

Sampling and Reconstruction with Adaptive Meshes

Demetri Terzopoulos^{1,3,4} and Manuela Vasilescu²

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

²MIT Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA, 02139

³Schlumberger Laboratory for Computer Science, P.O. Box 200015, Austin, TX 78720

Abstract

This paper introduces an approach to visual sampling and reconstruction motivated by concepts from numerical grid generation. We develop adaptive meshes that can nonuniformly sample and reconstruct intensity and range data. Adaptive meshes are dynamic models which are assembled by interconnecting nodal masses with adjustable springs. Acting as mobile sampling sites, the nodes observe interesting properties of the input data, such as intensities, depths, gradients, and curvatures. Based on these nodal observations, the springs automatically adjust their stiffnesses so as to distribute the available degrees of freedom of the reconstructed model in accordance with the local complexity of the input data. The adaptive mesh algorithm runs at interactive rates with continuous 3D display on a graphics workstation. We apply it to the adaptive sampling and reconstruction of images and surfaces.

1 Introduction

Visual reconstruction from noise corrupted data is a fundamental problem. Image reconstruction has occupied the image processing community for decades, and the surface reconstruction problem is presently attracting a significant amount of attention in computer vision (see, e.g., [1, 2, 3, 4] and references therein). Most approaches to surface reconstruction begin with a variational formulation of the problem, often based on generalized spline models defined on a bounded domain in the (x, y) image plane. This is followed by a discretization of the energy functional or of the associated partial differential equations and boundary conditions on a computational grid covering the domain, leading up to a numerical solution of the resulting system of algebraic equations using iterative techniques.

In conventional approaches to surface reconstruction, the continuous spline surface is represented as a single-valued function $z(x, y)$ and the computational grid is uniform, with its nodes in predetermined fixed positions on the image plane. The nodes on the surface can move only in the verti-

cal or z direction during the iterative reconstruction procedure.

We introduce a class of spatially adaptive sampling and reconstruction models whose nodes are also free to move over the (x, y) visual domain. This permits the nodes to migrate on the evolving surface during reconstruction. Enhanced representational power accrues from the adaptive placement of nodes in accordance with the complexity of the input data and the reconstructed surface. Clearly, it is beneficial to concentrate nodal degrees of freedom where they will do the most good—in regions where the data suggest rapid variations in shape; e.g., in regions where gradients or curvatures are large.

In this paper we develop *adaptive meshes* for nonuniformly sampling and reconstructing input data. Adaptive meshes are dynamic models assembled as topologically regular collections of nodal masses connected by adjustable springs. Acting as mobile sampling sites, the nodes observe interesting properties of the input data, such as depths, gradients, and curvatures. The springs automatically adjust their stiffnesses based on the locally sampled information in order to concentrate nodes near rapid shape variations. In this way, the meshes automatically adapt to the local variations in the input data and/or the reconstructed solution. Reconstruction using an adaptive mesh requires the solution of a second-order system of nonlinear ordinary differential equations. We are able to perform the simulation very efficiently using an explicit time integration procedure.

Our approach is generally applicable to the adaptive sampling and reconstruction of arbitrary data sets in any number of dimensions. In addition to surface reconstruction, this paper also considers the nonuniform sampling and reconstruction of grey-level images. We propose an adaptive image sampling technique which appears to be simpler and more natural than some elaborate multistep schemes (c.f., e.g., [5, 6]).

1.1 Background

Numerical analysts have demonstrated the benefits of spatially (and temporally) adaptive approximation schemes for the solution of differential equations (see, e.g., [7])

⁴Fellow, Canadian Institute for Advanced Research.

Sec. 16.5). The primary motivation for our approach comes from the field of numerical grid generation which has evolved powerful tools for solving field problems, especially on irregularly shaped domains [8]. In numerical grid generation, nodes are treated as mobile observers or sampling stations. The nodes distribute themselves over the domain so as to represent the entire physical field with sufficient accuracy given the finite set of observations. Each node influences the other nodes of the grid, so that if one node moves to gain a better position for observation, its neighbors will follow to some extent to maintain smooth coverage of the field. Although our approach to image and surface reconstruction adopts the idea of grid adaptation and nodes as observers, the adaptive mesh model that we propose differs from the adaptive grids found in the numerical grid generation literature.

