1996

Abstract. We present a model-based method for the multi-level shape, pose estimation and abstraction of an object's surface from range data. The surface shape is estimated based on the parameters of a superquadric that is subjected to global deformations (tapering and bending) and a varying number of levels of local deformations. Local deformations are implemented using locally adaptive finite elements whose shape functions are piecewise cubic functions with C^1 continuity. The surface pose is estimated based on the model's translational and rotational degrees of freedom. The algorithm first does a coarse fit, solving for a first approximation to the translation, rotation and global deformation parameters and then does several passes of mesh refinement, by locally subdividing triangles based on the distance between the given datapoints and the model. The adaptive finite element algorithm ensures that during subdivision the desirable finite element mesh generation properties of conformity, non-degeneracy and smoothness are maintained. Each pass of the algorithm uses physics-based modeling techniques to iteratively adjust the global and local parameters of the model in response to forces that are computed from approximation errors between the model and the data. We present results demonstrating the multi-level shape representation for both sparse and dense range data.

Keywords: multi-level shape representation, adaptive finite elements, deformable models, physics-based modeling

1. Introduction

We present a method for the multi-level shape estimation and abstraction of an object's surface from range data. The surface shape is estimated based on the parameters of a superquadric that is subjected to global deformations (tapering and bending) and a variable number of levels of local deformations. Local deformations are implemented based on locally adaptive finite elements, whose shape functions are piecewise cubic functions with C^1 continuity. The surface pose is estimated based on the model's translational and rotational degrees of freedom. The method is a generalization of the shape representation and estimation method developed by Metaxas and Terzopoulos (1993) and Terzopoulos and Metaxas (1991), due to the introduction of multiple levels of local deformations. Through the application of Lagrangian mechanics (Metaxas and Terzopoulos, 1993), the model's translational, rotational, global and local deformation parameters are modified based on forces that originate from the range datapoints. These forces are computed from approximation errors between the model and the data.

Even though the formulation of global deformations is independent of the number of model nodes used, local deformations require a tessellation of the model

*This work has been supported by the Army Research Office (ARO DAAL03-89-C-0031), and the National Science Foundation (NSF-IRI93-09917).

surface into a grid of finite elements. The quality of the model fit to the data depends on the number of the finite elements used. In (Terzopoulos and Metaxas, 1991; Metaxas and Terzopoulos, 1993) it was assumed that the sampling density of the finite element grid remained constant throughout the model fitting process. Clearly this poses a significant limitation in case of model fitting applications where the user assumes no prior knowledge of the complexity of the given data.

In this paper we generalize the shape representation approach presented in (Metaxas and Terzopoulos, 1993). The algorithm first does a coarse model fit, solving for a first approximation to the translation, rotation and global deformation parameters and then does several passes of mesh refinement, by locally subdividing triangles based on the distance between the given datapoints and the model. In this way with a resulting small number of model nodes we can very efficiently and accurately represent the shape of an object at various levels of detail. In order to further improve the fit of the model to datapoints we also modify our previously developed force assignment algorithm (Terzopoulos and Metaxas, 1991) so that each datapoint gets assigned to a point within a finite element whose distance from the datapoint is minimum, instead of to a model node. Once the force is assigned, it is appropriately distributed to the nodes of the corresponding finite element according to the finite element theory.

The local subdivision algorithm utilizes the properties of triangles bisected by the median that corresponds to the longest side. That is, the interior angles of the refined triangles do not go to zero as the level of subdivision goes to infinity. Also this triangulation improves the shape regularity of the subdivided triangles as the subdivision proceeds. The local subdivision algorithm can be shown to satisfy conformity, non-degeneracy and smoothness which are desirable properties for finite element meshes and ensure the accuracy of the solution.

The adaptive subdivision algorithm combined with the global deformations of the dynamic models, inherently allows the reconstruction, representation and abstraction of shape at various levels of detail. The shape hierarchy consists of using a superquadric with global deformations only, then global and one course level of local deformation and finally global and local deformations with various levels of local deformations extracted from our locally adaptive subdivision algorithm. This hierarchical surface detail representation is important in many computer vision and computer graphics applications, such as obstacle avoidance, object recognition and shape representation.

Using this new local subdivision algorithm, we present shape recovery experiments from range data sets which run at interactive rates on an Iris Crimson workstation with VGX graphics. In this paper we assume that there are no outliers and significant noise in the data. That problem was addressed in (Metaxas and Terzopoulos, 1993). Our current formulation naturally allows the use of methods such as Kalman filtering that were employed in (Metaxas and Terzopoulos, 1993). The technique we present allows accurate and efficient multi-level shape representation based on models with significantly fewer nodes than the given datapoints.

2. Overview

Section 3 presents previous research related to the new technique. Section 4 reviews the formulation of deformable models. Section 5 presents the locally adaptive subdivision algorithm which also includes the description of the finite elements used and our new force assignment technique. Section 6 gives a summary of the model fitting algorithm, and finally Section 7 presents experimental results that demonstrate the recovery of object shape from 3D range data using our deformable models.

3. Related Work

Most of the current shape recovery algorithms which use surface models, assume fixed-size grids (Terzopoulos et al., 1988; Pentland and Sclaroff, 1991; Terzopoulos and Metaxas, 1988; Metaxas, 1992; Cohen and Cohen, 1991; Cohen et al., 1992; Delingette et al., 1992; Delingette et al., 1993). Vasilescu and Terzopoulos (1992) proposed a technique for adaptive subdivision of meshes consisting of nodal masses interconnected by adjustable springs. In case of local subdivision of the mesh, a computationally expensive constraint procedure has to be applied to ensure that the triangular structure of the mesh is maintained. McInerney and Terzopoulos (1993) developed a deformable balloon model for shape estimation and tracking with uniform refinement capabilities. Huang and Goldgof (1992) present a geometric adaptive subdivision algorithm for nonrigid motion analysis which uses planar triangular patches. Tanaka and Kishino (1993)

develop a geometric adaptive mesh generation algorithm for surface reconstruction. Even though the algorithm supports local subdivision, it does not use a 3D model, and in order to guarantee a smooth solution special algorithms need to be employed to deal with cracks often occurring during subdivision. Delingette (1994) uses simplex meshes for reconstructing the geometry of complex objects from range data. The mesh is refined based on its curvature, its distance from the 3D data and its elongation. As opposed to our proposed method, this method also allows changes in the topology of the mesh, but the process requires manual intervention. This approach is most closely related to ours (see also Koh et al., 1994), but it is mostly suited for smoothly varying objects due to the use of curvature information. This method does not use global deformations which allow an additional level of shape abstraction, and was designed primarily for shape reconstruction from relatively dense data. Motivated by the work of Metaxas and Terzopoulos (1991) on deformable superquadrics with local and global deformations, Vemuri and Radisavljevic (1993) developed a probabilistic wavelet-based hierarchical representation for the local deformations defined in (Terzopoulos and Metaxas, 1991), where a uniform non-adaptive mesh was used. As opposed to shape abstraction, their main motivation is the use of this representation scheme as prior information in a probabilistic framework for efficient surface reconstruction. However, the method is most useful when used for the estimation of classes of objects for which training data is available from objects within that particular class.

