Sampling and Reconstruction with Adaptive Meshes

Demetri Terzopoulos and M. Vasilescu

Presented by Yuan Chen

Outline

Problem

- Adaptively sample the data to lower resolution
  --- for compression or simplification
- Conventional approaches: uniformly sampling

Background (Numerical Grid Generation)

- A tool for numerical solution of partial differential equations on arbitrarily shaped regions.
- The numerical solution is built on those grids.
- Make best use of those finite grids to represent the physical solution with sufficient accuracy.
- The grid is influenced by both the geometric configuration and by the physical solution being done thereon.

Dynamic Mesh

- Essential component of adaptive mesh
- Constructed from physically-based nodes and springs
- Finite elements: spring and its attached nodes

Dynamic Mesh (Node)

- Node: i, i = 1, ….N
  -- mass: m_i
  -- position : $X_i(t) = [x_i(t), y_i(t), z_i(t)]$
  -- velocity: $v_i = dX_i / dt$
  -- acceleration: $a_i = d^2 X_i / dt^2$
  -- force $f_i^n(t)$
- Be subject to a net nodal force: $f_i^n(t)$

Dynamic Mesh (Spring)

- Spring: *ij*
  -- connect node *j* to node *i*
  -- natural length l_{ij}
  -- stiffness c_{ij}
- Force exerted on node *i*: *i* ●——● *j*

$$s_{ij} = \frac{c_{ij} e_{ij}}{\|r_{ij}\|} r_{ij},$$

$$r_{ij} = X_j - X_i,\ eij = \|rij\| - lij$$

Dynamic Mesh (Motion equations)

- For each node:

$$m_i \frac{d^2 X_i}{dt^2} + \gamma_i \frac{dX_i}{dt} + g_i = f_i;\ i = 1, \ldots, N$$

m_i : mass X_i : position γ_i : damping coefficient
g_i : the total force exerted from all connected springs
f_i : an external force applied at node *i*

Dynamic Mesh (Numerical time-integration)

- At each time step t:

$$1.\ f_i^{nt} = f_i^t - \gamma_i v_i^t - g_i^t$$

$$2.\ a_i^t = \frac{B_i f_i^{nt}}{m_i}$$

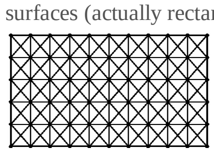
$$m_i \frac{d^2 X_i}{dt^2} + \gamma_i \frac{dX_i}{dt} + g_i = f_i;\ i = 1, \ldots, N$$

$$3.\ v_i^{t+\Delta t} = v_i^t + \Delta t a_i^t$$

$$4.\ X_i^{t+\Delta t} = X_i^t + \Delta t v_i^{t+\Delta t}$$

Dynamic Mesh (Utilized in this paper)

- Quadrilateral elements are assembled into bounded surfaces (actually rectanglar)



- Boundary condition: The boundary nodes can only move along the boundary in (x,y) visual domain

Adaptive Meshes

- Dynamic meshes with an adaptation function

-- stiffness of springs are changed according to the adaptation function during the process.

-- stiffness of spring increases in regions with rapid variation.

Adaptive Meshes (Adaptation functions)

- Adaptation function, $\alpha^d(\xi) \in [0,1]$

$$\alpha^d(\xi) = G(\xi) * H(d(\xi))$$

G is a normalized spatial filter; H is some function of partial derivative of the input data field.

- Nodal Observation, $O^t(d)$

$$O^t(d) = \alpha^d(\Pi X^t)$$

Adaptive Meshes (Feedback procedure)

- Feedback procedure

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

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

More interesting area → Higher H_i^t & H_j^t → larger O_i^t & O_j^t → larger c_{ij}^t

Algorithm

$$m_i \frac{d^2 X_i}{dt^2} + \gamma_i \frac{dX_i}{dt} + g_i = f_i;\ i = 1, \ldots, N$$

- Input: A 2D scalar-valued intensity or range image *d(k,l)*
- At each time step *t* :
  1. Evaluate the adaptation function and nodal observation for each node *i*
  2. Adjust the stiffness for each spring *ij*
  3. Compute the total force from springs for each node *i*
  4. Evaluate the current nodal forces, acceleration, the new velocity, and the new position.

Algorithm (Data forces)

- External data force at time t:

$$f_i^t = [0, 0, \alpha(d(\Pi X_i^t) - z_i^t)]$$

- The only trigger to shift nodes in z direction.
- For surface reconstruction, it deflects the mesh to fit the input data.

Results

(Image Reconstruction)

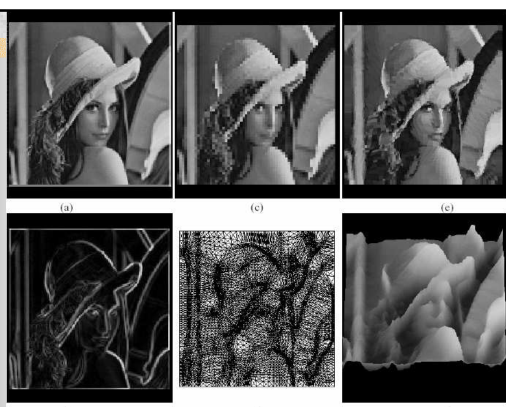


Image from D. Terzopoulos & M. Vasilescu

Results

(Surface Reconstruction)

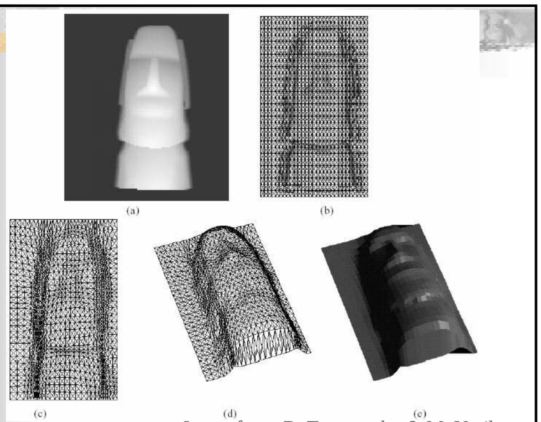


Image from D. Terzopoulos & M. Vasilescu

Advantages

- Applicable to arbitrary dimensional data sets
- Nodes are optimally distributed to preserve the interesting properties of the input data.
- Interactive rates (no actual timing data)

Disadvantages

- Not suitable for arbitrary mesh in 3D
  -- Assume a planar surface as initial guess
- No theoretical justification