Hierarchical Shape Representation Using Locally Adaptive Finite Elements

Eunyoung Koh, Dimitri Metaxas and Norm Badler

Department of Computer and Information Science
University of Pennsylvania, Philadelphia, PA, 19104

Abstract. This paper presents a physics-based algorithm for hierarchical shape representation using deformable models with locally adaptive finite elements. Our new adaptive finite element algorithm ensures that during subdivision the desirable finite element mesh generation properties of conformity, non-degeneracy and smoothness are maintained. Through our algorithm, we locally subdivide the triangular finite elements based on the distance between the given datapoints and the model. In this way, we can very efficiently and accurately represent the shape of an object with a resulting small number of model nodes. Furthermore, using our locally adaptive subdivision algorithm in conjunction with our model's global deformations we construct a hierarchical representation of the given 3D data.

1 Introduction

Recovering object shape for the purposes of shape reconstruction and object recognition is a very important problem in computer vision. Most of the current physics-based shape recovery algorithms assume fixed-size grids [12, 11, 6, 4, 14, 1]. Recently, there have been some attempts to develop techniques with adaptive size grids. McInerney and Terzopoulos [5] present a finite element technique with uniform grid refinement capabilities, while Vasilescu and Terzopoulos [13] proposed a technique for adaptive subdivision of meshes consisting of nodal masses interconnected by adjustable springs. Even though their technique works well for adaptive fixed-size meshes, 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. Huang and Goldgof [2] present a geometric adaptive subdivision algorithm for nonrigid motion analysis which uses planar triangular patches. Finally, Tanaka and Kishino [10] develop a geometric adaptive mesh generation algorithm for surface reconstruction. Even though the algorithm supports local subdivision, special algorithms need to be employed to deal with cracks often occurring during subdivision.

In this paper we extend our physics-based framework for shape estimation [11, 4] by developing a new technique which allows the robust local adaptive subdivision of an initial finite element grid based on the distances of the given datapoints to the model surface. Our technique is an adaptation of Rivara's local refinement process for finite element grids [7]. Starting from a small number

Lecture Notes in Computer Science, Vol. 800
Jan-Olof Eklundh (Ed.)
Computer Vision – ECCV '94

of finite elements tessellating the model surface, the model locally subdivides based on the above criterion so as to improve the model fitting. In this way with a resulting small number of model nodes we can very efficiently and accurately represent the shape of an object. Our local subdivision algorithm utilizes the properties of triangles bisected by 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 our dynamic models, allows the reconstruction and representation of shape hierarchically.

2 Deformable Models

In this section we briefly review the general formulation of deformable models; further detail can be found in [11, 4].

2.1 Geometry: Global and Local Deformations

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

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

where $\mathbf{c}(t)$ is the origin of the model-centered reference frame ϕ and the orientation of ϕ is given by the rotation matrix $\mathbf{R}(t)$. We further express $\mathbf{p}$ as the sum of a reference global shape $\mathbf{s}(\mathrm{u}, t)$ and a displacement function $\mathbf{d}(\mathrm{u}, t)$ that represents local deformations, i.e., $\mathbf{p} = \mathbf{s} + \mathbf{d}$. In our formulation the reference $\mathbf{s}$ is defined based on parameterized models (e.g., superquadrics) that can undergo global deformations (e.g., tapering, bending) and were introduced in [11, 4]. The vector $\mathbf{s}$ is computed based on some global parameters that we collect into the parameter vector $\mathbf{q}_s$.

We use finite element basis functions to represent local deformations [11, 4]. We collect all the finite element nodal degrees of freedom into a vector of degrees of freedom $\mathbf{q}_d$. We can then write $\mathbf{d} = \mathbf{S}\mathbf{q}_d$, where $\mathbf{S}$ is the shape matrix whose entries are the finite element basis functions. For the applications in this paper we select a thin plate strain energy [12] that ensures C^1 continuity of the solution and triangular finite elements [3] whose shape functions are tensor products of one-dimensional Hermite polynomials [15].

2.2 Dynamics

Our goal when fitting the model to visual data is to recover the vector of degrees of freedom $\mathbf{q} = (\mathbf{q}_c^T, \mathbf{q}_\theta^T, \mathbf{q}_s^T, \mathbf{q}_d^T)^T$, where $\mathbf{q}_c = \mathbf{c}$ and $\mathbf{q}_\theta$ is the vector of the

rotational degrees of freedom of the model. We make our model dynamic in $\mathbf{q}$ by introducing mass, damping, and a deformation strain energy. In applications to vision [4], it makes sense to simplify the motion equations while preserving useful dynamics by setting the mass density $\mu(\mathrm{u})$ to zero to obtain

$$\mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{f}_q, \tag{2}$$

where $\mathbf{D}$ and $\mathbf{K}$ are the damping and stiffness matrices respectively, and where $\mathbf{f}_q(\mathrm{u}, t)$ are the generalized external forces associated with the generalized coordinates of the model. The generalized forces are computed from 3D forces exerted from the data to the model using algorithms developed in [3].