The vision literature mentions a few models that are loosely related to adaptive meshes. Weiss [9] argues for nonuniform knot placement in the context of curve and surface fitting to data points. He defines a variational principle which automatically concentrates knots near regions of high curvature of a reconstructed curve model, thus enabling the curve to follow these regions more accurately. The surface models proposed by Vemuri *et al.* [10] and by Terzopoulos *et al.* [11] also permit variations in node proximity and they will tend to position nodes so as to distribute curvature as evenly as possible. However, the aforementioned vision models make no attempt to distribute nodes optimally under the direct control of nodal observations of the data, a powerful idea that we have incorporated into adaptive meshes.

1.2 Overview

The remainder of this paper is organized as follows: Section 2 formulates dynamic meshes. Section 3 incorporates special features into these models to turn them into adaptive meshes. Section 4 applies adaptive meshes to the adaptive sampling and reconstruction of images, and Section 5 applies them to the reconstruction of surfaces from range data. Section 6 concludes the paper.

2 Dynamic Mesh Models

A dynamic mesh is a type of discrete model [12] constructed from physically-based nodes and springs.

2.1 Dynamic Nodes and Springs

We define node i , for $i = 1, \dots, N$, as having a mass m_i and 3-space position $\mathbf{x}_i(t) = [x_i(t), y_i(t), z_i(t)]$. The velocity of the node is $\mathbf{v}_i = d\mathbf{x}_i/dt$ and its acceleration is $\mathbf{a}_i = d^2\mathbf{x}_i/dt^2$. The dynamic node is subject to a net nodal force $\mathbf{f}_i^n(t)$ (see below).

Let spring ij , which connects node j to node i , have natural length l_{ij} and stiffness c_{ij} . Let $\mathbf{r}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ be the vector separation of the nodes. We define the force that the

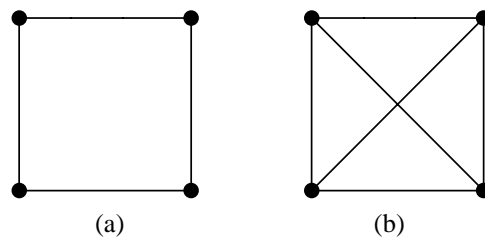


Figure 1: Quadrilateral elements. Black circles represent nodes, lines represent springs. (a) Basic element. (b) Element with shear-resisting cross springs.

spring exerts on node i as

$$\mathbf{s}_{ij} = \frac{c_{ij}e_{ij}}{\|\mathbf{r}_{ij}\|} \mathbf{r}_{ij}, \quad (1)$$

where $\|\mathbf{r}_{ij}\|$ is the actual length of the spring and $e_{ij} = \|\mathbf{r}_{ij}\| - l_{ij}$ is the deformation of the spring. Note that the spring force is a nonlinear function of node positions because $\|\mathbf{r}_{ij}\|$ involves roots of sums of squares.

2.2 Mesh Assembly and Boundary Conditions

It is natural to consider spring ij along with the attached nodes i and j as a uniaxial (linear) finite element. By allowing them to share nodes, uniaxial elements may be assembled into more complex composite elements, which may in turn be assembled into dynamic meshes. The discrete nature of the elements makes it easy to assemble dynamic meshes with diverse topologies. We utilize quadrilateral elements for image and surface reconstruction, and these elements may include cross springs to afford resistance against shearing (Fig. 1).

For the purposes of this paper, we assemble quadrilateral elements into bounded surfaces. We also restrict our attention to rectangular visual domains such as the one illustrated in Fig. 2. The domain in the figure is covered by a dynamic mesh of quadrilateral elements in their initial, regular configurations. The positions \mathbf{x}_i of interior nodes are unconstrained, whereas boundary nodes are subject to constraining boundary conditions. The boundary conditions may be expressed as $\mathbf{B}_i \mathbf{f}_i^n$, the application of a projection operator \mathbf{B}_i to the net nodal force of boundary nodes. The operator cancels the component of \mathbf{f}_i^n parallel to the image plane and normal to the boundary. This ensures that the x_i and y_i components of the positions of boundary nodes \mathbf{x}_i are constrained so as to maintain these nodes over the boundary, while allowing them the freedom to slide like beads along the boundary.

