Modeling Surfaces of Arbitrary Topology with Dynamic Particles

Richard Szeliski[1], David Tonnesen[2], and Demetri Terzopoulos[2]

[1]Digital Equipment Corporation, Cambridge Research Lab,
One Kendall Square, Bldg. 700, Cambridge, MA 02139, szeliski@crl.dec.com
[2]Department of Computer Science, University of Toronto,
Toronto, ON, M5S 1A4, davet@dgp.toronto.edu, dt@vis.toronto.edu

Abstract

This paper develops a new approach to surface modeling and reconstruction which overcomes some important limitations of existing surface representation methods, such as their tendency to impose restrictive assumptions about object topology. The approach features two components. The first is a dynamic, self-organizing, oriented particle system which discovers topological and geometric surface structure implicit in visual data. The oriented particles evolve according to Newtonian mechanics and interact through long-range attraction forces, short-range repulsion forces, co-planarity, co-normality, and co-circularity forces. The second component is an efficient triangulation scheme which connects the particles into a continuous global surface model that is consistent with the inferred structure. We develop a flexible surface reconstruction algorithm that can compute complete, detailed, viewpoint invariant geometric surface descriptions of objects with arbitrary topology. We apply our algorithms to 3-D medical image segmentation and surface reconstruction from object silhouettes.

1 Introduction

The representation of surfaces of 3D objects has attracted significant attention in computational vision. A problem central to this endeavor is the estimation of geometric surface models from various types of visual data, including range and surface normal measurements. These surface models can be used as an intermediate representation for object recognition, to guide robotics tasks such as grasping, to segment three-dimensional volumes (e.g., in medical applications), and to integrate different visual modalities.

Many vision researchers have investigated the reconstruction of $2\frac{1}{2}$-D viewer-centered surface representations [3, 20, 4, 14]. These representations are typically based on parametric spline models with internal strain energies. Equally intense effort has gone into the development of 3-D object-centered surface representations. These include generalized cylinders [1], superquadrics [9], and triangular meshes [5], as well as their physics-based generalizations, dynamic deformable cylinders [22], spheres [6, 7], and superquadrics [21], which have internal deformation energies and can be fitted through external forces to visual data such as 2-D images or 3-D range points.

Existing surface representations have limitations—the viewer-centered methods make no attempt to represent non-visible portions of object surfaces, while the object-centered methods make strong assumptions about object topology. In this paper, we propose a new approach to surface modeling which overcomes these limitations. Our approach leads to very flexible reconstruction algorithms which are able to compute detailed geometric descriptions that are not only inherently viewpoint invariant, but more importantly, are sufficiently powerful to represent the entire surface of objects with arbitrary topologies. The algorithms can interpolate regular or scattered 3-D data acquired from an imaged object, without any a priori knowledge about the topology of the object.

The surface model and reconstruction method presented in this paper has two components. The first is a dynamic particle system which discovers topological and geometric surface structure implicit in the data. The second component is an efficient triangulation scheme which connects the particles into a continuous global surface model that is consistent with the particle structure. The evolving global model supports the automatic extension of existing surfaces with few restrictions on connectivity, the joining of surfaces to form larger continuous surfaces, and the splitting of surfaces along arbitrary discontinuities as they are detected.

The most novel feature of our approach to surface reconstruction is the use of a *molecular dynamics* simulation in which particles interact through long-range attraction forces and short-range repulsion forces. Our particle system is inspired by recent molecular dynamics models of liquids for use in graphics [23]. In these models, which have spherically symmetric potential fields, particles tend to arrange themselves into densely packed volumes.

The particles in our system are more complex, i.e., each particle represents an oriented trihedral coordinate frame. Based on these frames, we design new interaction potentials which favor locally planar or locally spherical arrangements of particles. Thus, the oriented particles support smoothness constraints similar to those inherent in the deformation energies of

