A simple approach to mesh deformation

Shingo Ito

s1200045

February 8, 2016

Introduction

- Present a simple approach for mesh deformation.
- Sampling of the deformed mesh obtained from applying a surface reconstruction algorithm to the deformed point cloud.

Related works

- Surface deformation is an important research topic in shape and geometric modeling.
- The main technique consists in computing an approximation of the distance to the zero level-set and deforming this distance field.

Algorithm overview

Surface deformation work by moving the triangle mesh vertices according to some specified vector field corresponding to the deformation.

- Sample from the surface
- Apply the deformation field to the samples
- Apply a surface reconstruction algorithm to the samples

Surface sampling

Uniform sampling from a triangle

- Function Sample(p_1, p_2, p_3, ξ_1, ξ_2)
  p_1, p_2 and p_3 are the triangle vertices, ξ_1 and ξ_2 are samples from a uniform distribution
  Compute the barycentric coordinates: $u = 1 - \sqrt{\xi_1}$ and $v = \xi_2\sqrt{\xi_1}$
  Compute the sample:
  $P = u * p_1 + v * p_2 + (1 - u - v) * p_3$
  return P
  end function

Points deformation

Given a selected vertex from the input surface.
Apply a truncated Gaussian centered on this selected vertex

$$f(x) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(\mathbf{x}-\mathbf{x}_S)^2}{2\sigma^2}\right) & \text{if } \| x - \mathbf{x}_S \| < \epsilon \\ 0 & \text{otherwise} \end{cases}$$

Surface reconstruction

Implemented and experimented with three surface reconstruction algorithms.
All approaches are implicit surface based surface reconstruction algorithms

$$f : \mathbb{R}^3 \to \mathbb{R}$$

Hermite Radial Basis Functions

Given a set of points $P = \mathbf{x}_i$ with normal vector at each point $N = \mathbf{n}_i$, the output of the method is a function: $f : \mathbb{R}^3 \to \mathbb{R}$ defined as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{n} (\alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) - \beta_i \nabla \phi(\|\mathbf{x} - \mathbf{x}_i\|))$$

Hermite Radial Basis Functions

The conditions used to determine the coefficients are:

$$f(\mathbf{x}_i) = 0, \mathbf{x}_i \in \mathbb{R}^3$$

for interpolating points on the surface. And:

$$\nabla f(\mathbf{x}_i) = \mathbf{n}_i, \mathbf{x}_i \in \mathbb{R}^3, \mathbf{n}_i \in \mathbb{R}^3$$

for interpolating the normals on the surface.

Hermite Radial Basis Functions

$$f(\mathbf{x}_i) = \sum_{j=1}^{n} (\alpha_j \phi(\|\mathbf{x}_i - \mathbf{x}_j\|) - \beta_j \nabla \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)) = 0$$

$$\nabla f(\mathbf{x}_i) = \sum_{j=1}^{n} (\alpha_j \nabla \phi(\|\mathbf{x}_i - \mathbf{x}_j\|) - H\phi(\|\mathbf{x}_i - \mathbf{x}_j\|)\beta_j) = \mathbf{n}_i$$

where H is the Hessian matrix of ϕ.

Hermite Radial Basis Functions

$$\sum_{j=1}^{n} \begin{bmatrix} \phi(\|\mathbf{x}_i - \mathbf{x}_j\|) & -\nabla\phi(\|\mathbf{x}_i - \mathbf{x}_j\|)^T \\ \nabla\phi(\|\mathbf{x}_i - \mathbf{x}_j\|) & -H\phi(\|\mathbf{x}_i - \mathbf{x}_j\|) \end{bmatrix} \begin{bmatrix} \alpha_j \\ \beta_j \end{bmatrix} + \sum_{l=1}^{m} \lambda_l \begin{bmatrix} p_l(\mathbf{x}_i) \\ \nabla p_l(\mathbf{x}_i) \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{n}_i \end{bmatrix} \tag{1}$$