2.3 Equations of Motion

The discrete Lagrange equations of motion for the dynamic node/spring system is the set of coupled second-order ordi-

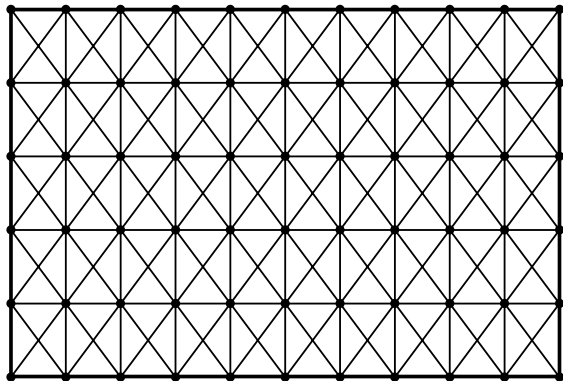


Figure 2: A mesh of quadrilateral elements with cross springs. Interior nodes move freely in (x, y, z) space. Only z coordinates of boundary nodes are free (z axis is perpendicular to page), while x and y coordinates are constrained such that boundary nodes may slide along boundary (bold lines).

nary differential equations

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

where \mathbf{f}_i is an external force applied at node i and

$$\mathbf{g}_i(t) = \sum_{j \in \mathcal{N}_i} \mathbf{s}_{ij} \quad (3)$$

is the total force on node i due to springs connecting it to neighboring nodes j . In applications to visual reconstruction, \mathbf{f}_i may be used to couple the dynamic mesh to input data (see Sec. 4.1). The quantity γ_i is a velocity-dependent damping coefficient that controls the rate of dissipation of kinetic energy which eventually brings the mesh to rest.

2.4 Numerical Time-Integration Procedure

To simulate the dynamics of the adaptive mesh, we provide initial positions \mathbf{x}_i^0 and (usually zero) initial velocities \mathbf{v}_i^0 for each node i and numerically integrate the equations of motion forward through time until the mesh stabilizes: $\mathbf{v}_i \approx \mathbf{a}_i \approx \mathbf{0}$.

At each time step t we evaluate the current nodal forces and accelerations, the new velocities, and the new positions using the explicit Euler time-integration procedure [7]

$$\begin{aligned} \mathbf{f}_i^{nt} &= \mathbf{f}_i^t - \gamma_i \mathbf{v}_i^t - \mathbf{g}_i^t; \\ \mathbf{a}_i^t &= \frac{\mathbf{B}_i \mathbf{f}_i^{nt}}{m_i}; \\ \mathbf{v}_i^{t+\Delta t} &= \mathbf{v}_i^t + \Delta t \mathbf{a}_i^t; \\ \mathbf{x}_i^{t+\Delta t} &= \mathbf{x}_i^t + \Delta t \mathbf{v}_i^{t+\Delta t}, \end{aligned} \quad (4)$$

for $i = 1, \dots, N$ and $t = 0, \Delta t, 2\Delta t, \dots$. A convenient way to compute the net nodal forces is to think of the \mathbf{f}_i^{nt} variables as nodal force accumulators. Each spring in the mesh makes the appropriate force contributions into the variables of the two nodes to which it is connected. Clearly, the force computation is parallelizable.

3 Adaptive Meshes

This section develops adaptive meshes by incorporating an adaptation process into dynamic meshes.

3.1 Adaptation Functions and Nodal Observations

The nodes of the adaptive mesh are able to make local observations about the input data. We denote a continuous input data set by $\mathbf{d}(\boldsymbol{\xi})$, where $\boldsymbol{\xi}$ is a point in the coordinate system spanning the data. The formulation in this section accommodates vector-valued data in general, but the applications to image and surface reconstruction illustrate the special case of scalar data.

We first transform the data into a scalar *adaptation function* $a_{\mathbf{d}}(\boldsymbol{\xi})$, normalized so that its values fall in the range $[0, 1]$. The transformations applied determine how the mesh will adapt to the data. A useful type of adaptation function is the convolution

$$a_{\mathbf{d}}(\boldsymbol{\xi}) = G(\boldsymbol{\xi}) * H(\mathbf{d}(\boldsymbol{\xi})), \quad (5)$$

where G is a normalized spatial smoothing filter and $H(\mathbf{d})$ is some function of the partial derivatives of the data field $\partial_{\boldsymbol{\xi}}^k \mathbf{d}$ of order $k \geq 0$.