1063-6919/93 $03.00 © 1993 IEEE

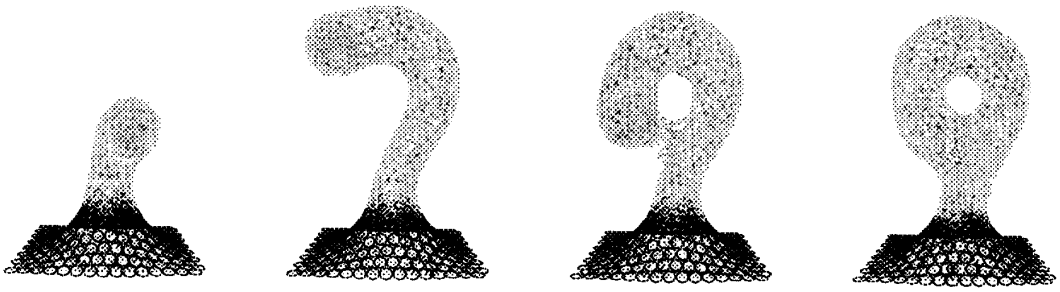

Figure 1: Forming a complex object. The initial surface is deformed upwards and then looped around. The new topology (a handle) is created automatically.

popular, elastic surface models. When reconstructing an object of arbitrary topology, the particles can be made to "flow" over the data, extracting and conforming to meaningful surfaces.

Our oriented particles have been used as the basis for an interactive surface modeling system [18]. With this system, users can spray collections of points into space to form elastic sheets, shape them using deformation tools, and then freeze the surfaces into the desired final shape. They can create any desired topology with this technique. For example, they can deform a flat sheet into an object with a protrusion and then change the topology to create a looped handle (Figure 1). Forming such surfaces with traditional spline patches is a difficult problem that requires careful attention to patch continuities. Other examples of modeling operations in this system include "cold welding" two surfaces together by abutting their edges, "cutting" surfaces with with a knife-like constraint tool, and "creasing" surfaces by designating certain particles to be *unoriented* [18].

2 Oriented Particle System

Particle systems have been used extensively in computer graphics to model certain natural phenomena such as fire [11]. A simple particle system consists of a large number of point masses (particles) moving according to Newton's laws under the influence of external forces such as gravity, vortex fields, and collisions with obstacles.

Ideas from molecular dynamics have been used to develop models of fluids and deformable solids using collections of *interacting* particles [23]. In these models, long-range attraction forces and short-range repulsion forces control the dynamics of the system. Typically, these forces are derived from an intermolecular potential function such as the Lennard-Jones function

$$\phi_{\mathrm{LJ}}(\mathbf{r}_{ij}) = A||\mathbf{r}_{ij}||^{-n} - B||\mathbf{r}_{ij}||^{-m} \qquad (1)$$

(Figure 2), where $||\mathbf{r}_{ij}|| = ||\mathbf{p}_j - \mathbf{p}_i||$ is the distance between molecules i at $\mathbf{p}_i$ and j at $\mathbf{p}_j$ and m, n, A, B are constants (we use the default values $A = B = 1.0$, $m = 1$, $n = 3$). When internal forces dominate over external forces, particles will tend to bond together into closely packed structures to minimize their total energy, thereby behaving like solids. As internal forces decrease, the behavior resembles that of viscous fluids.

To enable particles to model surfaces rather than volumes, we introduce *oriented particles*. An oriented particle has a position and an orientation, for a total of six degrees of freedom in each particle's state. Each oriented particle defines both a normal vector ($\mathbf{n}_i = z$ in Figure 2b) and a local tangent plane to the surface (defined by the local x and y vectors). More formally, we write the state of each particle as $(\mathbf{p}_i, \mathbf{R}_i)$, where $\mathbf{p}_i$ is the particle's position and $\mathbf{R}_i$ is a 3×3 rotation matrix which defines the orientation of its local coordinate frame (relative to the global frame (X, Y, Z)). The third column of $\mathbf{R}_i$ is the local normal vector $\mathbf{n}_i$.

To encourage oriented particles to group themselves into surface-like arrangements, we devise three new potential functions. These potential functions can be derived from the deformation energies of local triangular patches using finite element analysis [17]. The three functions are:

1. A *co-planarity* potential

$$\phi_{\mathrm{P}}(\mathbf{n}_i, \mathbf{r}_{ij}) = (\mathbf{n}_i \cdot \mathbf{r}_{ij})^2 \psi(||\mathbf{r}_{ij}||), \qquad (2)$$

which encourages neighboring particles to lie in each other's tangent planes, and therefore favors flat (planar) surfaces. The weighting function $\psi(r)$ is a monotone decreasing function used to limit the range of inter-particle interactions. We use $\psi(r) = e^{-r^2/2\sigma_r^2}$ with $\sigma_r = 1.0$.