In computer graphics there have been many attempts to develop techniques for surface reconstruction (Schmitt et al., 1986; Hoppe et al., 1994; Bajaj et al., 1995). The results are impressive, but the emphasis of the work is on accurate shape reconstruction of complete and dense data with very small amounts of noise, as opposed to a multi-level shape representation with abstraction. Schmitt et al. (1986) present an adaptive subdivision method for object reconstruction from range data. The method is restricted to surfaces or rectangular topology and assumes dense data. Hoppe et al. (1994) presented a method for the recovery of surfaces of variable topology with sharp features such as creases and corners. The method does not offer multiresolution shape representation and the authors have not yet shown its use in case of sparse range data with significant amount of noise and/or outliers.

Contrary to many of the above techniques, our proposed method provides an efficient and intuitive way for representing and abstracting shape at various levels of detail and can be applied to both sparse and dense range data. In this paper we do not address the issue of shape reconstruction and abstraction for the case of surfaces of arbitrary topology. We have addressed this issue in (DeCarlo and Metaxas, 1994, 1995). The method presented can be used in conjunction with these methods, since in other work of ours (DeCarlo and Metaxas, 1994; DeCarlo and Metaxas, 1995) we emphasized primarily the issue of shape abstraction based on deformable model blending and the use of only global deformations. We plan to unify these methods in the future, together with our work on automatic adaptation of the elastic parameters of a deformable model (Metaxas and Kakadiaris, 1996) to further improve the shape estimation results.

4. Deformable Models: Geometry, Kinematics, Dynamics

In this section we briefly review the general formulation of deformable models; further detail can be found in (Terzopoulos and Metaxas, 1991; Metaxas, 1992; Metaxas and Terzopoulos, 1993).

Geometrically, the models used in this paper are closed surfaces in space whose intrinsic (material) coordinates are $\mathbf{u} = (u, v)$, defined on a domain Ω. The positions $\mathbf{x}(\mathbf{u}, t)$ of points on the model relative to an inertial frame of reference Φ in space are given by

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

where $\mathbf{c}(t)$ is the origin of the model frame, ϕ, and $\mathbf{R}(t)$ is the rotation matrix expressing the orientation of ϕ. $\mathbf{p}(\mathbf{u}, t)$ denotes the positions of points on the model relative to the model frame. To introduce global and local deformations, we further express $\mathbf{p}$ as the sum of a reference shape $\mathbf{s}(\mathbf{u}, t)$ and a displacement function $\mathbf{d}(\mathbf{u}, t)$, i.e., $\mathbf{p} = \mathbf{s} + \mathbf{d}$. Global deformation parameters, $\mathbf{q}_s$, are used to define $\mathbf{s}$, while local deformation parameters, $\mathbf{q}_d$, are used to define $\mathbf{d}$.

For the applications in this paper, we define $\mathbf{s}$ as a superquadric ellipsoid that can undergo a tapering deformation in the x and y model frame axes, and a bending deformation in the x model axis (see Terzopoulos and Metaxas, 1991; Metaxas, 1992 for formulas). We then collect the global parameters used to define $\mathbf{s}$, into the vector $\mathbf{q}_s = (a, a_1, a_2, a_3, \epsilon_1, \epsilon_2, t_1, t_2, b_1, b_2, b_3)^T$, where $a \geq 0$ is a scale parameter, $0 \leq a_1, a_2, a_3 \leq 1$ are aspect ratio parameters, ϵ_1, $\epsilon_2 \geq 0$ are the superquadric "squareness" parameters,

$-1 \leq t_1, t_2 \leq 1$ are the tapering parameters in the x and y axes, respectively, b_1 defines the magnitude of the bending and can be positive or negative, $-1 \leq b_2 \leq 1$ defines the location on axis z where bending is applied and $0 < b_3 \leq 1$ defines the region of influence of bending. Our method of incorporating global deformations is not restricted to only tapering and bending deformations. Any other deformation that can be expressed as a continuous parameterized function can be incorporated similarly.

Local displacements $\mathbf{d}$ are computed based on the use of triangular finite elements (Terzopoulos and Metaxas, 1991; Metaxas, 1992). Associated with every finite element node i is a nodal vector variable $\mathbf{q}_{d,i}$. We collect all the nodal variables into a vector of local degrees of freedom $\mathbf{q}_d = (\ldots, \mathbf{q}_{d,i}^T, \ldots)^T$, and we compute the local displacement $\mathbf{d}$ based on the finite element theory as $\mathbf{d} = \mathbf{S}\mathbf{q}_d$. $\mathbf{S}$ is the shape matrix whose entries are the finite element shape functions. In a subsequent section, we will give more details about the type of finite elements and shape functions we use.

The velocity of points on the model is given by Metaxas (1992),

$$\dot{\mathbf{x}} = \mathbf{L}\dot{\mathbf{q}}, \tag{2}$$

where $\mathbf{L}$ is a Jacobian matrix, and $\mathbf{q} = (\mathbf{q}_c^T, \mathbf{q}_\theta^T, \mathbf{q}_s^T, \mathbf{q}_d^T)^T$, with $\mathbf{q}_c = \mathbf{c}$ and $\mathbf{q}_\theta$ is the model's rotational degrees of freedom expressed as a quaternion.

Our goal when fitting the model to visual data is to recover the vector $\mathbf{q}$ which expresses the model's degrees of freedom. The model fitting procedure is done in a physics-based way by enabling the data to apply traction forces to the surface of the model (Terzopoulos and Metaxas, 1991). Based on Lagrangian dynamics we make our model dynamic in $\mathbf{q}$, and we arrive at the following first order set of motion equations

$$\mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{f}_q, \tag{3}$$

where $\mathbf{D}$ and $\mathbf{K}$ are the damping and stiffness matrices, respectively, (see Metaxas, 1992 for their definitions), and where

$$\mathbf{f}_q(\mathbf{u}, t) = \int \mathbf{L}^T \mathbf{f} \tag{4}$$

are the generalized external forces computed from the 3D forces, $\mathbf{f}$, that the data exert on the model (their computation will be given in a later section). Equation (3) yields a model that has no inertia and comes to rest as soon as all the applied forces equilibrate or vanish. We also decouple the equations by assuming that $\mathbf{D}$ is diagonal and constant over time. Note that we never assemble the finite element stiffness matrix, but compute $\mathbf{K}\mathbf{q}$ in an element-by-element fashion (Metaxas, 1992).