Let $O_i(\mathbf{d})$ be a nodal observation; i.e., a measurement on \mathbf{d} made by node i through the adaptation function. We define

$$O_i(\mathbf{d}) = a_{\mathbf{d}}(\Pi \mathbf{x}_i), \quad (6)$$

where Π is an operator that projects the position \mathbf{x}_i of node i into the data coordinate system.

We design the adaptation function so that larger values of O_i will indicate that, from its vantage point \mathbf{x}_i , node i is observing “interesting features” in the data, such as rapid variations in intensity or shape.

3.2 Mesh Adaptation

The adaptive mesh incorporates a feedback procedure which automatically adjusts spring parameters c_{ij} according to the observations made at the nodes to which they are attached. Generally, it makes sense to increase the stiffness of spring ij with increasing O_i and O_j , which indicates that the two associated nodes are observing interesting features in the data, and decrease the stiffness otherwise. The intended effect is for the node density to increase around interesting observations, at the expense of the density over regions in the data where interesting features are absent.

A simple and effective relationship is

$$c_{ij}^t = (1 - \rho_{ij})c_{\min} + \rho_{ij}c_{\max}, \quad (7)$$

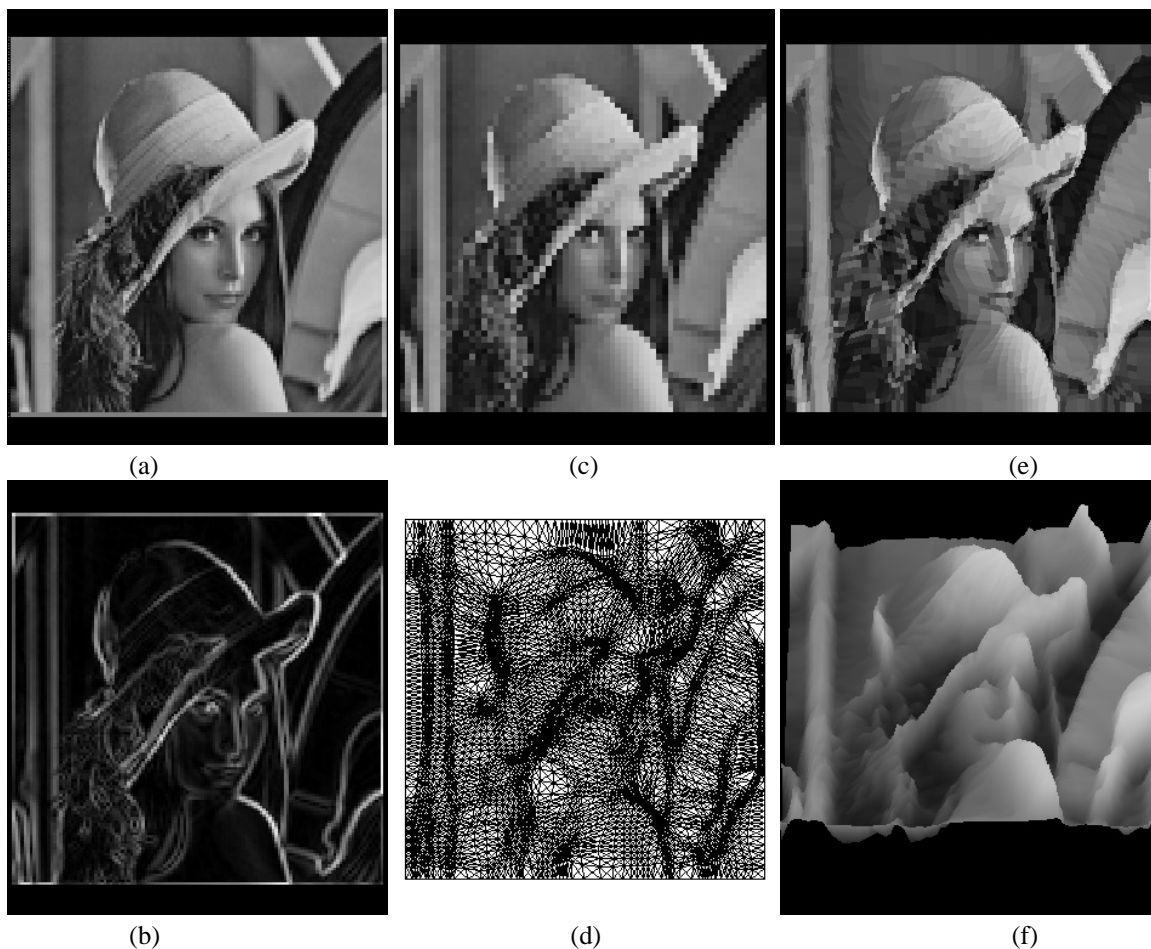


Figure 3: Adaptive sampling and reconstruction of image data.

