

Adaptive Meshes and Shells: Irregular Triangulation, Discontinuities, and Hierarchical Subdivision

M. Vasilescu¹ and Demetri Terzopoulos²

¹MIT Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA, 02139

²Department of Computer Science, University of Toronto, 10 King's College Rd, Toronto, ON, M5S 1A4

Abstract

Adaptive meshes are dynamic networks of nodal masses interconnected by adjustable springs. They are useful for nonuniformly sampling and reconstructing visual data. This paper extends the adaptive mesh model in the following ways: it (i) develops open adaptive meshes and closed adaptive shells based on triangular and rectangular elements, (ii) proposes a discontinuity detection and preservation algorithm suitable for the model, and (iii) develops techniques for adaptive hierarchical subdivision of adaptive meshes and shells. The extended model is applied to image and 3D surface data.

1 Introduction

In [1] we proposed the *adaptive mesh* model and demonstrated its application to the nonuniform sampling and reconstruction visual data. Adaptive meshes are dynamic networks of nodal masses interconnected by adjustable springs. Acting as mobile sampling sites, the nodes observe interesting properties of the data, such as intensities, depths, gradients, and curvatures. Based on the nodal observations, the springs automatically adjust their stiffnesses so as to distribute the available nodal degrees of freedom of the reconstructed model in accordance with the local complexity of the input data. Adaptation can significantly improve the quality of the reconstruction.

The adaptive mesh is a versatile model. In this paper we modify and extend it in several ways. We construct closed adaptive meshes, or adaptive shells. It is often preferable to use triangular rather than quadrangular spring-mass elements in adaptive meshes, because the triangle (the 2D simplex) offers the greatest flexibility in tessellating open and closed 2D manifolds in 3D Euclidean space. We further generalize the model to handle piecewise continuous data by introducing a suitable discontinuity detection and preservation algorithm. Finally we develop techniques for adaptive hierarchical subdivision of polygonal elements in adaptive meshes and shells. This allows us to significantly improve the model's range of adaptivity. Huang and Goldgof [2] have independently developed a related algorithm for reconstructing closed adaptive meshes with subdivisible triangular elements. Their technique differs from ours with regard to the details of the element subdivision and the absence of mechanisms for discontinuity detection/preservation and continuous adaptivity. After briefly reviewing the basic adaptive mesh model in the next section, we develop the aforementioned extensions to our algorithm and apply it to image and 3D surface data.

2 Review of Adaptive Meshes

An adaptive mesh is a discrete dynamic system constructed from a set of mobile nodes and adjustable springs. The nodes, indexed by i , for $i = 1, \dots, N$, have masses m_i and 3-space positions $\mathbf{x}_i(t) = [x_i(t), y_i(t), z_i(t)]$. The adaptive mesh is governed by the set of ordinary differential equations of motion

$$m_i \frac{d^2 \mathbf{x}_i}{dt^2} + \gamma_i \frac{d\mathbf{x}_i}{dt} + \mathbf{g}_i = \mathbf{f}_i; \quad i = 1, \dots, N, \quad (1)$$

where $d^2 \mathbf{x}_i / dt^2$ is the acceleration of node i , $d\mathbf{x}_i / dt$ is its velocity, γ_i is a damping coefficient that controls the rate of dissipation of the node's kinetic energy, \mathbf{f}_i is an external force applied at the node which typically serves to couple the mesh to input data for the purposes of reconstruction, and

$$\mathbf{g}_i(t) = \sum_{j \in \mathcal{N}_i} \mathbf{s}_{ij} \quad (2)$$

is the total force on node i due to adjustable springs connecting it to neighboring nodes j . Here,

$$\mathbf{s}_{ij} = \frac{c_{ij} e_{ij}}{\|\mathbf{r}_{ij}\|} \mathbf{r}_{ij}, \quad (3)$$

is the force that a spring with stiffness c_{ij} exerts on node i , where the norm of $\mathbf{r}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ is the actual length of the spring and $e_{ij} = \|\mathbf{r}_{ij}\| - l_{ij}$ is the deformation of the spring relative to a natural length l_{ij} . The l_{ij} control the minimal distances between nodes.

