CS354 Computer Graphics
Point-Based Modeling

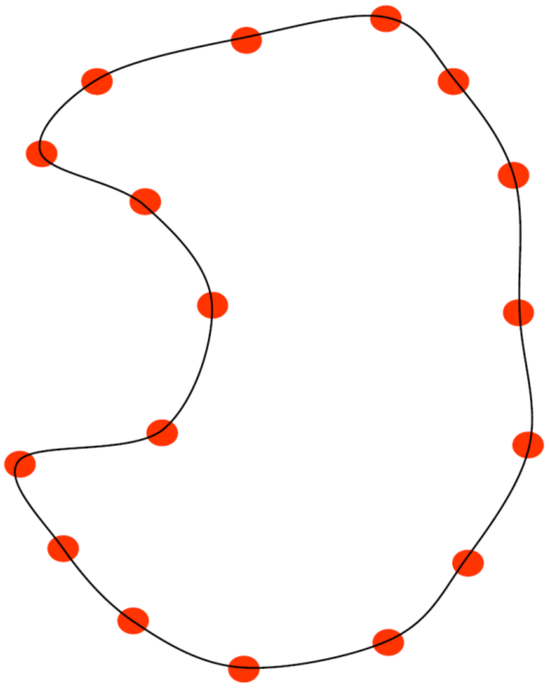
Qixing Huang

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

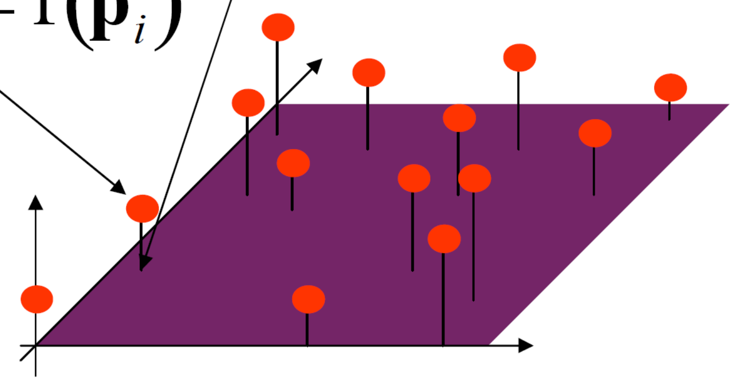
Notation

Consider functional (height) data for now

Data points are represented as

– Location in parameter space $\mathbf{p}_i$

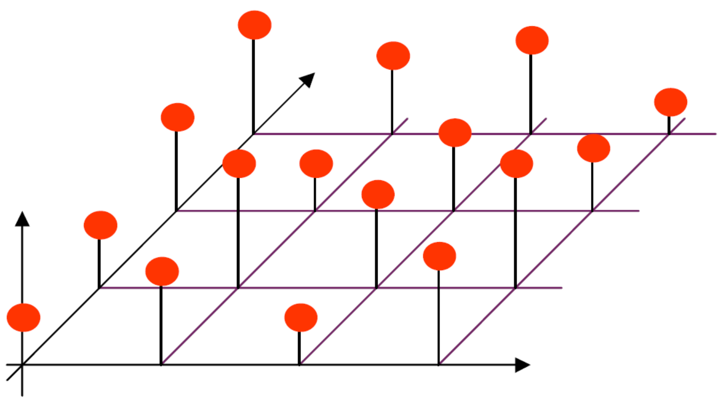
– With certain height $f_i = \mathrm{f}(\mathbf{p}_i)$