5. Locally Adaptive Finite Elements

In this section we will describe the local strain energy and the finite elements used, the algorithm for assigning forces from datapoints to points on the model, the criterion to locally select elements for subdivision and the local finite element subdivision algorithm. Based on this algorithm we can both accurately and efficiently represent shape. Without prior knowledge of the complexity of the given data, new nodes are added in order to minimize the error of fit of the model to the data. The experiments we present in a later section clearly demonstrate the limitations of a uniform static mesh in terms of the accuracy and efficiency of shape estimation. Another disadvantage of using a uniform grid is that the tessellation produces triangles whose orientation is uniform and does not necessarily match the complexity of the surface to fit. The adaptive subdivision algorithm overcomes this problem. In addition, the process of model fitting is now automated, since the user does not have to experiment with a variety of mesh sizes before an acceptable fit can be achieved.

5.1. *Finite Elements with C^1 Continuity*

For the applications in this paper we select a strain energy and the appropriate finite elements that guarantee C^1 continuity. In particular we use appropriate finite element shape functions defined in (Dhatt and Touzot, 1984) and also used in (McInerney and Terzopoulos, 1993). A thin plate under tension deformation energy, suitable for C^1 continuous model surface, is given by the functional

$$\begin{aligned}\mathcal{E}_p(\mathbf{d}) = \int & w_{20}\left(\frac{\partial^2 \mathbf{d}}{\partial u^2}\right)^2 + w_{11}\left(\frac{\partial^2 \mathbf{d}}{\partial u \partial v}\right)^2 \\ & + w_{02}\left(\frac{\partial^2 \mathbf{d}}{\partial v^2}\right)^2 + w_{10}\left(\frac{\partial \mathbf{d}}{\partial u}\right)^2 \\ & + w_{01}\left(\frac{\partial \mathbf{d}}{\partial v}\right)^2 + w_{00}\mathbf{d}^2 \, d\mathbf{u}.\end{aligned} \tag{5}$$

The nonnegative weighting functions w_{ij} control the elasticity of the material. Increasing w_{01} and w_{10} makes

the deformations have more membrane properties, while increasing the w_{20}, w_{11} and w_{02}, the deformations behave more like a thin plate. In our implementation, however, we reduce these functions to scalar stiffness parameters $w_{ij}(\mathbf{u}) = w_{ij}$.

The deformable models we use in this paper are topologically isomorphic to a sphere and can be mapped to a rectangular material coordinate space $\mathbf{u} = (u, v)$ (Metaxas and Terzopoulos, 1993; Metaxas, 1992). We discretize the deformable model in the rectangular material coordinate domain and define the triangular elements (Metaxas and Terzopoulos, 1993). Every triangular finite element has three nodes and in every node i, we store the vector of nodal variables

$$\mathbf{q}_{d,i} = \left(\mathbf{d}_i^T, \mathbf{d}_{u,i}^T, \mathbf{d}_{v,i}^T, \mathbf{d}_{uu,i}^T, \mathbf{d}_{uv,i}^T, \mathbf{d}_{vv,i}^T\right)^T, \quad (6)$$

where $\mathbf{d}_i$ is the vector nodal displacement, and subscripts u, v denote partial differentiation w.r.t to the u and v directions of the material coordinate space, respectively. From (6) it is clear that every triangular element has 18 vector degrees of freedom.

To approximate the above strain energy (5) we use triangular finite elements whose 18 shape functions, N_i (each corresponds to one of the 18 vector degrees of freedom of the element) are defined in (Dhatt and Touzot, 1984) and guarantee the C^1 continuity of the solution. The importance of these elements is that they do not impose any restrictions on the triangular grid, which is necessary due to the adaptive and irregular nature of our tessellation algorithm. Figure 1 shows such a triangular element in an arbitrary orientation with respect to the material coordinate space, $\mathbf{u}$, and where its local coordinate system $\xi\eta$ is also illustrated. The 18 nodal shape functions $\mathbf{N}_i(\xi, \eta)$ are defined in

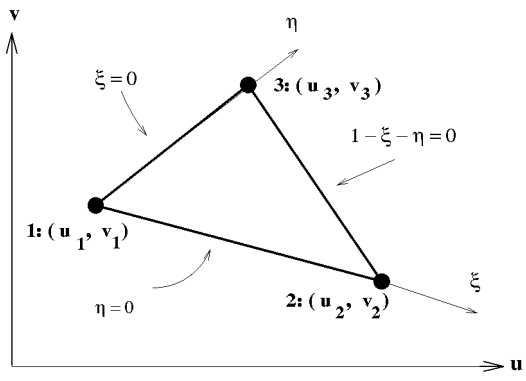

Figure 1. C^1 Continuous triangular element. The three nodes are numbered.

(Dhatt and Touzot, 1984; Metaxas, 1992; McInerney and Terzopoulos, 1993). The relationship between the uv and $\xi\eta$ coordinates is

$$\begin{aligned} u &= (1 - \xi - \eta)u_3 + \xi u_1 + \eta u_2 \\ v &= (1 - \xi - \eta)v_3 + \xi v_1 + \eta v_2, \end{aligned} \quad (7)$$

where (u_i, v_i) are the material coordinates at the nodes of the triangular element.

Using the above finite elements, the corresponding shape functions, and (7), we compute from the strain energy (5) the stiffness matrix $\mathbf{K}$ through a technique based on the theory of elasticity (Metaxas, 1992; Metaxas and Terzopoulos, 1993) and demonstrated for the case of a loaded membrane deformation energy in (Metaxas and Terzopoulos, 1993).

5.2. *Force Assignment*

In our applications, for each given range datapoint $\mathbf{z}$ we want to find a point on the model with material coordinates $\mathbf{u}_z$ that minimizes the distance d ($d(\mathbf{u}) = \|\mathbf{z} - \mathbf{x}(\mathbf{u})\|$) between $\mathbf{z}$ and the model. A brute-force approach to the above minimization problem that worked well in our previous efforts (Terzopoulos and Metaxas, 1991; Metaxas, 1992) is to select from all the model nodes the one that minimizes d. The above approach is inadequate for our local finite element subdivision algorithm since it only takes into account model nodes in computing $d(\mathbf{u})$. In this paper we will use the following algorithm which is the most accurate possible, given that we approximate the model surface with finite elements and there is no analytic formula for $\mathbf{x}(\mathbf{u})$.

