CS354 Computer Graphics
Point-Based Modeling

Qixing Huang
March 26th 2018

Slide Credit: Marc Alexa
Motivation

• Many applications need a definition of surface based on point samples
  – Reduction
  – Up-sampling
  – Ray tracing

• Desirable surface properties
  – Manifold
  – Smooth
  – Local (efficient computation)
Overview

• Introduction & Basics
• Fitting Implicit Surfaces
• Surfaces from Local Frames
Introduction & Basics

• Notation, Terms
  – Regular/Irregular, Approximation/Interpolation, Global/Local

• Standard interpolation/approximation techniques
  – Global: Triangulation, Voronoi-Interpolation, Least Squares (LS), Radial Basis Functions (RBF)
  – Local: Shepard/Partition of Unity Methods, Moving LS

• Problems
  – Sharp edges, feature size/noise

• Functional -> Manifold
Consider functional (height) data for now.

Data points are represented as:
- Location in parameter space $p_i$
- With certain height $f_i = f(p_i)$

Goal is to approximate $f$ from $f_i, p_i$. 
Terms: Regular/Irregular

- Regular (on a grid) or irregular (scattered)
- Neighborhood (topology) is unclear for irregular data
Terms: Approximation/Interpolation

- Noisy data $\Rightarrow$ Approximation

- Perfect data $\Rightarrow$ Interpolation
Terms: Global/Local

- Global approximation

- Local approximation

- Locality comes at the expense of fairness
Introduction & Basics

• Notation, Terms
  – Regular/Irregular, Approximation/Interpolation, Global/Local

• **Standard interpolation/approximation techniques**
  – Global: Triangulation, Voronoi-Interpolation, Least Squares (LS), Radial Basis Functions (RBF)
  – Local: Shepard/Partition of Unity Methods, Moving LS

• Problems
  – Sharp edges, feature size/noise

• Functional -> Manifold
Triangulation

- Exploit the topology in a triangulation (e.g. Delaunay) of the data
- Interpolate the data points on the triangles
  - Piecewise linear $\rightarrow C^0$
  - Piecewise quadratic $\rightarrow C^1$?
  - ...
Triangulation: Piecewise linear

- Barycentric interpolation on simplices (triangles)
  - given point $\mathbf{x}$ inside a simplex defined by $\mathbf{p}_i$
  - Compute $\alpha_i$ from
    $$\mathbf{x} = \sum_i \alpha_i \mathbf{p}_i \quad \text{and} \quad 1 = \sum_i \alpha_i$$
  - Then
    $$f(\mathbf{x}) = \sum_i \alpha_i f_i$$
Voronoi Interpolation

- compute Voronoi diagram (dual of Delaunay triangulation)
- for any point \( x \) in space
  - add \( x \) to Voronoi diagram
  - Voronoi cell \( \tau \) around \( x \) intersects original cells \( \tau_i \) of natural neighbors \( n_i \)
  - interpolate
    \[
    f(x) = \sum_i \lambda_i(x) f_i / \sum_i \lambda_i(x)
    \]

  with
  \[
  \lambda_i(x) = \frac{|\tau \cap \tau_i|}{|\tau| \cdot \|x - p_i\|}
  \]
Voronoi Interpolation

- Compute Voronoi diagram (dual of Delaunay triangulation)
- For any point \( x \) in space
  - Add \( x \) to Voronoi diagram
  - Compute weights from the areas of new cell relative to old cells
- Properties
  - Piecewise cubic
  - Differentiable, continuous derivative
Voronoi Interpolation

Properties of Voronoi Interpolation:

- linear Precision
- local
- \( f(x) \in C^1 \) on domain
- \( f(x,x_1,...,x_n) \) is continuous in \( x_i \)
Least Squares

- Fits a primitive to the data
- Minimizes squared distances between the $p_i$'s and primitive $g$

$$g(x) = a + bx + cx^2$$

$$\min_g \sum_i (f_i - g(p_i))^2$$
Least Squares - Example

- Primitive is a (univariate) polynomial
  \[ g(x) = (1, x, x^2, \ldots) \cdot c^T \]

- \[ \min \sum_i \left( f_i - (1, p_i, p_i^2, \ldots) c^T \right)^2 \Rightarrow \]