Each node i of an adaptive mesh acts as a mobile sampling site. It computes local observations $O_i \in [0, 1]$ on input data $\mathbf{d}(\boldsymbol{\xi})$ by sampling an adaptation function derived from \mathbf{d} which identifies important features of the data, such as high gradients, curvatures, etc. (see [1] for examples). The spring stiffnesses c_{ij} adapt as a function of the observations according to $c_{ij} = (1 - \rho_{ij})c_{\min} - \rho_{ij}c_{\max}$, where c_{\min} and c_{\max} are minimum and maximum allowable values for the spring constant, and $\rho_{ij} = \frac{1}{2}(O_i + O_j)$, is the average value of the observations computed at nodes i and j connected by the spring. The stiffness of spring ij grows with larger ρ_{ij} , thereby pulling nodes i and j closer together. Thus the node density adapts to the magnitude of the adaptation function and, consequently, to the complexity of the data.

The second-order equations of motion (1) of the adaptive mesh are simulated using an explicit Euler procedure (see [1] for further details).

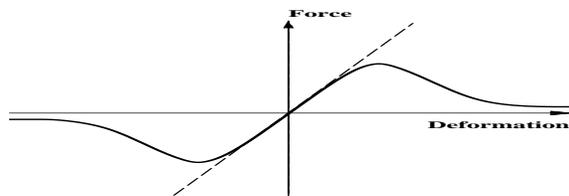


Figure 1: Discontinuity spring force.

3 Discontinuities

We now extend adaptive meshes to accommodate jump (C^0) discontinuities in the data.¹ Discontinuity detection and preservation in visual surface reconstruction is a difficult and intriguing problem (e.g., [4, 5, 6, 3]). Our discrete mechanical model suggests interesting variations upon recent discontinuity-preserving Markov random field [7] and variational surface reconstruction [8] techniques.

The original adaptive mesh model responded to discontinuities in the data by concentrating nodes (i.e., degrees of freedom) in the vicinity of abrupt changes, enabling the mesh to conform better to the changes. The discontinuity algorithm that we develop here deals with discontinuities directly, removing the burden from the adaptation process, whose job then becomes to produce a mesh that adapts to complexity of the data in continuous regions.

Discontinuity detection and preservation involves a modification of the force (s) vs deformation (e) relationship of the adaptive springs. We modify the linear relationship (3) illustrated by the dashed line (with slope c_{ij}) in Fig. 1 to obtain the nonlinear relationship shown by the solid curve. The humped curve represents a spring which asymptotically “gives up” trying to pull nodes together that are forced far apart by data on opposite sides of a large discontinuity. We can obtain the humped form using several different functions. A simple one is $e/(1 + e^2)$.²

Since the adaptive mesh is under tension, however, the force/deformation profile of springs in an adaptive mesh ought to be modified only in the direction of the discontinuity. For example, in the case of single-valued surface reconstruction, we modify only the z component of the spring force. If we were also to modify the x and y components, the balance of forces in the plane of the mesh would be altered and the mesh would develop large gashes in the vicinity of discontinuities. We therefore modify (3) as follows:

$$\mathbf{s}_{ij} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/(1 + \beta e_{ij}^2) \end{pmatrix} \frac{c_{ij} e_{ij}}{\|\mathbf{r}_{ij}\|} \mathbf{r}_{ij}, \quad (4)$$

where β is a discontinuity sensitivity coefficient.

Fig. 2 illustrates the behavior of the discontinuity procedure when applied to the range map of a statuette which was also used in [1]. A triangular dynamic mesh model is applied in Fig. 2(a) using $\beta = 0$ in order to disable the discontinuity mechanism (refer to

¹Since the adaptive mesh has no flexural rigidity (unlike a thin-plate model) it automatically preserves C^1 discontinuities in the data. However, the lack of rigidity can be a drawback when dealing with sparse, noisy data sets [3].

²Li [8] mentions a version of this function for variational surface reconstruction.