Algorithm. We perform a minimization over all finite elements to compute a point (with material coordinates $\mathbf{u}$) within a finite element j, that minimizes the Euclidean distance of datapoint $\mathbf{z}$ to the deformable model, i.e.,

$$\begin{aligned} d(\mathbf{u}) &= \min_j \; d(\mathbf{z}, j) = \min_{j,\mathbf{u}} \|\mathbf{z} - \mathbf{x}^j(\mathbf{u})\| \\ &= \min_{j,\mathbf{u}} \left\|\mathbf{z} - \left(\mathbf{c} + \mathbf{RS}(\mathbf{u})\mathbf{q}_d^j\right)\right\|, \end{aligned} \quad (8)$$

where $\mathbf{S}(\mathbf{u})$ is the shape function matrix whose entries are the element shape functions (Dhatt and Touzot, 1984) and $\mathbf{q}_d^j$ are the element's nodal degrees of freedom. From this minimization we select the model point

with material coordinates $\mathbf{u}_z$ with minimum distance $d(\mathbf{u}_z)$ from the datapoint. The complexity of the algorithm is $O(mn)$, where m is the number of finite elements used and n is the number of given datapoints. The complexity of the algorithm can be further reduced to $O(n)$, if we keep track of the history of datapoint assignment to the finite elements from one step of the algorithm to the next. After the initial fit of the model to the given data with only global deformations, its pose does not change significantly. Therefore only a local search in the neighborhood of the finite element to which the datapoint was assigned in the previous step, is necessary. However, in cases where the model surface shape changes significantly over time this faster algorithm can fail.

We then assign to $\mathbf{u}_z$ the following force

$$\mathbf{f}(\mathbf{u}_z) = \beta(\mathbf{z} - \mathbf{x}(\mathbf{u}_z)), \tag{9}$$

based on the separation between the datapoint $\mathbf{z}$ in space and the force's point of influence $\mathbf{u}_z$ on the model's surface and where β is the strength of the force given as an input parameter. We then extrapolate $\mathbf{f}(\mathbf{u}_z)$ to the element nodes using the formula

$$\mathbf{f}_i = N_i(\mathbf{u}_z)\,\mathbf{f}(\mathbf{u}_z), \tag{10}$$

where N_i is the shape function that corresponds to the displacement, $\mathbf{d}_i$, of node i and $\mathbf{f}_i$ is the extrapolated value of $\mathbf{f}(\mathbf{u}_z)$ to node i (see Fig. 2). We then use (4) to compute the corresponding generalized forces which we incorporate in (3) to estimate the model's parameters.

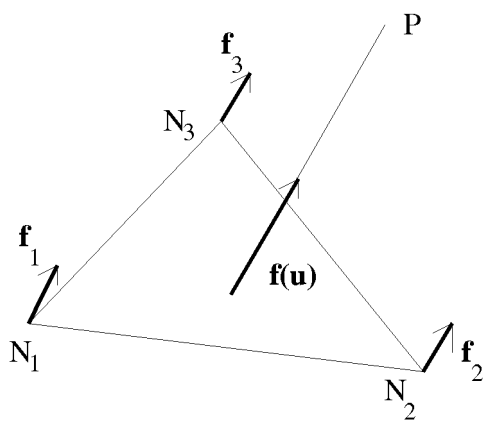

Figure 2. Extrapolation of force to the element nodes.

5.3. *Criterion for Finite Element Subdivision*

We use a criterion based on the distance from the datapoints to the finite elements, to decide whether an element or elements should be subdivided. We first compute using the above algorithm the point $\mathbf{u}_z$ on the model whose distance $d(\mathbf{u}_z)$ is the minimum from the given datapoint $\mathbf{z}$. If

$$d(\mathbf{u}_z) > \tau_d, \tag{11}$$

where τ_d is a threshold, we subdivide the elements that this nearest model point is on. We distinguish the following three cases.

1. If $\mathbf{u}_z$ lies inside an element, then the element is selected for subdivision.
2. If $\mathbf{u}_z$ lies on an edge, then the two adjacent elements to the edge are subdivided.
3. If $\mathbf{u}_z$ is a model node, then all the elements adjacent to the node are subdivided.

Once the above criterion is satisfied we subdivide the chosen elements and apply the following subdivision algorithm to ensure that the resulting grid has properties necessary for the application of the finite element method. It is worth mentioning that the curvature calculation is very sensitive to noise, and since we want to use our technique in case of sparse data, we did not consider using the data curvature as a criterion for subdivision.

5.4. *Subdivision Algorithm*

Our subdivision algorithm has the following two basic steps and is an adaptation for computer vision applications of the adaptive finite element technique developed by Rivara (1984). The first is a *bisection* operation in which a single finite element is subdivided according to a certain rule. The second is a *conforming* recursive operation which ensures that the rest of the finite elements satisfy properties necessary to apply the finite element method. We will now describe in detail each of these operations. The subdivision algorithm is employed in the material coordinate domain of the model which for shapes of genus 0 can be mapped to a rectangle (Metaxas and Terzopoulos, 1993) whose two out of the four sides are identical. However, during the application of the algorithm we treat them as being distinct

and we remedy any node discrepancies on the identical sides based on the approach we will explain later.

Step 1: Bisection Operation. If the above defined distance criterion is met for a particular finite element, we perform a *bisection operation* as follows:

- Let T be a triangle with vertices A, B, and C; We subdivide the triangle T into two triangles by bisecting it along its longest edge (the length is defined in 3D). Let AB be the longest edge of T, and D the midpoint of AB. Then T is subdivided into two triangles, ADC and BCD as shown in Fig. 3(a).

This subdivision has been shown to provide properties desirable for use in finite element applications. First, none of the elements will become 'skinny' (triangles with very acute angles) as the level of subdivision increases. Rosenberg and Stenger (1975) have proved that if α_i is the smallest angle of the triangulation obtained by the ith iterative subdivision, then $\alpha_i \geq \frac{\alpha_0}{2}$ for any i, where α_0 is the smallest interior angle of the initial triangulation. Second, the subdivision improves the shape regularity of the triangles that is, the ratio between the longest and shortest sides of the triangles gets smaller as the level of subdivision increases (Stynes, 1980), in case of triangles with large such ratios.

Step 2: Conforming Operation. The second part of the algorithm ensures that the resulting finite element grid generates properties necessary for the application of the finite element method. A triangulation is defined to be conforming if any two adjacent triangles must share either a common vertex or a common edge (Zienkiewicz, 1977). In Fig. 3(b), the triangulation is not conforming, because conformity is violated between T_1 and T, and between T_2 and T. In the finite element method, we must maintain the continuity across inter-element boundaries, i.e., it is necessary to maintain the conformity of the triangulation.