- \[ 0 = \sum_i 2p_i^j \left( f_i - (1, p_i, p_i^2, \ldots) c^T \right) \]

- Linear system of equations
Least Squares - Example

- Resulting system

\[ 0 = \sum_i 2p_i^j \left( f_i - \left(1, p_i, p_i^2, \ldots \right)^T \right) \iff \]

\[ \sum_i \begin{pmatrix} 1 & p_i & p_i^2 & \cdots \\ p_i & p_i^2 & p_i^3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \end{pmatrix} = 2 \sum_i f_i \begin{pmatrix} 1 \\ p_i \\ p_i^2 \\ \vdots \end{pmatrix} \]
Radial Basis Functions

- Solve 
  \[ f_j = \sum_{i} w_i r\left(\|p_i - p_j\|\right) \]
  to compute weights \( w_i \)
- Linear system of equations
  \[
  \begin{pmatrix}
  r(0) & r(\|p_0 - p_1\|) & r(\|p_0 - p_2\|) & \cdots \\
  r(\|p_1 - p_0\|) & r(0) & r(\|p_1 - p_2\|) \\
  r(\|p_2 - p_0\|) & r(\|p_2 - p_1\|) & r(0) \\
  \vdots & \vdots & \vdots & \ddots 
  \end{pmatrix}
  \begin{pmatrix}
  w_0 \\
  w_1 \\
  w_2 \\
  \vdots 
  \end{pmatrix}
  =
  \begin{pmatrix}
  f_0 \\
  f_1 \\
  f_2 \\
  \vdots 
  \end{pmatrix}
\]
Radial Basis Functions

- Represent approximating function as
  - Sum of radial functions \( r \)
  - Centered at the data points \( p_i \)

\[
 f(x) = \sum_{i} w_i r(\|p_i - x\|) 
\]
Radial Basis Functions

- Solvability depends on radial function
- Several choices assure solvability
  - \( r(d) = d^2 \log d \) (thin plate spline)
  - \( r(d) = e^{-d^2/h^2} \) (Gaussian)
    - \( h \) is a data parameter
    - \( h \) reflects the feature size or anticipated spacing among points
Function Spaces!

- Monomial, Lagrange, RBF share the same principle:
  - Choose basis of a function space
  - Find weight vector for base elements by solving linear system defined by data points
  - Compute values as linear combinations

- Properties
  - One costly preprocessing step
  - Simple evaluation of function in any point
Functional Spaces!

- Problems
  - Many points lead to large linear systems
  - Evaluation requires global solutions

- Solutions
  - RBF with compact support
    - Matrix is sparse
    - Still: solution depends on every data point, though drop-off is exponential with distance
  - Local approximation approaches
Introduction & Basics

• Notation, Terms
  – Regular/Irregular, Approximation/Interpolation, Global/Local

• **Standard interpolation/approximation techniques**
  – Global: Triangulation, Voronoi-Interpolation, Least Squares (LS), Radial Basis Functions (RBF)
  – **Local**: Shepard/Partition of Unity Methods, Moving LS

• Problems
  – Sharp edges, feature size/noise

• Functional -> Manifold
Shepard Interpolation

- Approach: \( f(x) = \sum_i \phi_i(x) f_i \)

  with basis functions

\[
\phi_i(x) = \frac{\|x - x_i\|^{-p}}{\sum_j \|x - x_j\|^{-p}}
\]

- define \( f(p_i) = f_i = \lim_{x \to p_i} f(x) \)
Shepard Interpolation

- $f(x)$ is a convex combination of $\phi_i$, because all $\phi_i \in [0,1]$ and $\sum \phi_i(x) = 1$
- $f(x)$ is contained in the convex hull of data points
- $|\{p_i\}| > 1 \Rightarrow f(x) \in C^\infty$ and $\nabla f(p_i) = 0$
  - Data points are saddles
- global interpolation
  - every $f(x)$ depends on all data points
- Only constant precision, i.e. only constant functions are reproduced exactly
Shepard Interpolation

Localization:
• Set \( f(x) = \sum_i \mu_i(x) \phi_i(x) f_i \)

• with \( \mu_i(x) = \begin{cases} (1 - \|x - p_i\|/R_i)\nu & \text{if } \|x - p_i\| < R_i \\ 0 & \text{else} \end{cases} \)

for reasonable \( R_i \) and \( \nu > 1 \)