2. A *co-normality* potential

$$\phi_{\mathrm{N}}(\mathbf{n}_i, \mathbf{n}_j, \mathbf{r}_{ij}) = ||\mathbf{n}_i - \mathbf{n}_j||^2 \psi(||\mathbf{r}_{ij}||), \qquad (3)$$

which attempts to line up neighboring normals.

3. A *co-circularity* potential

$$\phi_{\mathrm{C}}(\mathbf{n}_i, \mathbf{n}_j, \mathbf{r}_{ij}) = ((\mathbf{n}_i + \mathbf{n}_j) \cdot \mathbf{r}_{ij})^2 \psi(||\mathbf{r}_{ij}||) \qquad (4)$$

which is zero when normals are antisymmetrical with respect to the vector joining two particles, and therefore favors surfaces with constant curvature (spherical surfaces).

To control the bending and stiffness characteristics of our deformable surface, we use a weighted sum of potential energies

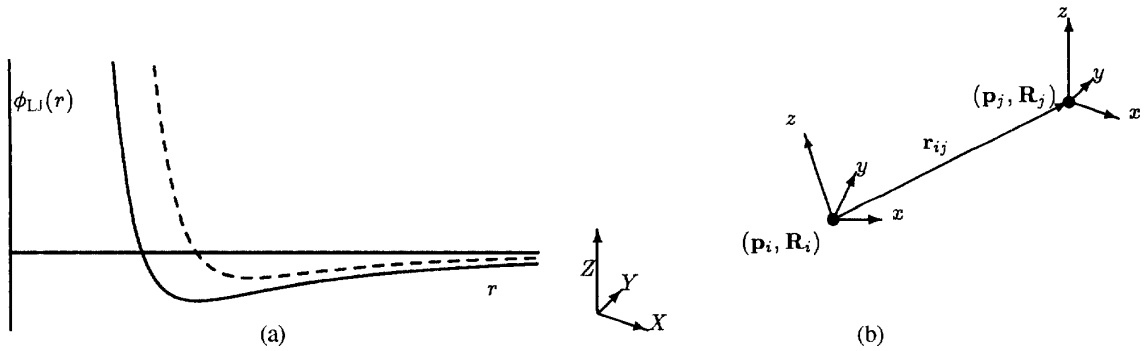

Figure 2: (a) Lennard-Jones type function: the solid line shows the potential function $\phi_{LJ}(r)$, and the dashed line shows the force function $f(r) = -\frac{d}{dr}\phi_{LJ}(r)$; (b) two interacting oriented particles: the interparticle distance $\mathbf{r}_{ij}$ is computed from the global coordinates $\mathbf{p}_i$ and $\mathbf{p}_j$ of particles i and j.

$$E_{ij} = \alpha_{LJ}\phi_{LJ}(\|\mathbf{r}_{ij}\|) + \alpha_P\phi_P(\mathbf{n}_i, \mathbf{r}_{ij}) + \alpha_N\phi_N(\mathbf{n}_i, \mathbf{n}_j, \mathbf{r}_{ij}) + \alpha_C\phi_C(\mathbf{n}_i, \mathbf{n}_j, \mathbf{r}_{ij}). \quad (5)$$

The first term controls the average inter-particle spacing, the next two terms control the surface's resistance to bending, and the last controls the surface's tendency towards uniform local curvature. We use the default values $\alpha_{LJ} = 2.0$, $\alpha_P = \alpha_C = 1.7$, $\alpha_N = 1.0$ (normally only one of α_P or α_C is non-zero). The total internal energy of the system E_{int} is computed by summing the inter-particle energies

$$E_{int} = \sum_i \sum_{j \in \mathcal{N}_i} E_{ij},$$

where $\mathcal{N}_i$ are the neighbors of i (Section 3).

3 Particle Dynamics

Having defined the internal energy associated with our system, we can derive its equations of motion. The derivative of the inter-particle potential with respect to the particle position and orientations gives rise to forces acting on the positions and torques acting on the orientations. The formulas for the inter-particle forces $\mathbf{f}_{ij}$ and torques $\boldsymbol{\tau}_{ij}$ are given in [17]. The standard Newtonian equations of motion are then integrated using an explicit Euler's method [17, 18], which is equivalent to energy minimization with gradient descent.