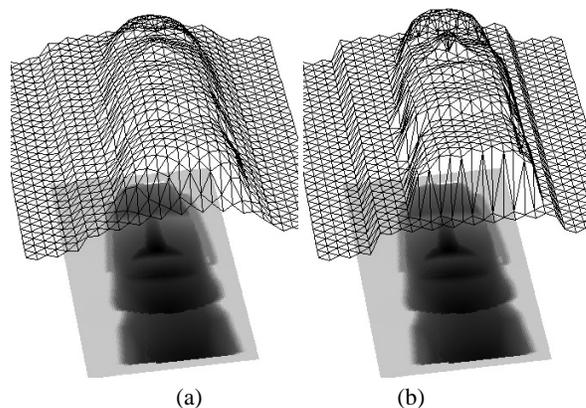


Figure 2: Adaptive mesh reconstruction without (a) and with (b) discontinuity detection.

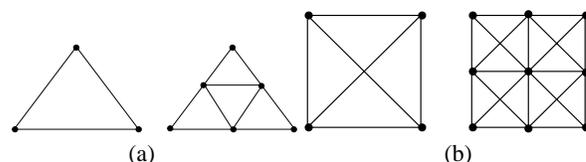


Figure 3: Dividing a triangle (a), a quadrangle (b).

[1] for simulation parameter values). Note how the jump discontinuities between the statuette and background are inappropriately smoothed. Fig. 2(b) shows how the discontinuity process detects and preserves the occluding boundary and other prominent depth changes when $\beta = 1.0$. We have not enabled the mesh adaptation process in this example in order to illustrate the effects of the discontinuity procedure more clearly.

4 Hierarchical Subdivision & Merging

In this section, we enhance the adaptivity of the mesh model by introducing an algorithm for subdividing and merging triangle and quadrangle elements. The model supports the recursive application of these operations by maintaining a hierarchical data structure, where elements include pointers to their constituent springs and nodes and, if they have been divided, to their children elements.

Fig. 3(a) illustrates the division of a triangle of the mesh into 4 children, where each existing spring is divided in two at its midpoint and 3 new nodes along with 3 new springs are added to the mesh data structure.³ This type of subdivision produces finer meshes with the same triangular structure and it maintains the “fat” shape of the subtriangles.⁴

³Fig. 3(b) illustrates the division of the quadrangle element that we employed in [1] into 4 children. Each existing spring is divided in half and 8 new springs are added along with 5 new nodes.

⁴The subdivision illustrated in Fig. 3(a) is superior to the subdivision obtained by connecting the 3 nodes to a new node introduced at the center, as is employed in [2]. Since the latter method produces increasingly long, thin triangles (“slivers”) with successive subdivisions, it is inappropriate for recursive subdivision.

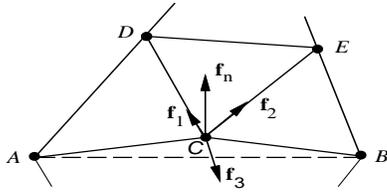


Figure 4: Illustration of collinearity constraint.

The possibility of adaptively subdividing only some of the triangles in the mesh raises complications, again because all the springs are normally under tension. Consider two triangles that share a common spring. If only one of these triangles is subdivided as in Fig. 3(a), the new node introduced on the divided spring must be constrained to slide like a bead along the common edge, otherwise the springs added to the subdivided triangle will pull the node away from the edge, thereby destroying the triangular structure of the mesh. The constraint procedure remedies this undesirable effect which would otherwise be exacerbated by subsequent subdivisions.

The constraint procedure illustrated by Fig. 4 is applied at each time step of the Euler integration of (1) and recursively to each subdivided spring. Node C must be constrained to slide along the common edge AB (every line segment in the diagram represents a spring). The procedure first cancels the component of the net nodal force $\mathbf{f}_C^{n,t}$ of node C that acts perpendicular to line AB. The remaining tangential force can displace C parallel to line AB only. Let \mathbf{f}_3 be the sum of the forces exerted by springs AC and BC on node C. The constraint procedure adds the normal component of \mathbf{f}_3 to $\mathbf{f}_C^{n,t}$ which will draw C towards line AB. Finally, since ACB should behave like a rigid “rod,” the normal component \mathbf{f}_n of force $\mathbf{f}_1 + \mathbf{f}_2$ due to springs DC and EC is partitioned and added to the net nodal forces on nodes A and B, according to the ratio of the distances of C from these nodes.