→ no constant precision because of possible holes in the data
Partition of Unity Methods
Partition of Unity Methods

- Subdivide domain into cells
Partition of Unity Methods

- Compute local interpolation per cell
Partition of Unity Methods

• Blend local interpolations?
Partition of Unity Methods

- Subdivide domain into *overlapping* cells
Partition of Unity Methods

• Compute local interpolations
Partition of Unity Methods

- Blend local interpolations
Partition of Unity Methods

- Weights should
  - have the (local) support of the cell
Partition of Unity Methods

- Weights should
  - sum up to one everywhere (Shepard weights)
  - have the (local) support of the cell
Moving Least Squares

- Compute a local LS approximation at $x$
- Weight data points based on distance to $x$

\[ g(x) = a + bx + cx^2 \]

\[
\min \sum_i (f_i - g(p_i))^2 \theta(||x - p_i||) 
\]
Moving Least Squares

• The set

\[ f(x) = g_x(x), g_x : \min \sum_i (f_i - g(p_i))^2 \theta(\|x - p_i\|) \]

is a smooth curve, iff \( \theta \) is smooth
Moving Least Squares

- Typical choices for $\theta$:
  - $\theta(d) = d^{-r}$
  - $\theta(d) = e^{-d^2/h^2}$

- Note: $\theta_i = \theta(\|x - p_i\|)$ is fixed
- For each $x$
  - Standard weighted LS problem
  - Linear iff corresponding LS is linear
Introduction & Basics

• Notation, Terms
  – Regular/Irregular, Approximation/Interpolation, Global/Local

• Standard interpolation/approximation techniques
  – Global: Triangulation, Voronoi-Interpolation, Least Squares (LS), Radial Basis Functions (RBF)
  – Local: Shepard/Partition of Unity Methods, Moving LS

• Problems
  – Sharp edges, feature size/noise

• Functional -> Manifold
Typical Problems

- Sharp corners/edges
- Noise vs. feature size
Functional -> Manifold

- Standard techniques are applicable if data represents a function

- Manifolds are more general
  - No parameter domain
  - No knowledge about neighbors, Delaunay triangulation connects non-neighbors
Overview

- Introduction & Basics
- Fitting Implicit Surfaces
- Surfaces from Local Frames
Implicits

• Each orientable 2-manifold can be embedded in 3-space

• Idea: Represent 2-manifold as zero-set of a scalar function in 3-space
  – Inside: \( f(x) < 0 \)
  – On the manifold: \( f(x) = 0 \)
  – Outside: \( f(x) > 0 \)
Implicits from point samples

• Function should be zero in data points
  \[ f(p_i) = 0 \]

• Use standard approximation techniques to find \( f \)

• Trivial solution: \( f = 0 \)

• Additional constraints are needed
Implicits from point samples

- Constraints define inside and outside
- Simple approach (Turk, O’Brien)
  - Sprinkle additional information manually
  - Make additional information soft constraints
Implicits from point samples

- Use normal information
- Normals could be computed from scan
- Or, normals have to be estimated
Estimating normals

- Normal orientation (Implicits are signed)
  - Use inside/outside information from scan
- Normal direction by fitting a tangent
  - LS fit to nearest neighbors
  - Weighted LS fit
  - MLS fit
Estimating normals

- General fitting problem
  \[
  \min_{\|n\|=1} \sum_i^N \langle q - p_i, n \rangle^2 \theta(\|q - p_i\|)
  \]
  - Problem is non-linear because \(n\) is constrained to unit sphere
Estimating normals

- The constrained minimization problem

\[
\min_{\|n\|=1} \sum_i \left( \langle q - p_i, n \rangle \right)^2 \theta_i
\]

is solved by the eigenvector corresponding to the smallest eigenvalue of the following covariance matrix

\[
\sum_i (q - p_i) \cdot (q - p_i)^T \theta_i
\]

which is constructed as a sum of weighted outer products.
Implicits from point samples