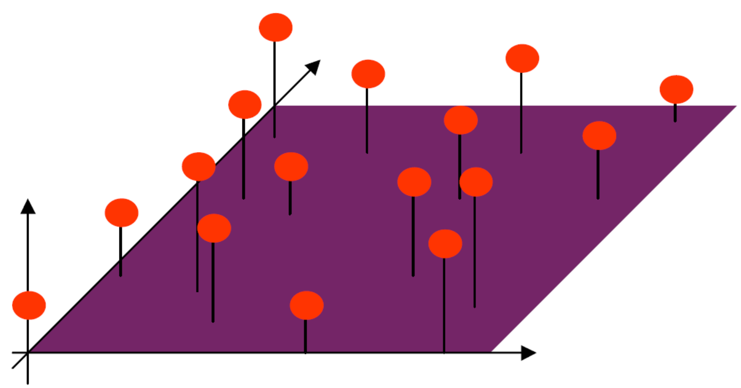


Goal is to approximate f from $f_i, \mathbf{p}_i$

Terms: Regular/Irregular

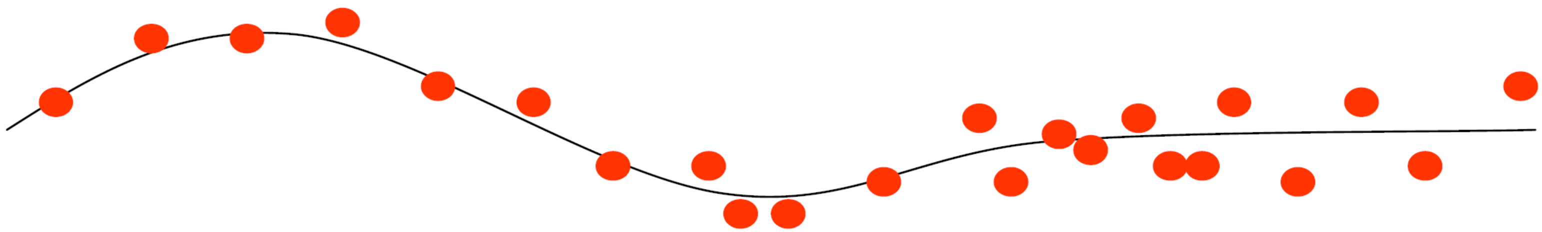
- Regular (on a grid) or irregular (scattered)
- Neighborhood (topology) is unclear for irregular data