With the additional condition:

$$\sum_{j=1}^{n} [p_k(x_j) \nabla p_k(x_j)^T] \begin{bmatrix} \alpha_j \\ \beta_j \end{bmatrix} = 0 \tag{2}$$

Hermite Radial Basis Functions

$$\nabla\phi(\|x\|) = 3x\|x\|$$

$$H\phi(\|x\|) = \begin{cases} 3/\|x\|(\|x\|^2 I_{3\times3}) + xx^T, & \|x\| \neq 0 \\ 0_{3\times3}, & \|x\| = 0 \end{cases}$$

Closed form solution for compactly supported functions

$$f(x) = -\sum_{j=1}^{n} < \frac{\rho_j^2}{20 + \eta\rho_j^2}\mathbf{n}_j, \nabla\phi_{\rho_j}(\|\mathbf{x} - \mathbf{x}_j\|) > \qquad (3)$$

where ρ_j is the radius of support of the basis function associated to the center j, and ϕ is the compactly supported basis function.

$$\phi_\rho(r) = \phi(r/\rho)$$

where:

$$\phi(t) = \begin{cases} (1-t)^4(4t+1), & t \in [0,1] \\ 0, & \text{otherwise} \end{cases}$$

Poisson surface reconstruction

$$f_S(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in S \\ 0 & \textit{otherwise} \end{cases}$$

The indicator function f is obtained by solving the Poisson equation

$$\Delta f = div(\mathbf{n})$$

where **n** is an extrapolation of the given normal vector field.

Marching Cubes

Algorithm for rendering isosurfaces from volumetric data.

- Bounding box for the object to be meshed and subdivide it regularly into smaller cells.
- the function f is sampled at the eight corners of each cell.
- If one or more values is less than the user-specified iso-value, and one or more have values is greater than this isovalue, the cell must intersect the isosurface.

Marching Cubes

- Determining the edges in the cell that are intersected by the isosurface.
- connecting the patches from all cells, we get a linear approximation of the isosurface.

Delaunay based implicit surface meshing

- Marching Cubes based algorithm do not look like the most effective approach.
- Compute a Delaunay tetrahedralization of the deformed point-cloud.
- Peel off outside tetrahedra using the fitted function

The environment used for prototype and in the experiments

The development and all experiments were run on a regular note PC.

CPU	Intel Corei5 1.4GHz
GPU	Intel HD Graphics5000
Memory	4.0GB RAM
OS	OS X Yosemite ver- sion 10.10.5
Programming Language	C++
Libraries	CGAL4.7,OpenGL,Eigen

Deformation of a sphere

Start from a sphere repre-sented by a triangle mesh, made of 174 vertices and 344 triangles.

Figure: Deformed sphere using (from left to right): the HRBF approach, the closed form solution to the HRBF approach and the Poisson surface reconstruction approach

Example of sculpture

Figure: Example to model a simple character face

- The eyes and the mouth were carved by changing the sign of the field used for the deformation.
- Changing the width of the Gaussian function used for the deformation could allow to add or carve smaller details.

Deformation of a more complex surface

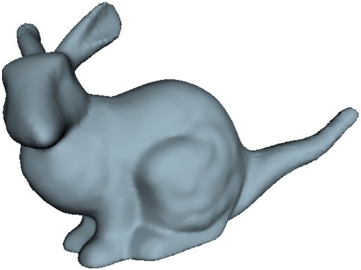

Figure: Deformed bunny object with stretched tail and nose

- The input mesh for this object contains 14072 vertices and 28042 triangles.
- Both the tail and the nose are stretched.

Conclusion

- Proposed a simple algorithm for mesh deformation.
- Reconstruction from the point-cloud is made by fitting an implicit surface to the point-cloud and meshing it.
- Future works : Implementing parts of the algorithm on the graphics card (GPU).

Thank you.