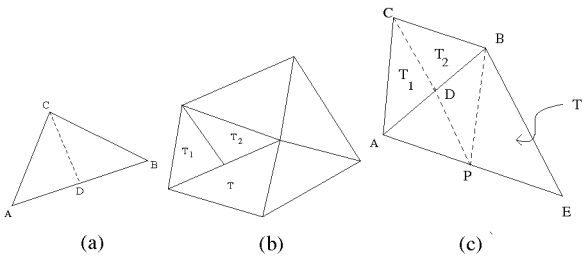

Figure 3. Various subdivision examples. (a) subdivision of a triangle by the longest edge, (b) an example of non-conforming triangulation, (c) an illustration of conforming operation.

In Fig. 3(c) we demonstrate how to address this problem. If we introduce a new node, D, as a result of bisecting element ABC, the element, T, adjacent to the subdivided edge AB becomes non-conforming. In order to ensure conformity, further subdivision must be performed on T along the common edge with midpoint D. However, it is possible that the common edge may not be the longest edge of T. Therefore, this subdivision will cause the triangulation to lose the aforementioned properties of shape regularity. To remedy this problem, we take the following approach in subdividing element T as shown in Fig. 3(c). We first bisect T by its longest edge, AE, at its midpoint, P. If AE is the common edge, then we stop subdividing. Otherwise, we further subdivide T by connecting P to the midpoint D of AB. As a result of this process, conformity is preserved and the subdivision will not produce 'skinny' triangles. This process is called a *conforming operation*.

In our local subdivision algorithm, the conforming operation is performed whenever subdivision of an element causes non-conformity. The conforming operation, however, may create new non-conformity. In order to ensure the conformity, this conforming operation is recursively applied until the triangulation becomes entirely conforming. This recursive process is guaranteed to stop because there is only a limited number of triangles to start with. Figure 4 illustrates an example of applying a series of conforming operations which were necessary because of propagating

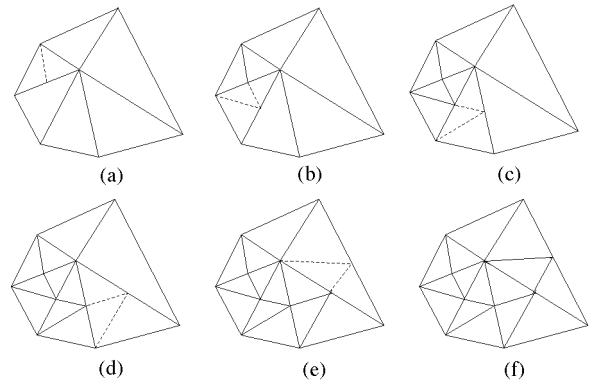

Figure 4. An example of recursive local subdivision in material coordinate space.

non-conformity. This example captures shapes that are topologically equivalent to a sphere (these are the types of objects whose shape we estimate in this paper) since they can be mapped in material coordinates to a polygon. The simplest type of polygon to define such objects is a rectangle whose two of the four sides are identical (edges AE and $A'E'$ in Fig. 5) and we have used in the past (Metaxas and Terzopoulos, 1993; Metaxas, 1992). As mentioned earlier, the triangulation is performed in the material coordinate space and the two corresponding sides AE and $A'E'$ of the rectangle are treated as non-corresponding when the algorithm is applied. Therefore the algorithm is guaranteed to terminate. After the termination of the recursive conforming operation (Fig. 5(a)), all nodes on the two identical sides of the rectangle are checked so that an equal number of them exists on each of the corresponding sides AE and $A'E'$, and are also in the same location. Wherever there is a discrepancy (e.g., node B in Fig. 5(a) has no corresponding node on side $A'E'$), it is automatically corrected by the introduction of new nodes (e.g., node B' on side $A'E'$ is introduced in Fig. 5(b)) at the correct locations on each of the corresponding sides of the rectangle. We also adjust locally the triangulation (e.g., introduction of the two triangles $OA'B'$ and $0B'C'$ to replace $OA'C'$ in Fig. 5(b)) to ensure that all finite elements share the same nodes.

Algorithm. Let the *subdivision set* be the set of finite elements which have been chosen for subdivision, but have not yet been subdivided. Based on the above two operations, our local subdivision algorithm can be described as follows:

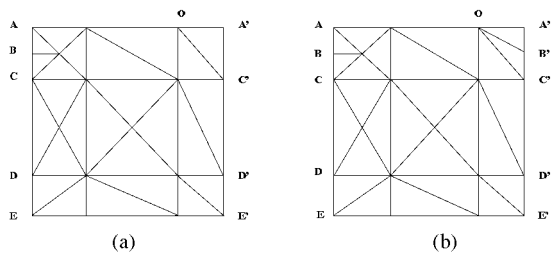

Figure 5. An example of how node discrepancy is accounted at the end of the recursive subdivision in case of objects that are topologically equivalent to a sphere. In this example sides AE and $A'E'$ are identical and should have the same number of nodes. (a) shows the result of recursive subdivision where node B has no corresponding node B'. (b) shows the correction to this discrepancy through the introduction of node B' and the two triangles $OA'B'$ and $OB'C'$ to replace $OA'C'$.

While the subdivision set is not empty
BEGIN
  Set T as a triangle from the subdivision set.
  Remove T from the subdivision set.
  If T has not been bisected then
  BEGIN
    Set E_{longest} as the longest edge of T.
    Subdivide E_{longest}.
    Bisect T by E_{longest}.
    Set T' as the adjacent triangle of T
    by the edge E_{longest}.
    Conform(T', E_{longest}).
  END
END

The conforming operation, Conform(T', E'), is described in pseudocode as follows:

Set E'_{longest} as the longest edge of T'
if E'_{longest} is the same as E' then
BEGIN
  Bisect T' by E'.
  Return.
END
Subdivide E'_{longest}.
Bisect T' by E'_{longest}.
Set $\hat{T}$ as one of the sub-triangles from the
previous step which contains the edge E'.
Bisect $\hat{T}$ by E'.
Set $\tilde{T}$ as the adjacent triangle of T' by the
edge E'_{longest}.
Conform($\tilde{T}$, E_{longest}).

In summary, our local subdivision algorithm satisfies several desirable properties for finite element mesh generation. They are: 1) conformity: any two adjacent elements share only either a node or an edge; 2) non-degeneracy: the triangulation maintains the shape regularity of the refined elements, i.e., the elements do not become 'skinny'; and, 3) smoothness: there is no abrupt size difference between adjacent elements, and hence the transition between small and large elements is smooth.