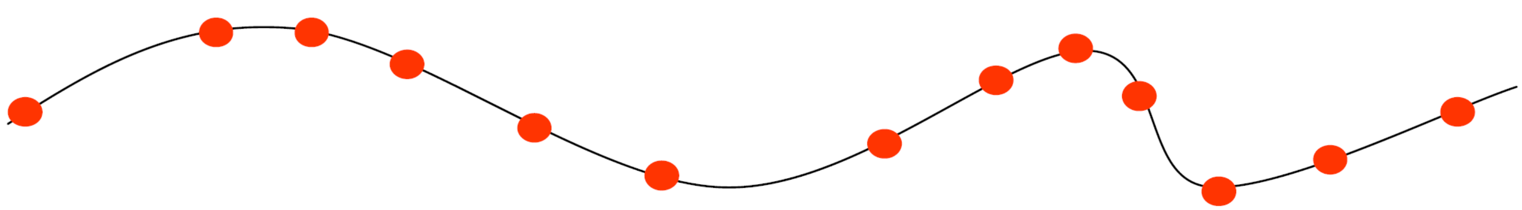


Terms: Approximation/Interpolation

- Noisy data $\Rightarrow$ Approximation

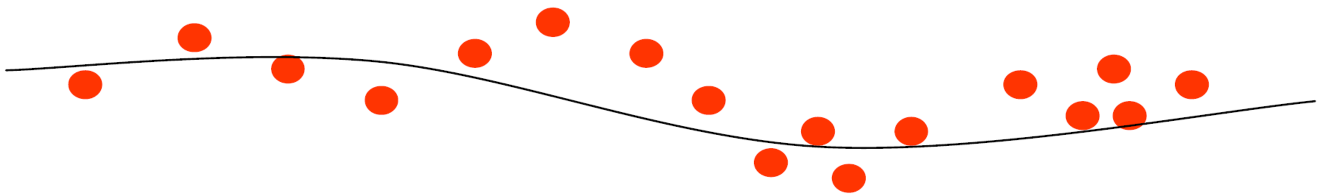


- Perfect data $\Rightarrow$ Interpolation

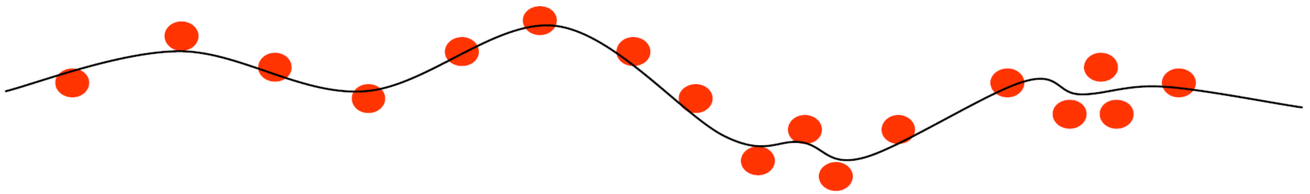


Terms: Global/Local

- Global approximation



- Local approximation



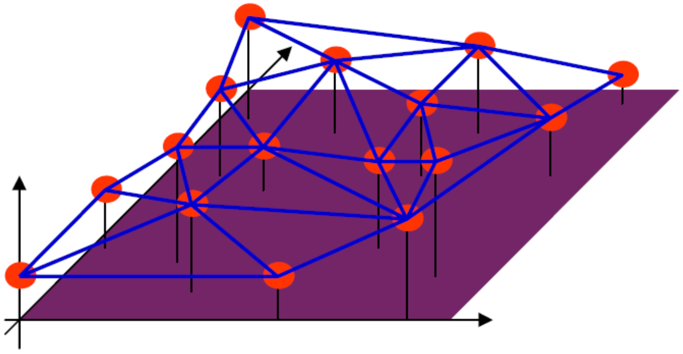
- Locality comes at the expense of fairness

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 → C^0
  - Piecewise quadratic → C^1?
  - …



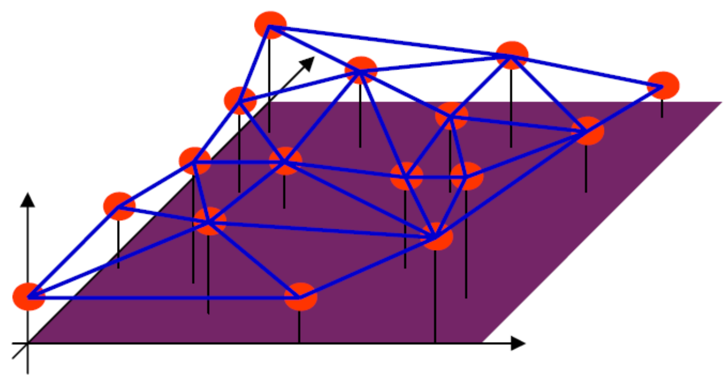
Triangulation: Piecewise linear

- Barycentric interpolation on simplices (triangles)
  - given point $\mathbf{x}$ inside a simplex defined by $\mathbf{p}_i$
  - Compute α_i from

$$\mathbf{x} = \sum_i \alpha_i \mathbf{p}_i \quad \text{and} \quad 1 = \sum_i \alpha_i$$