3 Locally Adaptive Finite Elements

To decide whether an element or elements should be subdivided, we use a criterion based on the magnitude of the distance of each datapoint from its corresponding finite element. To compute the corresponding finite element to a datapoint $\mathbf{z}$, we compute the point u_z on the model whose distance $d(\mathrm{u}_z)$ from the given datapoint is minimum. If

$$d(\mathrm{u}_z) > \tau_d, \tag{3}$$

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 u_z lies inside an element, then the element is selected for subdivision.
2. If u_z lies on an edge, then the two adjacent elements to the edge are subdivided.
3. If 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 given that curvature calculation is very sensitive to noise and that we want to use our technique in case of sparse data, we did not consider using the data curvature as a criterion for subdivision. Also our criterion for subdivision does not use the model surface curvature, because the chosen finite elements ensure a smooth C^1 continuous solution.

3.1 Subdivision Algorithm

Our subdivision algorithm is essentially based on Rivara's local refinement algorithm [7]. The subdivision algorithm has the following two basic steps.

Step 1: Bisection Operation

In the first step, the chosen finite element based on the above criterion performs a *bisection operation* as follows: let T be a triangle with vertices A, B, and

C; a natural way of subdividing the triangle T into two triangles for finite element methods [8] is to bisect it along its longest edge; 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. 1(a). This subdivision has been shown to provide properties desirable for use in finite element applications. First, none of the interior angles of the refined triangles will become obtuse as the level of subdivision increases. Rosenberg and Stenger [8] proved that if α_i is the smallest angle of the triangulation obtained by the i-th 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 triangles become approximately equilateral as the level of subdivision increases [9].

Step 2: Conforming Operation

The second part of the algorithm ensures that the resulting finite element grid generates properties necessary for 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 [15]. In Fig. 1(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. In Fig. 1(c), if we introduce a new node, D, as a result of bisecting element T, the element, T_1, adjacent to the subdivided edge AB becomes non-conforming. In order to ensure conformity, further subdivision must be performed on T_1 along the edge common with midpoint D. However, it is possible that the common edge may not be the longest edge of T_1. 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_1 as shown in Fig.1(c). We first bisect T_1 by its longest edge, AE, at its midpoint, P. If AE is the common edge, then we stop subdividing. Otherwise, we further subdivide T_1 by connecting P to the midpoint D of AB. As a result of this process, conformity is preserved and the subdivision will not produce triangles with obtuse angles. This process is called a *conforming operation*. In

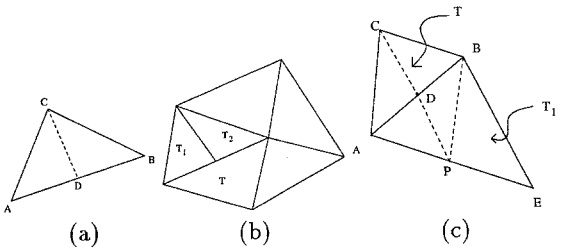

Fig. 1. Various subdivision examples. (a) Subdivision of a triangle by the longest edge, (b) An example of non-conforming triangulation, (c) An illustration of conforming operation.

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.

4 Experiments

Our experiments run at interactive rates on an R4000 Iris Crimson workstation with VGX graphics. In Fig. 2 we fit a deformable model with 627 nodes initially to 5070 3D range datapoints obtained from a head. Fig. 2(a) shows a view of the range data and the initial model. Fig. 2(b) shows the model fitted to the data without local subdivision, Fig. 2(c) shows the final model fitted to the data after five levels of local subdivision. The improvement in shape representation with respect to the level of subdivision is obvious. The new final number of model nodes is 1597, which is significantly smaller than the number of given datapoints.

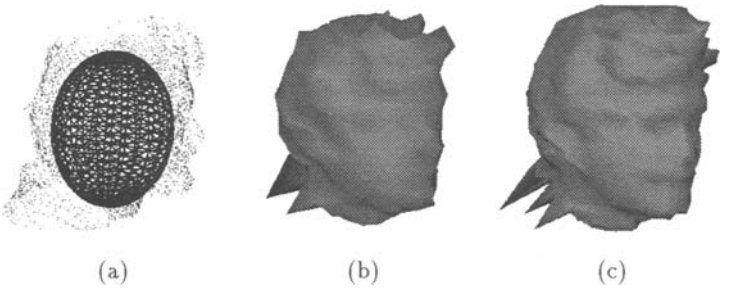

Fig. 2. Fitting of model to head data. (a) Model Initialization, (b) Model fitted to data without local subdivision, (c) Model fitted to data after five levels of local subdivision.