A straightforward evaluation of the forces and torques at all of the particles requires $O(N^2)$ computation, where N is the number of particles. For large values of N, this can be prohibitively expensive. This computation has been shown to be reducible to $O(N \log N)$ time by hierarchical structuring of the particles [2]. In our work, we use a *k-d tree* [12] to subdivide space so that we can efficiently find all the particle's neighbors within some radius (usually 3 r_0, where r_0 is the natural inter-particle spacing). To further reduce computation, we perform this operation only occasionally and cache the list of neighbors for intermediate time steps.

4 Triangulation

Because our particle system does not give us an explicitly triangulated surface, we have developed an algorithm for triangulating particles. A commonly used technique for triangulating a 2-D surface or a 3-D volume is the Delaunay triangulation [5, 10]. In 2-D, a triangle is part of the Delaunay triangulation if no other vertices are within the circle circumscribing the triangle. To extend this idea to 3-D, we check the smallest sphere circumscribing each triangle (this is a 3-D analogue of the Gabriel graph [10]). We also limit the length of valid triangle edges (to 2 units, by default). This heuristic works well in practice when the surface is adequately sampled with respect to the curvature.

To better visualize the resulting surface, Gouraud, Phong, or flat shading can be applied to each triangle. To obtain a smoother surface, a cubic patch can be interpolated at each triangle (since we know the normals at each vertex). The major benefit of smoothly interpolating the surface across each triangle is that we can compute local differential geometric quantities and support a finite element analysis on the patch deformation energies [21, 7]. In [19] we describe an approximately G^1 (first order geometric) continuous triangular Bezier patch that we have developed to perform this interpolation.

5 Surface Fitting

An important application of our oriented particle systems is the interpolation and extrapolation of sparse 3-D data. This is a particularly difficult problem when the topology of the surface to be fitted is unknown. Oriented particles can provide a solution to the unknown topology problem by extending the surface out from known data points. This technique is particularly useful for interpolating sparse position measurements

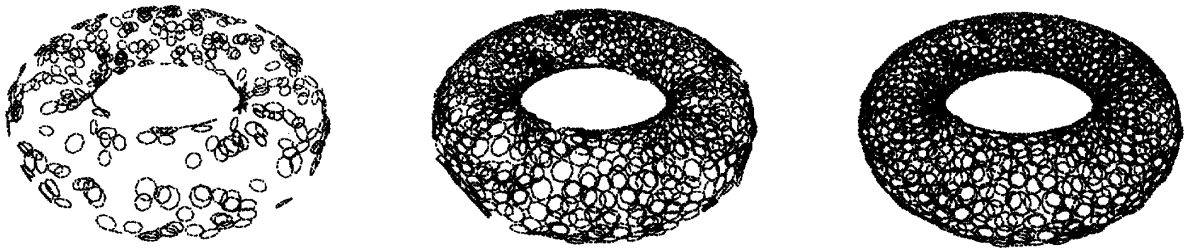

Figure 3: Surface interpolation through a collection of 3-D points. The surface extends outward from the seed points until it fills in the gaps and forms a complete surface.

available from stereo or tactile sensing [15].

The basic components of our particle-based surface extension algorithm are two heuristic rules controlling the addition of new particles. These rules are based on the assumption that the particles on the surface are in a near-equilibrium configuration with respect to the flatness, bending, and inter-particle spacing potentials.

The first (*stretching*) rule checks to see if two neighboring particles have a large enough separation between them to add a new particle. If two particles are separated by a distance d such that $d_{\min} \leq d \leq d_{\max}$, we create a candidate particle at the midpoint and check that there are no other particles within $0.5\, d_{\min}$. Typically $d_{\min} \approx 2.0\, r_0$ and $d_{\max} \approx 2.5\, r_0$, where r_0 is the natural inter-particle spacing. An example of this stretching rule in action are shown in Figure 1.

The second (*growing*) rule allows particles to be added in all directions with respect to a particle's local x-y plane. The rule is generalized to allow a minimum and maximum number of neighbors and to limit growth in regions of few neighboring particles, such as at the edge of a surface. The rule counts the number of immediate neighbors n_N to see if it falls within a valid range $n_{\min} \leq n_N \leq n_{\max}$. It also computes the angles between successive neighbors $\Delta\theta_i = \theta_{i+1} - \theta_i$ using the particle's local coordinate frame, and checks if these fall within a suitable range $\theta_{\min} \leq \Delta\theta_i \leq \theta_{\max}$. If these conditions are met, one or more particles are created in the gap. In general, a sheet at equilibrium will have interior particles with six neighbors spaced 60° apart while edge particles will have four neighbors with one pair of neighbors 180° apart.