  - Then

$$f(\mathbf{x}) = \sum_i \alpha_i f_i$$



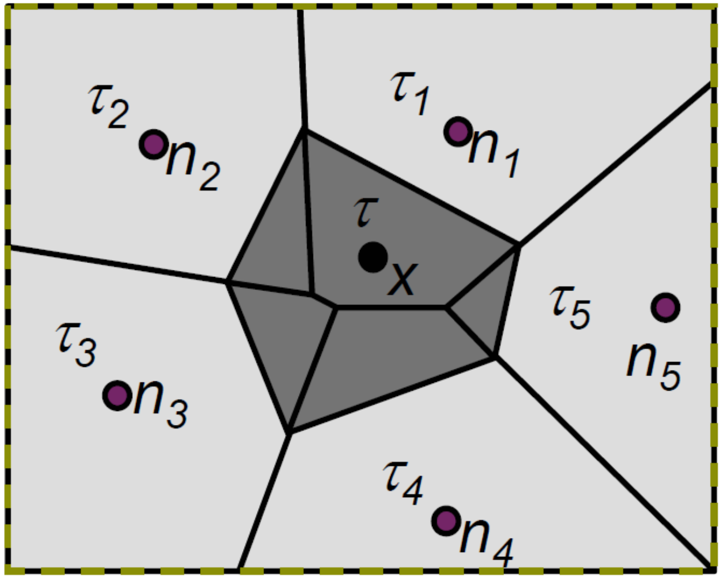
Voronoi Interpolation

- compute Voronoi diagram (dual of Delaunay triangulation)
- for any point $\mathbf{x}$ in space
  - add $\mathbf{x}$ to Voronoi diagram
  - Voronoi cell τ around $\mathbf{x}$ intersects original cells τ_i of natural neighbors n_i
  - interpolate $$f(\mathbf{x}) = \sum_i \lambda_i(x) f_i \Big/ \sum_i \lambda_i(x)$$

with $$\lambda_i(\mathbf{x}) = \frac{|\tau \cap \tau_i|}{|\tau| \cdot \|\mathbf{x} - \mathbf{p}_i\|}$$

Voronoi Interpolation

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



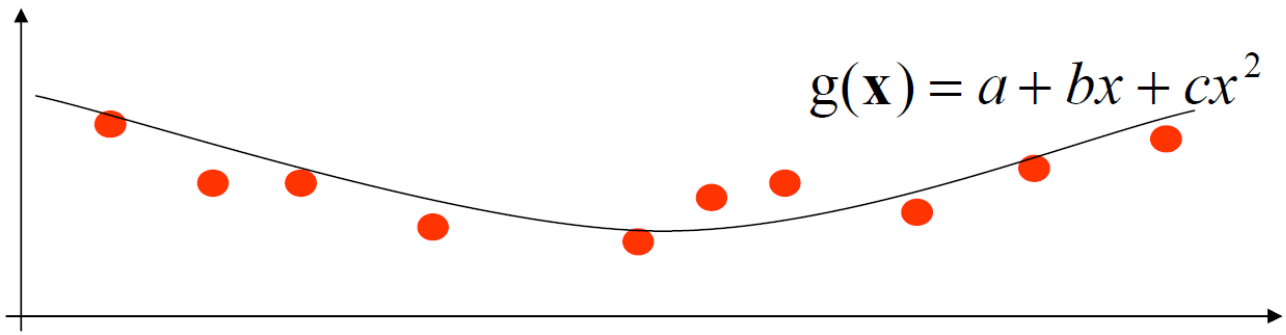
Voronoi Interpolation

Properties of Voronoi Interpolation:

- linear Precision
- local
- $f(\mathbf{x}) \in C^1$ on domain
- $f(\mathbf{x},\mathbf{x}_1,\ldots,\mathbf{x}_n)$ is continuous in $\mathbf{x}_i$

Least Squares

- Fits a primitive to the data
- Minimizes squared distances between the $\mathbf{p}_i$'s and primitive g



$$\min_{\mathrm{g}} \sum_i \left(f_i - \mathrm{g}(\mathbf{p}_i)\right)^2$$

Least Squares - Example

- Primitive is a (univariate) polynomial

$$g(x) = \left(1, x, x^2, \ldots\right) \cdot \mathbf{c}^T$$

- $$\min \sum_i \left(f_i - \left(1, p_i, p_i^2, \ldots\right) \mathbf{c}^T \right)^2 \Rightarrow$$

$$0 = \sum_i 2 p_i^j \left(f_i - \left(1, p_i, p_i^2, \ldots\right) \mathbf{c}^T \right)$$

- Linear system of equations

Least Squares - Example

- Resulting system

$$0 = \sum_i 2 p_i^j \left(f_i - \left(1, p_i, p_i^2, \ldots\right) \mathbf{c}^T \right) \Leftrightarrow$$

$$\sum_i \begin{pmatrix} 1 & p_i & p_i^2 & \cdots \\ p_i & p_i^2 & p_i^3 & \\ p_i^2 & p_i^3 & p_i^4 & \\ \vdots & & & \ddots \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \end{pmatrix} = 2 \sum_i f_i \begin{pmatrix} 1 \\ p_i \\ p_i^2 \\ \vdots \end{pmatrix}$$