The vision literature abounds with methods for subdividing and merging hierarchical image data structures based on various criteria (e.g., [9, 10]). In the context of adaptive meshes, we can extend the idea of nodes as sampling sites to the triangles or quadrangles that they define. A possibility that is suitable for image data is to compute regional properties of the intensity function (such as average gradients) within element areas and subdivide or merge elements based on property values relative to prespecified thresholds.⁵ This is illustrated in Fig. 5. The image in Fig. 5(a) is sampled and reconstructed in Fig. 5(b) using a 30×30 uniform triangular mesh, with poor results. In Fig. 5(c) the uniform mesh has been allowed to adapt and subdivide adaptively based on the average intensity gradient within each triangle. The triangles have subdivided around high gradient areas. Fig. 5(d) shows a closeup of part of the mesh to illustrate the effect of the collinearity constraint procedure. Fig. 5(e–f) demonstrate the results obtained after a second round of subdivision and adaptation. Note the wide range of triangle sizes over the image.

⁵A better criterion for subdividing and merging elements for the case of 3D data is considered in the next section.

5 Adaptive Shells

This section considers adaptive meshes in the form of closed shells in space constructed from triangular elements (we also employ generalized, viscoelastic springs, but must omit the details due to lack of space). We can use adaptive shells to reconstruct surface models from scattered 3D points in space. The reconstruction is based on applying traction forces (\mathbf{f}_i in (1)) from each data point to the nearest node on the shell to deform the shell into a consistent shape. Each element of the mesh defines a local approximating function (a plane in the case of triangles). When the shell achieves equilibrium, we can compute the residual or discrepancy between this function and the 3D data and subdivide the element if the residual exceeds a threshold.

To illustrate the method, we initialize the shell as a regular icosahedron within a “cloud” of 3D points which were sampled from a mannequin foot using the Cyberware, Inc., 3D digitizer (Fig. 6(a)). Fig. 6(b) shows the equilibrium position of the shell after one subdivision step. Fig. 6(b) shows the shell at equilibrium after the third subdivision step, rendered as a shaded surface.

The final figure illustrates the adaptive shell algorithm with the discontinuity process applied to 3D samples of a friend’s head. Fig. 7(a) shows the 3D point data. Fig. 7(b) shows a preliminary reconstructed model with 900 nodes. Here, the discontinuity process has been disabled. Note the excessive smoothness of the model. Fig. 7(c) shows the model after another subdivision step (3197 nodes), and with the discontinuity and adaptation processes enabled.

References

- [1] D. Terzopoulos and M. Vasilescu. Sampling and reconstruction using adaptive meshes. In *Proc. Computer Vision and Pattern Recognition Conf. (CVPR'91)*, pages 70–75, 1991.
- [2] W.-C. Huang and D. B. Goldgof. Adaptive-Size meshes for sampling and surface reconstruction. In *Proc. Applications of AI X: Machine Vision and Robotics*, Orlando, FL, 1992. to appear.
- [3] D. Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):417–438, 1988.
- [4] D. Terzopoulos. The role of constraints and discontinuities in visible-surface reconstruction. In *Proc. 8th Int. J. Conf. AI*, pages 1073–1077, 1983.
- [5] W.E.L. Grimson and T. Pavlidis. Discontinuity detection for visual surface reconstruction. *Computer Vision, Graphics, and Image Processing*, 30:316–330, 1985.
- [6] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.
- [7] D. Geiger and F. Girosi. Parallel and deterministic algorithms from MRF’s: Surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):401–412, 1991.
- [8] S.Z. Li. Reconstruction without discontinuities. In *Proc. Computer Vision and Pattern Recognition Conf. (CVPR'91)*, pages 709–712, 1991.
- [9] S. Tanimoto and A. Klinger. *Structured Computer Vision: Machine Perception through Hierarchical Computation Structures*. Academic Press, New York, 1980.
- [10] A. Rosenfeld. *Multiresolution Image Processing and Analysis*. Springer-Verlag, Berlin, 1984.