Fig. 3 (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 experiment. The numbers of elements used at each level of representation were 18155, 7292, and 2260 respectively, and the numbers of nodes were 18005, 3696, and 1180 respectively.

References

1. L. Cohen and I. Cohen. Finite Element Methods for Active Contour Models and Balloons for 2D and 3D Images. PAMI # 91-11-19.

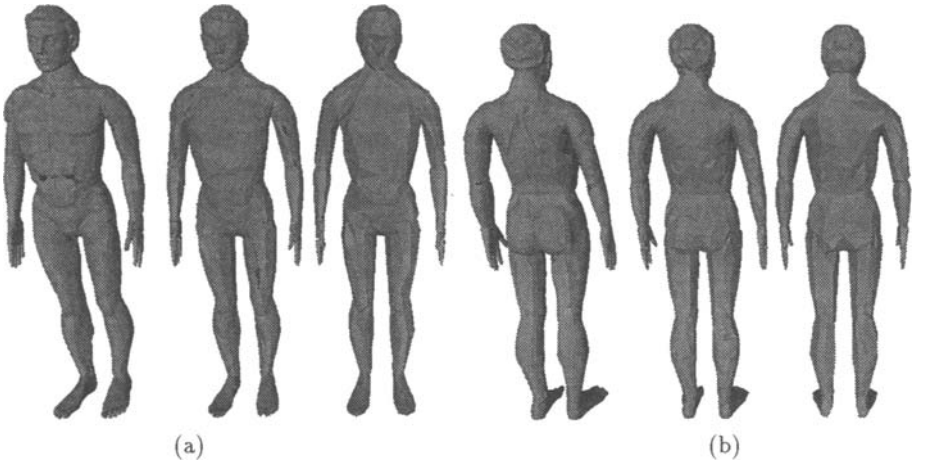

Fig. 3. A human body displayed at three different levels of detail: (a) the front view, (b) the back view.

2. W. C. Huang, and D. Goldgof, Adaptive-Size Physically-Based Models for Nonrigid Motion Analysis, Proc. IEEE CVPR'92, pp. 833–835, Champaign, IL, June 1992.
3. D. Metaxas, Physics-Based Modeling of Nonrigid Objects for Vision and Graphics, Ph.D. Thesis, Dept. of CS, Univ. of Toronto, 1992.
4. D. Metaxas, and D. Terzopoulos, Shape and Nonrigid Motion Estimation Through Physics-Based Synthesis, IEEE PAMI, 15(6), pp. 580-591, June, 1993.
5. T. McInerney and D. Terzopoulos, Proc. of ICCV'93, pp. 518-523, Berlin, 1993.
6. A. Pentland and B. Horowitz. Recovery of Non-rigid Motion and Structure. *IEEE PAMI*, 13(7):730–742, 1991.
7. M. C. Rivara, Algorithms for refining triangular grids suitable for adaptive and multigrid techniques, International Journal for Numerical Methods in Engineering, 20, pp. 745–756, 1984.
8. I. G. Rosenberg, and F. Stenger, A lower bound on the angles of triangles constructed by bisecting the longest side,*Math. Comp.*, 29, pp. 390–395, 1975.
9. M. Stynes, On fast convergence of the bisection method for all triangles, *Math. Comp.*, 35, pp. 1195–1201, 1980.
10. H. Tanaka and F. Kishino. Adaptive Mesh Generation for Surface Reconstruction: Parallel Hierarchical Triangulation Without Discontinuities. Proc. CVPR'93, pp. 88-94, NY, 1993.
11. D. Terzopoulos and D. Metaxas. Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics. *IEEE PAMI*, 13(7):703–714, 1991.
12. D. Terzopoulos, A. Witkin, and M. Kass. Constraints on Deformable Models: Recovering 3D Shape and Nonrigid motion. *A.I.*, 36(1):91–123, 1988.
13. M. Vasilescu, and D. Terzopoulos, Adaptive Meshes and Shells: Irregular Triangulation, Discontinuities, and Hierarchical Subdivision, Proc. IEEE CVPR'92, pp. 829–832, Champaign, Illinois, June 1992.
14. B.C. Vemuri and A. Radisavljevic. From Global to Local, a Continuum of Shape Models with Fractal Priors. Proc. CVPR'93, pp. 307-313, NY, 1993.
15. O. Zienkiewicz. *The Finite Element Method.* McGraw-Hill, 1977.