Radial Basis Functions

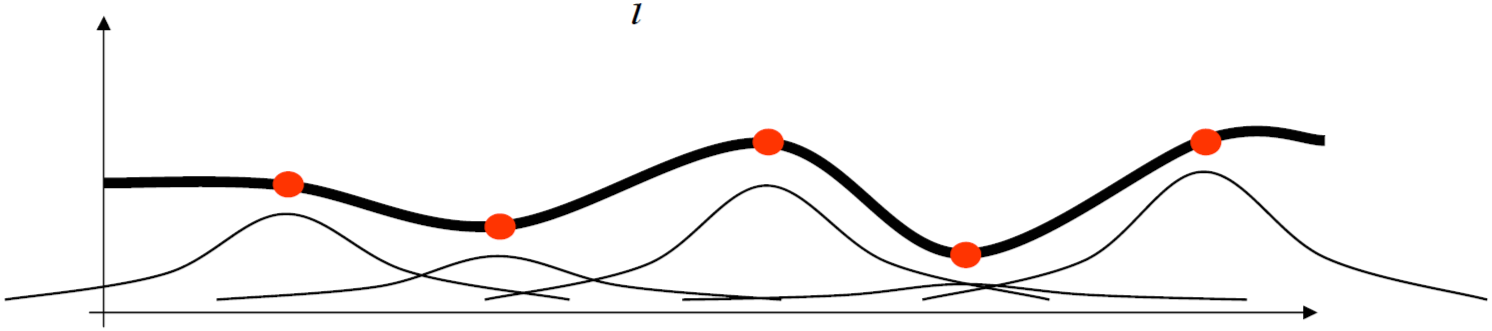
- Solve $$f_j = \sum_i w_i r(\|\mathbf{p}_i - \mathbf{p}_j\|)$$ to compute weights w_i
- Linear system of equations

$$\begin{pmatrix} \mathrm{r}(0) & \mathrm{r}(\|\mathbf{p}_0 - \mathbf{p}_1\|) & \mathrm{r}(\|\mathbf{p}_0 - \mathbf{p}_2\|) & \cdots \\ \mathrm{r}(\|\mathbf{p}_1 - \mathbf{p}_0\|) & \mathrm{r}(0) & \mathrm{r}(\|\mathbf{p}_1 - \mathbf{p}_2\|) & \\ \mathrm{r}(\|\mathbf{p}_2 - \mathbf{p}_0\|) & \mathrm{r}(\|\mathbf{p}_2 - \mathbf{p}_1\|) & \mathrm{r}(0) & \\ \vdots & & & \ddots \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \end{pmatrix}$$

Radial Basis Functions

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

$$\mathrm{f}(\mathbf{x}) = \sum_i w_i \, \mathrm{r}\left(\left\|\mathbf{p}_i - \mathbf{x}\right\|\right)$$



Radial Basis Functions

- Solvability depends on radial function
- Several choices assure solvability
  - $\mathrm{r}(d) = d^2 \log d$ (thin plate spline)
  - $\mathrm{r}(d) = e^{-d^2/h^2}$ (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