With these two rules, we can automatically build a surface from collections of 3-D points. We create particles at each sample location and fix their positions and orientations. We then start filling in gaps by growing particles away from isolated points and edges. After completing a rough surface approximation, we can release the original sampled particles to smooth the final surface, thereby eliminating excessive noise. If the set of data points is reasonably distributed, this approach will result in a smooth continuous closed surface (Figure 3). The fitted surface does not assume a particular topology, unlike previous 3-D surface fitting models such as [22].

In conjunction with fitting the surface, we can estimate local differential geometric quantities such as the principal directions and minimum and maximum curvatures. This can be achieved by simply adding an extra potential function that induces a torque around the local z axis and which forces the x and y axes to align themselves in the directions of minimum and maximum curvature [19]. The resulting system of oriented particles resembles the collection of interacting Darboux frames used by Sander and Zucker [13].

6 3-D Volume Segmentation

Our surface fitting algorithm may be used to help segment structures in 3-D volumetric data such as CT, MRI, or other 3-D medical imagery. To perform this segmentation, we first apply a 3-D edge operator [8] to the data and use the edges to initialize and attract particles, or directly use gradients in the 3-D image as external forces on the particles. In this application, our 3-D surface model can be viewed as a generalization of the active deformable surface model [22, 7], but without the restrictions imposed by a manually selected surface topology.

Figure 4a–c shows slices from a CT scan of a plastic "phantom" vertera model (decimated to $120 \times 128 \times 52$ resolution). Figure 4d shows the reconstructed 3-D model. This smooth, triangulated model contains 6650 particles and 13829 triangles, and was created by seeding a single particle and extending the surface along high 3D edge values until a closed surface was obtained. Figure 4e shows a Gouraud shaded rendering of the reconstructed surface.

7 Surfaces from Silhouettes

We have applied our particle-based approach to the reconstruction of triangulated surface models from the output of a shape-from-silhouettes algorithm [16]. The algorithm constructs a bounding volume for the object by intersecting silhouettes from a sequence of views taken around an object—in this case, a cup—as it rotates on a turntable. The algorithm represents the volume using an octree [12] (Figure 5a).

To reconstruct the surface model of the cup, we first create a volume occupancy array from the octree representation and then apply the 3D edge operator and our reconstruction algorithm as in the vertebra example. Figure 5b shows the reconstructed model of the cup. The reconstructed surface has 3722 particles and 7568 triangles.