- Compute non-zero anchors in the distance field
- Use normal information directly as constraints

\[ f(p_i + n_i) = 1 \]
Implicits from point samples

- Compute non-zero anchors in the distance field
- Use normal information directly as constraints

\[ f(p_i + n_i) = 1 \]
Implicits from point samples

- Compute non-zero anchors in the distance field
- Compute distances at specific points
  - Vertices, mid-points, etc. in a spatial subdivision
Computing Implicits

• Given N points and normals $p_i, n_i$ and constraints $f(p_i) = 0, f(c_i) = d_i$

• Let $p_{i+N} = c_i$

• An RBF approximation

$$f(x) = \sum_{i} w_i \theta(||p_i - x||)$$

leads to a system of linear equations
Computing Implicits

- Given $N$ points and normals $\mathbf{p}_i, \mathbf{n}_i$ and constraints $f(\mathbf{p}_i) = 0, f(\mathbf{c}_i) = d_i$

- Let $\mathbf{p}_{i+N} = \mathbf{c}_i$

- An RBF approximation

$$f(\mathbf{x}) = \sum_{i} w_i \theta(\|\mathbf{p}_i - \mathbf{x}\|)$$

leads to a system of linear equations
Computing Implicits

- Practical problems: $N > 10000$
- Matrix solution becomes difficult
- Two solutions
  - Sparse matrices allow iterative solution
  - Smaller number of RBFs
Computing Implicits

- Sparse matrices

\[
\begin{pmatrix}
\theta(0) & \theta(\|p_0 - p_1\|) & \theta(\|p_0 - p_2\|) & \cdots \\
\theta(\|p_1 - p_0\|) & \theta(0) & \theta(\|p_1 - p_2\|) & \cdots \\
\theta(\|p - p_0\|) & \theta(\|p_2 - p_1\|) & \theta(0) & \cdots \\
\cdots & \cdots & \cdots & \cdots
\end{pmatrix}
\]

- Needed: \(d > c \rightarrow r(d) = 0, r'(c) = 0\)

- Compactly supported RBFs
Computing Implicits

- Smaller number of RBFs
- Greedy approach (Carr et al.)
  - Start with random small subset
  - Add RBFs where approximation quality is not sufficient
RBF Implicits - Results

- Images courtesy Greg Turk
RBF Implicits - Results

- Images courtesy Greg Turk
Overview

• Introduction & Basics
• Fitting Implicit Surfaces
• Surfaces from Local Frames
Projection

- Idea: Map space to surface
- Surface is defined as fixpoints of mapping
Projection

• Projection procedure (Levin)
  – Local polynomial approximation
    • Inspired by differential geometry
  – "Implicit" surface definition
  – Infinitely smooth &
  – Manifold surface
Surface Definition

- Constructive definition
  - Input point $r$
  - Compute a local reference plane $H_r = \langle q, n \rangle$
  - Compute a local polynomial over the plane $G_r$
  - Project point $r' = G_r(0)$
  - Estimate normal
Surface Definition

- Constructive definition
  - Input point $\mathbf{r}$
  - Compute a local reference plane $H_r = \langle q, n \rangle$
  - Compute a local polynomial over the plane $G_r$
  - Project point $\mathbf{r}' = G_r(0)$
  - Estimate normal
Local Reference Plane

- Find plane 
  \[ H_r = \langle q, n \rangle + D \]
  
  \[ \min_{q, \|n\|=1} \sum_i \langle q - p_i, n \rangle^2 \theta(\|q - p_i\|) \]

- \[ \theta(d) = e^{d^2/h^2} \]
  - \( h \) is feature size/point spacing
  - \( H_r \) is independent of \( r \)'s distance
  - Manifold property

Weight function based on distance to \( q \), not \( r \)
Projecting the point

- MLS polynomial over $H_r$
  $$\min_{G \in \Gamma_d} \sum_i \left( \langle q - p_i, n \rangle - G(p_i|_{H_r}) \right)^2 \theta(||q - p_i||)$$
- LS problem
- $r' = G_r(0)$
- Estimate normal
Spatial data structure

- Regular grid based on support of $\theta$
  - Each point influences only 8 cells

- Each cell is an octree
  - Distant octree cells are approximated by one point in center of mass
Summary

• Projection-based surface definition
  – Surface is smooth and manifold
  – Surface may be bounded
  – Representation error mainly depends on point density
  – Adjustable feature size $h$ allows to smooth out noise