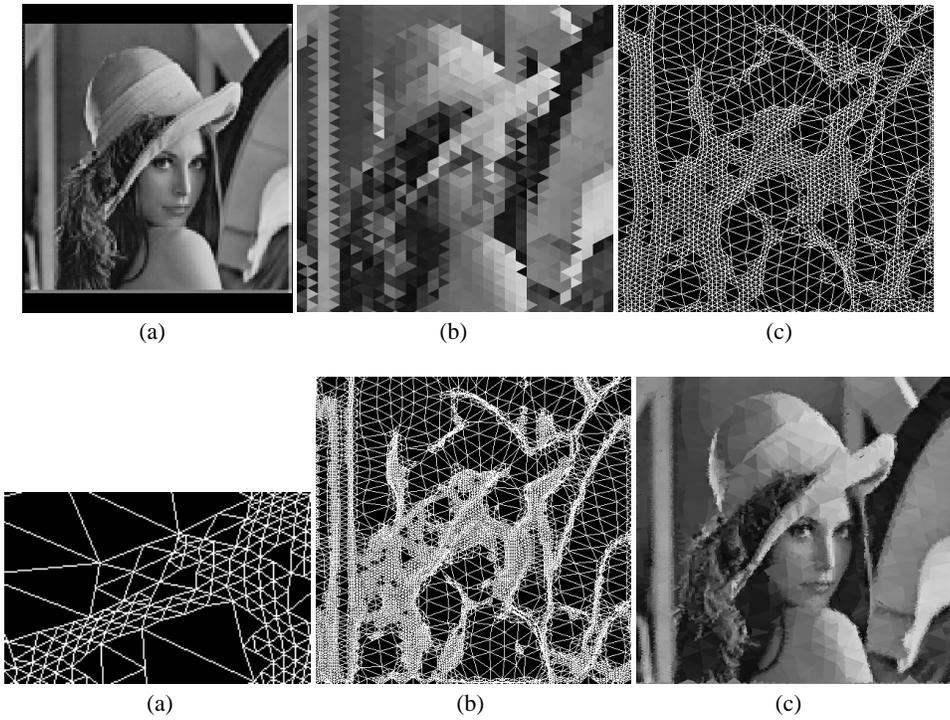


Figure 5: Hierarchical adaptive sampling and reconstruction of image data.

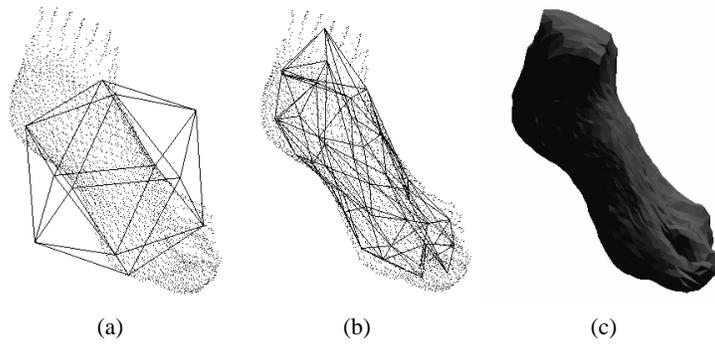


Figure 6: Reconstruction of a foot using an adaptive shell.

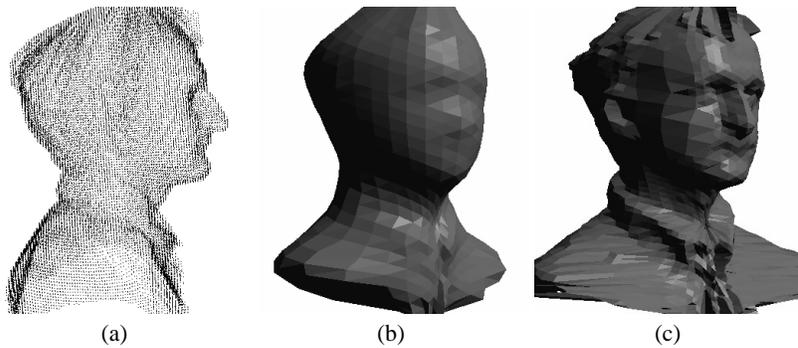


Figure 7: Reconstruction of a head using an adaptive shell.