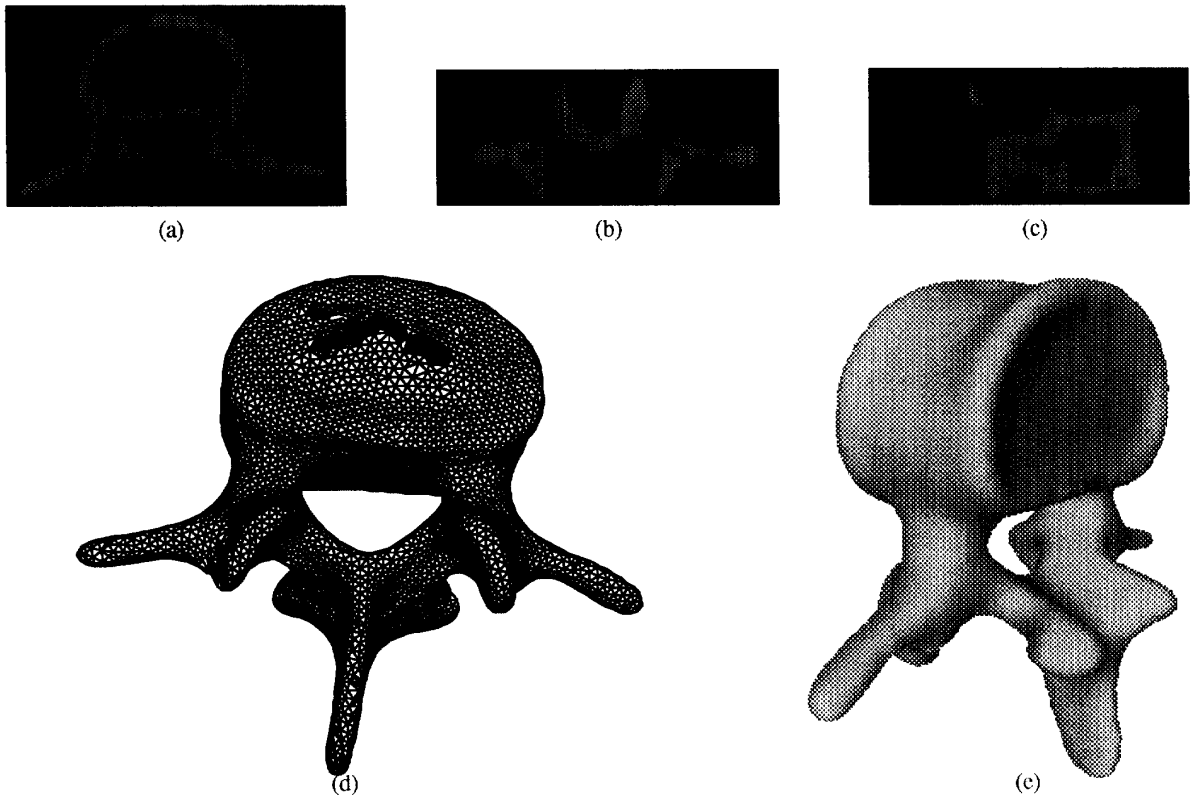

Figure 4: 3-D Reconstruction of a vertebra from 120 × 128 × 52 CT volume data: (a) xy slice, (b) xz slice, (c) yz slice, (d) reconstructed 3-D surface model with triangulated particles, (e) shaded surface.

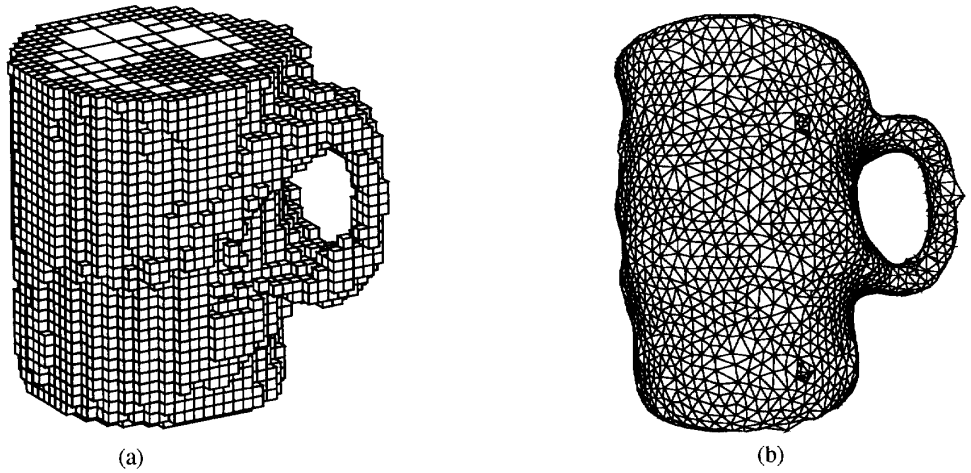

Figure 5: Reconstruction of a surface model of a cup from silhouettes: (a) cup bounding volume represented as an octree, (b) triangulated surface of reconstructed model.

8 Discussion

The particle-based surface model we have developed has a number of advantages over traditional spline-based surfaces. Particle-based surfaces are easy to shape, extend, join, and separate. By adjusting the relative strengths of various potential functions, the surface's resistance to stretching, bending, or variation in curvature can all be controlled. The topology of particle-based surfaces can easily be modified, as can the sampling density, and surfaces can be fitted to arbitrary collections of 3-D data points.

Our particle-based surface model shares some characteristics with local patch models such as those developed by Sander and Zucker [13]. Our particles interact in a manner similar to their frames, but in addition, particle positions are not fixed, so that a more even distribution of samples can be achieved. More importantly, the triangulation process ensures that a globally consistent smooth analytic surface is defined at all times. This enables us to derive interaction potentials based on finite element analysis, so that arbitrary smoothness conditions or material properties can be simulated.