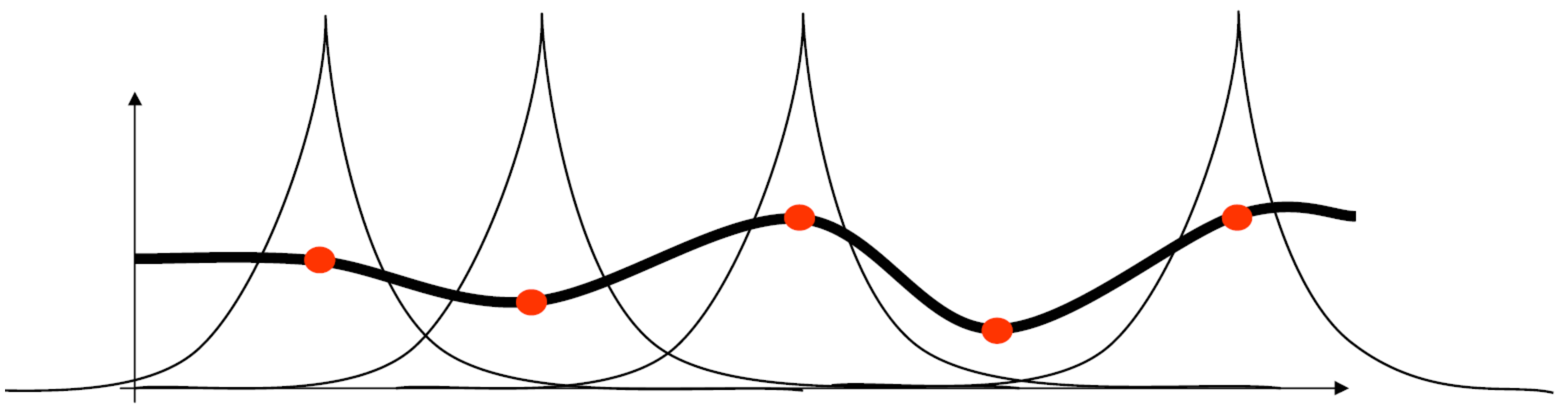
- Notation, Terms
  - Regular/Irregular, Approximation/Interpolation, Global/Local
- **Standard interpolation/approximation techniques**
  - Global: Triangulation, Voronoi-Interpolation, Least Squares (LS), Radial Basis Functions (RBF)
  - **Local: Shepard/Partition of Unity Methods, Moving LS**
- Problems
  - Sharp edges, feature size/noise
- Functional -> Manifold

Shepard Interpolation

- Approach: $\mathrm{f}(\mathbf{x}) = \sum_i \phi_i(\mathbf{x}) f_i$

  with basis functions $$\phi_i(\mathbf{x}) = \frac{\|\mathbf{x}-\mathbf{x}_i\|^{-p}}{\sum_j \|\mathbf{x}-\mathbf{x}_j\|^{-p}}$$

- define $\mathrm{f}(\mathbf{p}_i) = f_i = \lim_{\mathbf{x}\to\mathbf{p}_i} \mathrm{f}(\mathbf{x})$



Shepard Interpolation

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

Shepard Interpolation

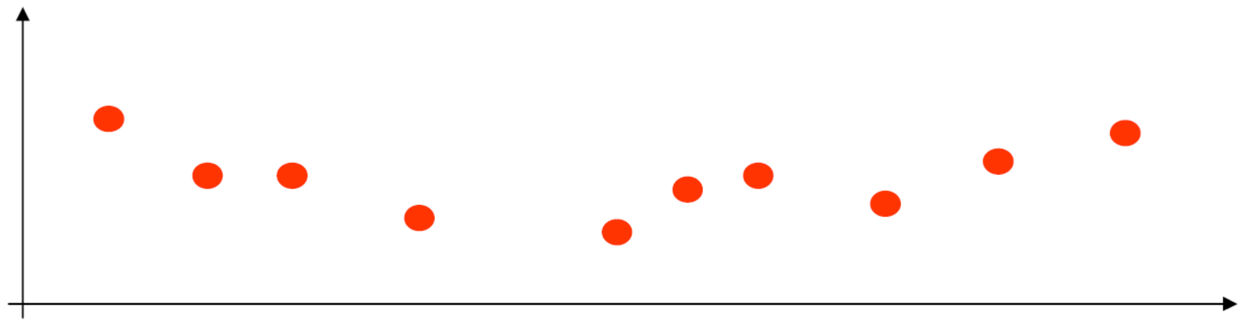
Localization:

- Set $\mathrm{f}(\mathbf{x}) = \sum_i \mu_i(\mathbf{x}) \phi_i(\mathbf{x}) f_i$
- with $\mu_i(\mathbf{x}) = \begin{cases} \left(1 - \|\mathbf{x} - \mathbf{p}_i\| / R_i\right)^\nu & \text{if } \|\mathbf{x} - \mathbf{p}_i\| < R_i \\ 0 & \text{else} \end{cases}$