(a) Lenna Sjööblom grey-level image. (b) Gradient adaptation function. (c) Image sampled on 64×64 regular mesh. (d) Planar adapted mesh (no data forces). (e) Adaptively sampled image. (f) Oblique view of reconstructed image intensity function (data force strength $\alpha = 10.0$). Simulation parameters: $m_i = 1.9$, $\gamma_i = 0.2$, $l_{ij} = 1.0$, $c_{\min} = 1.0$, $c_{\max} = 50.0$, $\Delta t = 0.05$.

with

$$\rho_{ij} = \frac{1}{2}(O_i^t + O_j^t), \quad (8)$$

where c_{\min} and c_{\max} are the minimum and maximum allowable values for the spring constant. Equations (7) and (8) are computed at each time step t of (4) as part of the computation of \mathbf{g}_i^t . Since the springs will repel nodes when they are compressed smaller than their natural lengths l_{ij} , we can set these lengths to control the minimal distances between nodes.

4 Adaptive Image Reconstruction

Our goal in this section is to sample an image at a reduced rate and reconstruct it using an adaptive mesh. The fidelity of the reconstruction is enhanced by concentrating the nodes of the mesh where the image function is changing

rapidly. We therefore adapt to the image gradient such that the node density increases in high-gradient regions.

Starting with a digital image $d(k, l)$, we compute a discrete version of (5), choosing G to be an iterated 5-point approximation to a Gaussian and $H(d) = \|\nabla d\|$, where ∇ denotes the discrete gradient operator. We normalize the convolution to obtain a discrete adaptation function $a_d(k, l)$ whose range is $[0, 1]$. Projecting node positions \mathbf{x}_i into the adaptation function using the orthographic projection $\xi_i = \Pi \mathbf{x}_i = (x_i, y_i)$, the nodes compute the observations (c.f. (6))

$$O_i(d) = a_d(x_i, y_i), \quad (9)$$

which is computed by bilinearly interpolating the values of $a_d(k, l)$ at the four pixels surrounding (x_i, y_i) .

We tested the adaptive mesh algorithm on the standard image in Fig. 3(a). Fig. 3(b) shows the adaptation func-

tion. Fig. 3(c) shows the image intensity function sampled at the nodes of a 64×64 regular mesh. Fig. 3(d) shows the adaptive mesh at equilibrium after it has adapted to the image. The nodes of the adapted mesh are clearly denser in high contrast regions. Fig. 3(e) shows the image sampled nonuniformly by the adaptive mesh. Both Fig. 3(e) and Fig. 3(c) represent an image compression ratio of 64 : 1, but note the difference in quality.

4.1 Data Forces

To reconstruct the image intensity function itself, we apply image data forces \mathbf{f}_i in (2) which deflect the mesh perpendicular to the image plane so that its 3D shape becomes consistent with the image intensity surface. More specifically, we introduce into (4) an external data force at time t which acts along the z axis:

$$\mathbf{f}_i^t = [0, 0, \alpha (d(\Pi\mathbf{x}_i^t) - z_i^t)], \quad (10)$$

where α is a scaling parameter and $d(\Pi\mathbf{x}_i) = d(x_i, y_i)$ is computed by bilinear interpolation as in (9). Equation (10) may be visualized as a vertical spring with one end attached to node i and the other end able to slide along the data surface. The degree to which the mesh approximates the image surface is inversely related to the magnitude of α , smaller values producing a greater smoothing effect on the noise in the data.

Note that no data forces were used to compute Fig. 3(d), consequently the adapted mesh is planar. Fig. 3(f) shows an oblique view of the adaptive mesh at equilibrium after the image data forces have been turned on. Here, the mesh has reconstructed the image surface and its elevation is proportional to intensity. Image intensity is sampled by the nodes and painted with bilinear interpolation onto the elements of the adaptive mesh to produce a shaded surface. The value of α has been set for moderate smoothing (refer to figure caption for simulation parameter values).