6. Summary of Model Fitting to Range Data

In the previous sections we defined the translation, rotation, global and local degrees of freedom, and the adaptive finite element subdivision algorithm. Based

on the above formulations our model fitting algorithm has the following steps:

1. Initialize the model to the data by placing the model's model frame to the center of mass of the data and estimate its orientation based on the matrix of central moments (Metaxas and Terzopoulos, 1993).
2. Fit the model to the given data by using only the translation, rotation and global degrees of freedom.
3. Fit the model to the data using both global and local (do not apply any subdivision) deformations.
4. While the error of fit between the model and the data is beyond a user specified threshold, apply the recursive finite element subdivision algorithm to the respective locations. At the end of every subdivision level (the end of each recursion step), fit the model to the data using both global and local deformations to further refine the fit.
5. Repeat Step 4 until the error of fit is everywhere below the user specified threshold.

It is very important to mention that the fitting of the model to the given data is done automatically. The user only has to specify a threshold for terminating the fitting of the model to the data, and the strength of the forces. The threshold is defined as a percentage of the longest dimension of the data, which is computed based on the use of the matrix of central moments (Metaxas, 1992).

7. Experiments

We have carried out various experiments to test our locally adaptive finite element algorithm. These include 3D range data taken from the the Cyberware® 3D digitizer, and a human head model provided by Viewpoint®. Using our new force assignment algorithm the models deform globally and locally and subdivide locally to fit the given data. Our experiments run at interactive rates on an R4000 Iris Crimson workstation with VGX graphics. In all the experiments we used a time step size equal to 10^{-5} (iterating with the Euler method) and a unit damping matrix $\mathbf{D}$, while the threshold for local subdivision was roughly $\tau_d = 0.1\%$ of the biggest dimension of the given data. In addition, the force strength is defined by the user. In every experiment we compute the model's error of fit with respect to the input data. The error of fit is defined in terms of the distances of the fitted deformable model's finite elements from the original input data (we normalize it with respect to the model's biggest dimension). The fitted model is obtained by fitting the deformable model to the input data until the movement of each element becomes relatively negligible. This is achieved by measuring the difference between two positions of each element obtained from two consecutive iterations during fitting, and we allow the model to continue fitting until the maximum relative difference in the position of the model elements is less than a threshold value of the order of 10^{-5}. Even though we did not experience them, there are cases where the fitting process can become unstable. In such cases an adaptive integration algorithm (e.g., adaptive Runge-Kutta) solves this problem. In addition, we initialize the model based on the center of mass of the data and the matrix of central moments, and we first fit the model to the data using only the global deformations.

In the first experiment, a deformable model with 66 nodes was fitted to 857 3D datapoints sampled from a biomedical image of the left part of a human lung (Fig. 6(a)). The initial fitting model was an ellipsoid

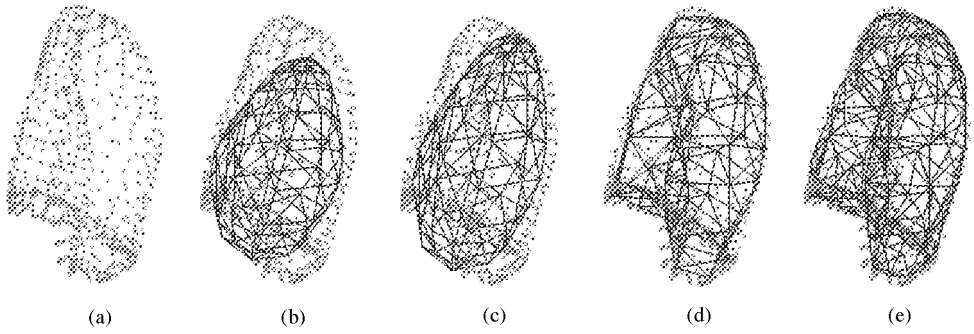

Figure 6. Fitting of model to data of a human lung. (a) input data of a human lung (left part), (b) model initialization, (c) model fitted to data with only global deformations, (d) model fitted to data with global and local deformations, (e) model fitted to data after three levels of local subdivision.

Table 1. Error statistics from fitting the model to 857 data points of a human lung. The second column shows statistics from fitting the initial model with 66 nodes to the data. The third, fourth and fifth columns represent statistics after the first, second and third levels of grid subdivision, respectively. The sixth and seventh columns represent fitting results in case of non-adaptive grids with a constant number of nodes.

Model	Initial fit	Sub-1	Sub-2	Sub-3	No sub #1	No sub #2
# of nodes	66	129	168	243	256	627
# of elements	128	254	332	482	512	1250
# of iterations	114	169	212	373	209	222
# CPU time in mins.	1	2.9	3.8	5.5	3	6
$\text{Mean}_{\text{error}}$	0.081739	0.023317	0.013317	0.008100	0.019040	0.011834
$\text{Max}_{\text{error}}$	1.351201	0.371757	0.318139	0.104073	0.281506	0.247791
σ_{error}	0.162214	0.038362	0.017991	0.010340	0.034987	0.026694
σ^2_{error}	0.026313	0.001472	0.000324	0.000107	0.001224	0.000713

(Fig. 6(b)). Figure 6(c) shows the model after global deformations. Figure 6(d) shows the model after local deformations. Figure 6(e) shows the model with 243 nodes after three levels of our locally adaptive subdivision. Comparison of Figs. 6(d) and (e) shows that the subdivision was concentrated in the boundary area with more complex geometry. The fitting process took approximately five and a half minutes and 400 iterations.

Table 1 shows error statistics collected from the hierarchical fitting of a deformable model to data from a human lung. In the rows we present for each level of shape detail the number of nodes in the model, the number of elements in the model, the number of iterations, the CPU time in minutes, the mean error value, the maximum error value, the standard deviation and the variance of the error of fit. Then we subdivided the model elements as explained in the previous sections. We started fitting the model (Initial Fit column in the table) with 66 nodes and 128 elements while the number of input data points was 857. At the first level of subdivision (Sub-1 column), 87 elements were selected for subdivision, and the model resulted in a total of 129 nodes and 254 elements. At the second level of subdivision (Sub-2 column), 50 elements were selected and the subdivision process produced a total of 168 nodes and 332 elements. At the third level (Sub-3 column), 77 elements were selected, and the subdivision process generated a total of 243 nodes and 482 elements.

We also fitted a model with a regular constant grid whose number of elements is comparable to the number of elements of the model after three levels of local subdivision and measured the error of fit. This model contained 256 nodes and 512 elements. As shown in the table, the mean and maximum errors of fit derived using this model notably exceed those of the subdivided model. In the last column of Table 1 we also show the error of fit obtained by fitting a model with a constant number of 627 nodes and 1250 elements. From this last example it is apparent that our locally adaptive subdivision algorithm significantly improves the quality of model fitting.