In future work, we plan to extend our finite element analysis to the large deflection case, and to integrate external forces over each triangle, thus achieving a higher degree of accuracy. We also plan to track deformable objects (e.g., objects undergoing non-rigid motion such as a beating heart) using a Kalman filter. We have begun developing techniques for curvature-dependent adaptive meshing of our surfaces, which should increase fidelity and decrease overall complexity, and on combining our particle-based surface model with a similar particle-based line process (curve model), which could be used to model surface terminators, creases, and surface markings [19].

9 Conclusion

We have developed a particle-based model of deformable surfaces and applied it to several computer vision problems. Our new model, which is based on oriented particles with new interaction potentials, has characteristics of both physics-based surface models and of particle systems. It can be used to model smooth, elastic, moldable surfaces, like traditional splines, and it also allows for arbitrary interactions and topologies.

Our oriented particle surface model can be used to automatically fit a surface to sparse 3-D data even when the topology of the surface is unknown. We can also use our surfaces to segment 3-D volumetric data, and to incrementally construct 3-D object models from motion sequences. Because of the flexibility of the technique, and because of its close relationship to analytic finite element modeling, we believe this approach will form the basis of a powerful new class of shape models for numerous computer vision applications.

Acknowledgements

The Canny-Deriche-Monga edge detector was provided courtesy of Nicholas Ayache and Gregoire Malandain of INRIA, France.

References

[1] G. J. Agin and T. O. Binford. Computer description of curved objects. *IEEE Trans. Comput.*, C-25(4):439–449, April 1976.

[2] A. Appel. An efficient algorithm for many-body simulations. *SIAM J. Sci. Stat. Comput.*, 6(1), 1985.

[3] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. In A. R. Hanson and E. M. Riseman, editors, *Computer Vision Systems*, pp. 3–26. Academic Press, New York, 1978.

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

[5] J.-D. Boissonat. Representing 2D and 3D shapes with the Delaunay triangulation. In *ICPR'84*, pp. 745–748, Montreal, Canada, July 1984.

[6] H. Delingette, M. Hebert, and K. Ikeuichi. Shape representation and image segmentation using deformable surfaces. In *CVPR'91*, pp. 467–472, Maui, Hawaii, June 1991.

[7] T. McInerney and D. Terzopoulos. A finite element model for 3D shape reconstruction and nonrigid motion tracking. In *ICCV'93*, Berlin, Germany, May 1993.

[8] O. Monga, R. Deriche, G. Maladin, and J. P. Cocquerez. Recursive filtering and edge closing: two primary tools for 3d edge detection. In *ECCV'90*, pp. 56–65, Antibes, France, April 23–27 1990.

[9] A. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986.

[10] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Academic Press, New York, 1985.

[11] W. T. Reeves. Particle systems—a technique for modeling a class of fuzzy objects. *ACM Trans. Graphics*, 2(2):91–108, April 1983.

[12] H. Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, Reading, Massachusetts, 1989.

[13] P. T. Sander and S. W. Zucker. Inferring surface trace and differential structure from 3-D images. *IEEE Trans. Patt. Anal. Mach. Intel.*, 12(9):833–854, September 1990.

[14] R. Szeliski. Bayesian modeling of uncertainty in low-level vision. *Int. J. Comput. Vision*, 5(3):271–301, December 1990.

[15] R. Szeliski. Shape from rotation. In *CVPR'91*, pp. 625–630, Maui, Hawaii, June 1991.

[16] R. Szeliski. Rapid octree construction from image sequences. *CVGIP: Image Understanding*, May 1993.

[17] R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. Technical Report 91/14, Digital Equipment Corporation, Cambridge Research Lab, December 1991.

[18] R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. *Computer Graphics (SIGGRAPH'92)*, 26(2):185–194, July 1992.

[19] R. Szeliski, D. Tonnesen, and D. Terzopoulos. Curvature and continuity control in particle-based surface models. In *SPIE Geometric Methods in Computer Vision II*, San Diego, July 1993. Society of Photo-Optical Instrumentation Engineers.

[20] D. Terzopoulos. The computation of visible-surface representations. *IEEE Trans. Patt. Anal. Mach. Intel.*, PAMI-10(4):417–438, July 1988.

[21] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Trans. Patt. Anal. Mach. Intel.*, 13(7):703–714, July 1991.

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

[23] D. Tonnesen. Modeling liquids and solids using thermal particles. In *Graphics Interface '91*, pp. 255–262, 1991.