5 Adaptive Surface Reconstruction

This section demonstrates the adaptive reconstruction of surfaces from regularly sampled laser range data. Our goal is to automatically reconstruct 3D meshes which have the properties of manually tailored meshes used to represent CAD models. For reasons of convenience and efficiency, a CAD mesh must have significantly lower resolution than a typical range data array. To compensate for their limited resolution, CAD meshes are usually adapted to the shape details of models. In our demonstration of adaptive surface reconstruction, we enable a relatively low-resolution mesh to adapt to range data so as to elicit shape details by automatically increasing its node density in high curvature regions. We also apply data forces that deflect the mesh from its planar initial shape into a sculpted 3D surface consistent with the range data.

To derive an adaptation function a_d from the 2D scalar-valued range data array $d(k, l)$, we apply a discretized version of (5) as described in Sec. 4, but with $H(d) = \|\nabla \|\nabla d\|\|$, and again compute the observations as in (9). Note that O_i increase with increasing curvature in the range field. Once again, we employ data forces (10).

We tested the adaptive mesh on a 128×128 pixel range image of a statuette (Fig. 4(a)), a reduced version of STAT 1 (CAT # 155) taken from the NRCC 3D image database [13]. Fig. 4(b) shows a 30×30 node adaptive mesh in its initial regular and planar configuration superimposed on the adaptation function derived from the range data. Fig. 4(c) shows the mesh at equilibrium, adapted to the range data. Here, the adapted mesh is planar because data forces were turned off ($\alpha = 0$ in (10)) and the nodes of the adapted mesh are clearly denser in areas of high curvature of the statuette data. Fig. 4(d) and (e) show as a wireframe and as a flat-shaded 3D surface model, respectively, the reconstructed surface of the statuette obtained at equilibrium after the data forces are turned on.

6 Conclusion

We have developed a new approach to visual sampling and reconstruction motivated by concepts from numerical grid generation. We introduced adaptive mesh models that nonuniformly sample and reconstruct input intensity or range images. Adaptive meshes are dynamic models assembled from nodal masses connected by adjustable springs. Acting as mobile sampling sites, the nodes observe interesting properties of the input data, such as intensities, depths, gradients, and curvatures. The springs automatically adjust their stiffnesses based on the locally sampled information in order to concentrate nodes near rapid variations in the input data. The representational power of an adaptive mesh stems by its ability to optimally distribute the available degrees of freedom of the reconstructed model in accordance with the local complexity of the input data.

We have implemented the adaptive mesh algorithm in C on a Silicon Graphics 4D-240VGX workstation. Even for fairly large meshes (upwards of 4000 nodes), the simulation runs at interactive rates on a single CPU with continuous 3D display of the mesh at every time step as it adapts to input data. The user can interactively alter the viewpoint, shading mode, and other visualization parameters as the simulation is running.

We have demonstrated the adaptive sampling and reconstruction approach by applying it to intensity and range images. However, our approach is generally applicable to the adaptive sampling and reconstruction of arbitrary dimensional data sets. The nodes can sample both the evolving solution as it is being reconstructed, and they can directly sample arbitrary local properties of the input data, including positions, gradients, curvatures, etc., and in different applications, intensity, color, and texture information as well.