In the second experiment we use range datapoints obtained from the Cyberware, Inc., 3D digitizer. In Fig. 7

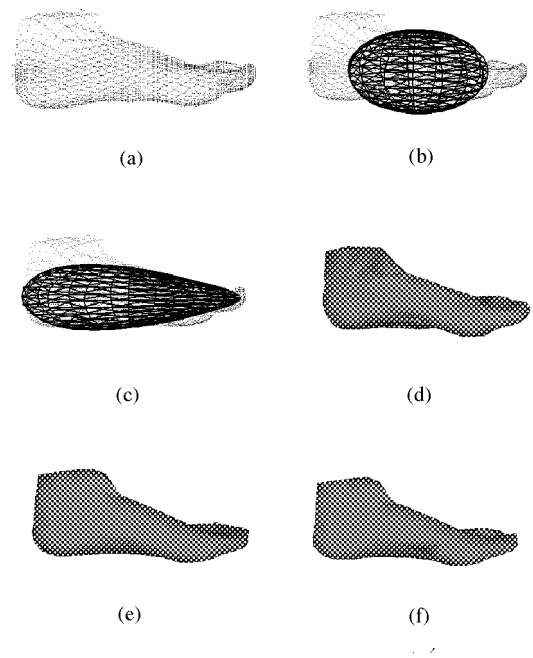

Figure 7. Fitting of model to foot data. (a) foot data, (b) model initialization, (c) intermediate step of model fitting to the data with apparent global deformations, (d) model fitted to data without local subdivision, (e) model fitted to data after one level of local subdivision, (f) model fitted to data after four levels of local subdivision.

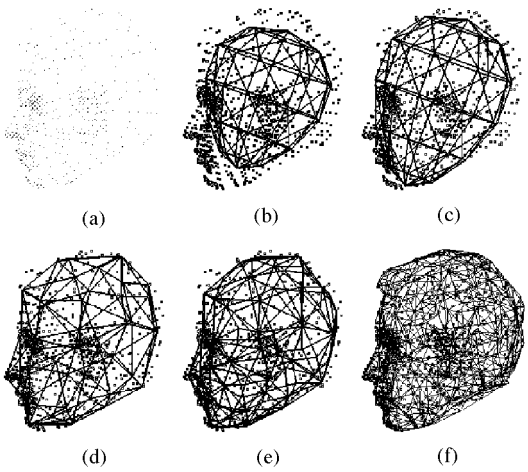

Figure 8. Fitting of the model to input data of a human head. (a) input data, (b) the initial model, (c) the model after global deformations, (d) the model after local deformations, (e) the model after two levels of subdivision, (f) the model after four levels of subdivision.

we fit a deformable model with initially 227 nodes, to 3825 3D range datapoints obtained from a mannequin foot. The local deformation stiffness parameters of the model were $w_{00} = 0.5$, $w_{01} = 0.5$, $w_{10} = 0.5$, $w_{02} = 0.1$, $w_{11} = 0.1$ and $w_{20} = 0.1$. The initial model was an ellipsoid ($\mathbf{q}_s = (2.3,\ 0.3, 0.5, 0.3,\ 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0)^T$) and the force strength parameter was $\beta = 20.0$. Figure 7(a) shows the given foot data. Figure 7(b) shows a view of the range data and the initial model, while Fig. 7(c) shows an intermediate step in the fitting of the model to the data where the global deformations are apparent. Figure 7(d) shows the model fitted to the data without local subdivision, Fig. 7(e) shows the model fitted to the data after one level of local subdivision, while Fig. 7(f) shows the final model fitted to the data after four levels of local subdivision. After six and a half minutes and a total of 452 iterations, the new final number of model nodes is 640, which is significantly smaller than the number of given datapoints.

In the next experiment we fit a deformable model with 47 nodes to 1269 3D data points defining a human head. The input data were obtained from the Viewpoint, Inc. The local deformation stiffness parameters of the model were $\omega_{ij} = 0.05$. The initial model was an ellipsoid ($\mathbf{q}_s = (7.0, 0.5, 0.5, 0.6, 1.0, 1.0, 0.0, -0.3, 0.0, 0.0, 1.0)^T$) and the force strength parameter was $\beta = 1.0$. Figure 8(a) shows a view of the range data. Figure 8(b) shows the initial model. Figure 8(c) shows the model after global deformations. Figure 8(d) shows the model after local deformations. Figure 8(e) shows the intermediate model after two levels of local subdivision. Figure 8(f) shows the final model after four levels of local subdivision that took five minutes and a total of 283 iterations.

Figures 9(a) and (b), respectively, show a front view and a back view of human body figures displayed at three different levels of detail. The human body figure consists of 15 parts: head, torso, lower torso, 3 parts for each arm, 3 parts for each leg. For coarser levels of detail, approximations of each body part were obtained as described in the previous experiments. The numbers of polygons used at each level of representation were 18155, 7292, and 2260, respectively, and the numbers of nodes were 18005, 3696, and 1180, respectively.

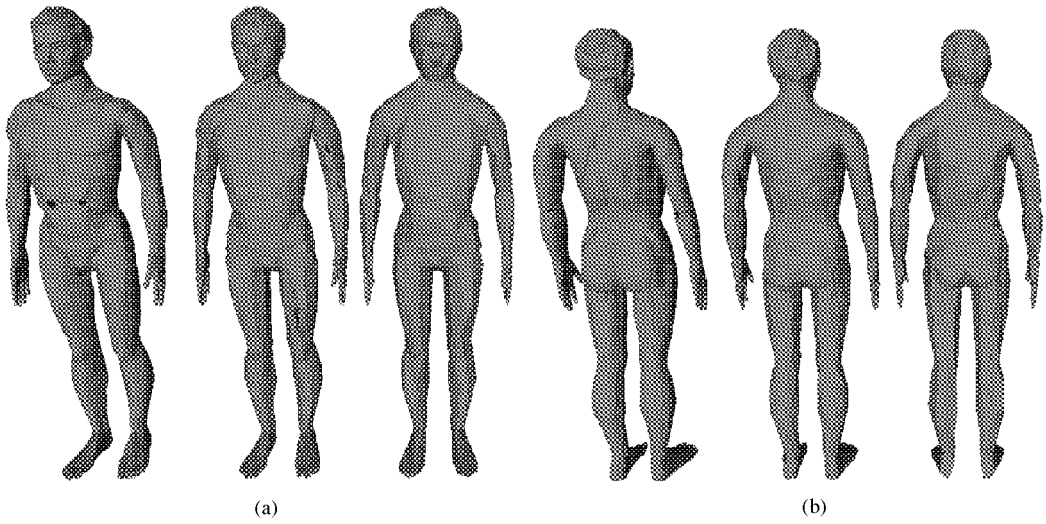

Figure 9. A human body displayed at three different levels of detail: (a) the front view, (b) the back view.