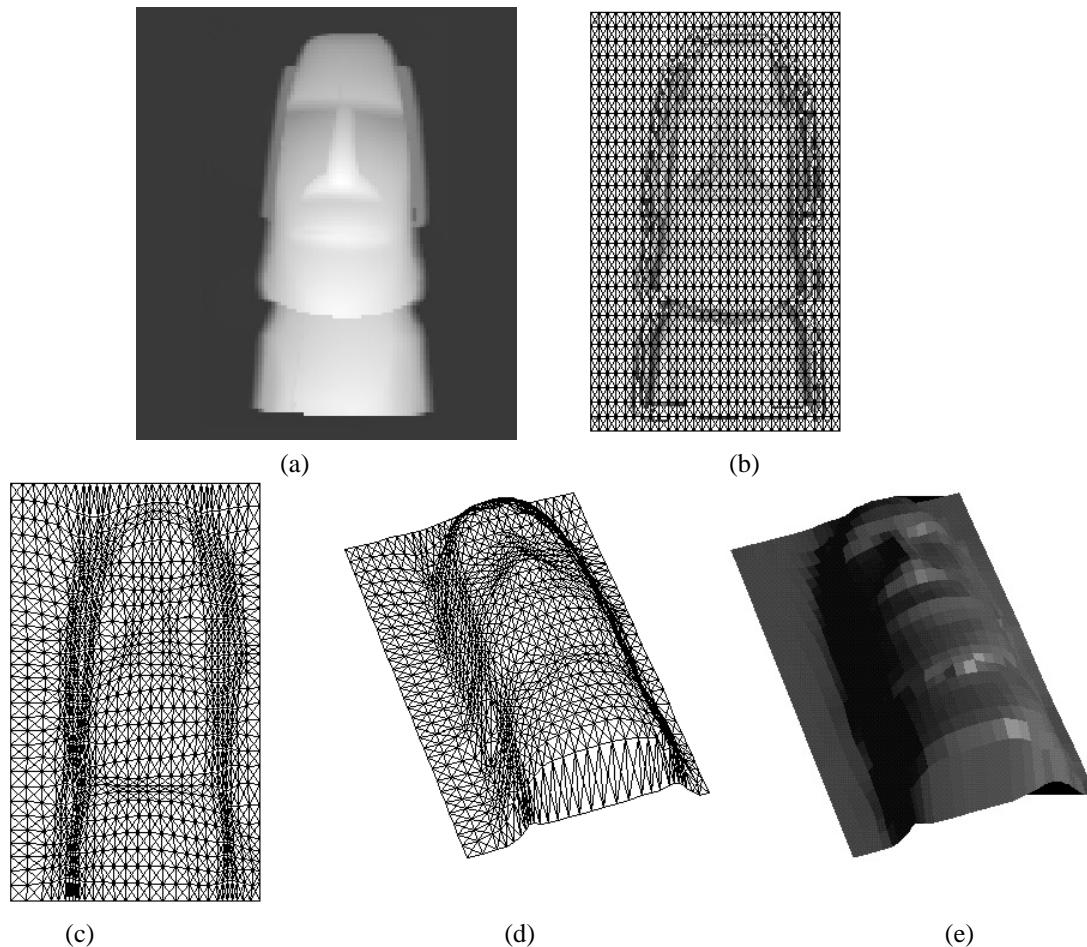


Figure 4: Adaptive sampling and reconstruction of range data.

(a) Range data image. (b) Initial regular mesh superimposed over adaptation function (see text). (c) Adapted planar mesh (no data forces). (d) 3D adapted mesh (with data forces, $\alpha = 10.0$). (e) Flat-shaded rendition of (d). Simulation parameters: $m_i = 0.02$, $\gamma_i = 0.9$, $l_{ij} = 1.0$, $c_{\min} = 1.0$, $c_{\max} = 10.0$, $\Delta t = 0.01$.

References

- [1] D. Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):417–438, 1988.
- [2] R. Szeliski. *The Modeling of Uncertainty in Low-Level Vision*. Kluwer Academic, Reading, MA, 1989.
- [3] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.
- [4] W. E. L. Grimson. *From Images to Surfaces: A Computational Study of the Human Early Visual System*. MIT Press, Cambridge, MA, 1981.
- [5] C.-H. Lee. Image surface approximation with irregular samples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):206–212, 1989.
- [6] D. G. McCaughey and Andrews H. C. Image approximation by variable knot bicubic splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(3):299–310, 1981.
- [7] W. H. Press, B. P. Fannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK, 1986.
- [8] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin. *Numerical Grid Generation: Foundations and Applications*. North-Holland, New York, 1985.
- [9] I. Weiss. Shape reconstruction on a varying mesh. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(4):345–362, 1990.
- [10] B. C. Vemuri, A. Mitiche, and J. K. Aggarwal. Curvature-based representation of objects from range data. *Image and Vision Computing*, 4:107–114, 1986.
- [11] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.
- [12] D. Greenspan. *Discrete Models*. Addison-Wesley, Reading, MA, 1973.
- [13] M. Rioux and L. Cournoyer. The NRCC three-dimensional image data files. Technical Report CNRC No. 29077, National Research Council of Canada, Ottawa, 1988.