8. Conclusion

The emphasis of the work presented has been on shape representation at multiple levels of detail. The models we used have both global deformations that allow a gross shape description, and local deformations for fine shape description. In addition, for a more complete shape abstraction, we have developed a new technique that allows the local adaptive subdivision of a deformable model's finite elements. The algorithm ensures that during subdivision the desirable finite element mesh generation properties of conformity, non-degeneracy and smoothness are maintained. In conjunction with the use of our new force assignment technique from datapoints to model points, we demonstrated that we can represent accurately and efficiently[1] an object surface starting from a small regular grid. In our technique new model nodes are added only when necessary in a local fashion and therefore it is not sensitive to the size of the initial model grid. Finally, using our new locally adaptive finite element technique and the global deformations of our models we can achieve a smooth hierarchical shape representation which is necessary in many computer vision and graphics applications. The shape coverage of our models is limited at this point to surfaces of genus 0 (surfaces with no holes).

We have developed recently (DeCarlo and Metaxas, 1994, 1995) new deformable model-based methods for the shape abstraction and representation of objects with arbitrary topology where we always initialize the deformable model to a sphere. Based on the use of blending and global deformations the sphere can automatically deform to significantly more complex objects and can also change genus. It is our intention to unify this work with the method presented in this paper, and with additional work on the automatic adaptation of the elastic parameters of a deformable model (Metaxas and Kakadiaris, 1996) to improve significantly our automated object shape abstraction and estimation capabilities.

Note

1. We can get better shape estimation results with at least 50% fewer model nodes compared with a model with no local subdivision and a regular grid.

References

Bajaj, C.L., Bernandini, F., and Xu, G. 1995. Automatic reconstruction of surfaces and scalar fields from 3D scans. *Proc. Computer Graphics (Siggraph'95)*, Los Angeles, CA, pp. 109–118.

Barr, A. 1981. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1:11–23.

Bramble, J.H., Pasciak, J.E., and Xu, J. 1991. The analysis of multigrid algorithms with nonnested spaces of noninherited quadratic forms. *Mathematics of Computation*, 56(193):1–34.

Cohen, L. and Cohen, I. 1991. Finite element methods for active contour models and balloons for 2D and 3D images. *Pattern Analysis and Machine Intelligence*, 11(19).

Cohen, I., Cohen, L., and Ayache, N. 1992. Using deformable surfaces to segment 3D images and infer differential structure. *CVGIP: Image Understanding*, 56(2):242–263.

DeCarlo, D. and Metaxas, D. 1995. Adaptive shape evolution using blending. In *Proceedings International Conference on Computer Vision*, Cambridge, MA, pp. 834–839.

DeCarlo, D. and Metaxas, D. 1996. Blended deformable models. (See also *Procs. Computer Vision and Pattern Recognition Conference'94*, Seattle, Washington, June 1994), *IEEE*, PAMI-18(4):834–839.

Delingette, H. 1994. Simplex meshes: A general representation for 3D shape reconstruction. Technical Report No. 2214, INRIA, Sophia Antipolis, France.

Delingette, H., Hebert, M., and Ikeuchi, K. 1992. Shape representation and image segmentation using deformable surfaces. *Image and Vision Computing*, 10(3):132–144.

Delingette, H., Hebert, M., and Ikeuchi, K. 1993. A spherical representation for the recognition of curved objects. *Proc. of Intern. Conf. on Computer Vision*, Berlin, Germany.

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

Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., and Stuetzle, W. 1994. Piecewise smooth surface reconstruction. *Proc. Computer Graphics (SIGGRAPH'94)*, Orlando, Florida, pp. 295–302.

Huang, W.C. and Goldgof, D. 1992. Adaptive-size physically-based models for nonrigid motion analysis. *Proc. IEEE Computer Vision and Pattern Recognition Conference (CVPR'92)*, Urbana-Champaign, Illinois, pp. 833–835.

Koh, E., Metaxas, D., and Badler, N. 1994. Hierarchical shape representation using locally adaptive finite elements. *Proc. Third European Conference on Computer Vision (ECCV'94)*, Stockholm, Sweden, pp. 441–446.

McInerney, T. and Terzopoulos, D. 1993. A finite element model for 3D shape reconstruction and non-rigid motion tracking. In *Procs. of the International Conference on Computer Vision (ICCV'93)*, Berlin, Germany, pp. 518–523.

Metaxas, D. 1992. Physics-based modeling of nonrigid objects for vision and graphics. Ph.D. thesis, Department of Computer Science, University of Toronto.

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

Metaxas, D. and Kakadiaris, I. 1996. Elastically adaptive deformable models. *Procs. European Conference on Computer Vision (ECCV'96)*, Cambridge, UK, pp. 550–559.

Pentland, A. and Sclaroff, S. 1991. Closed-form solutions for physically based shape modeling and recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):715–729.

Rivara, M.C. 1984. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. *International Journal for Numerical Methods in Engineering*, 20:745–756.

Rosenberg, I.G. and Stenger, F. 1975. A lower bound on the angles of triangles constructed by bisecting the longest side. *Math. Comp.*, 29:390–395.

Schmitt, F., Barsky, B., and Du, W.-H. 1986. An adaptive subdivision method for surface-fitting from sampled data. *Proc. Siggraph'86*, Dallas, TX, pp. 179–188.

Stynes, M. 1980. On fast convergence of the bisection method for all triangles. *Math. Comp.*, 35:1195–1201.

Tanaka, H. and Kishino, F. 1993. Adaptive mesh generation for surface reconstruction: Parallel hierarchical triangulation without discontinuities. *Proc. CVPR'93*, NY, pp. 88–94.

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

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

Vasilescu, M. and Terzopoulos, D. 1992. Adaptive meshes and shells: Irregular triangulation, discontinuities, and hierarchical subdivision. *Proc. IEEE Computer Vision and Pattern Recognition Conference (CVPR'92)*, Champaign, Illinois, pp. 829–832.

Vemuri, B.C. and Radisavljevic, A. 1993. From global to local, a continuum of shape models with fractal priors. *Proc. CVPR'93*, NY, pp. 307–313.

Zienkiewicz, O. 1977. *The Finite Element Method*. McGraw-Hill.