  for reasonable R_i and $\nu > 1$

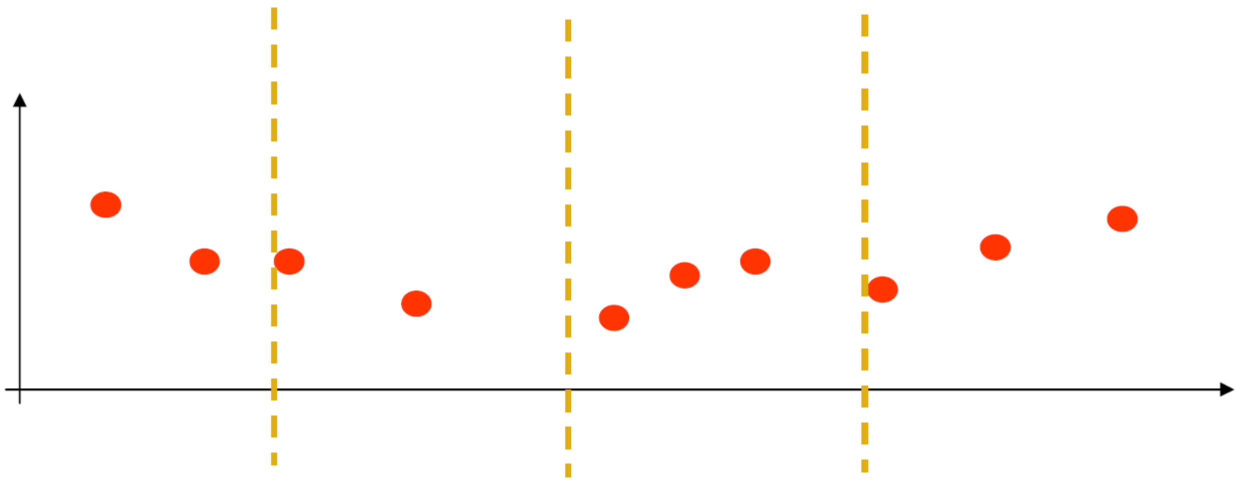
  ➔ no constant precision because of possible holes in the data

Partition of Unity Methods



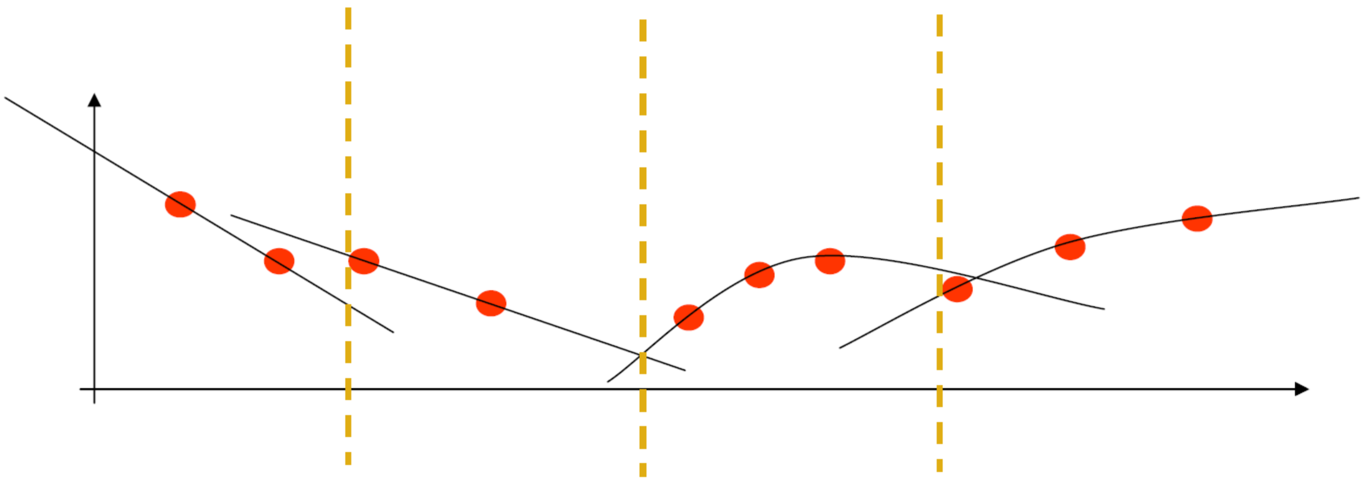
Partition of Unity Methods

- Subdivide domain into cells



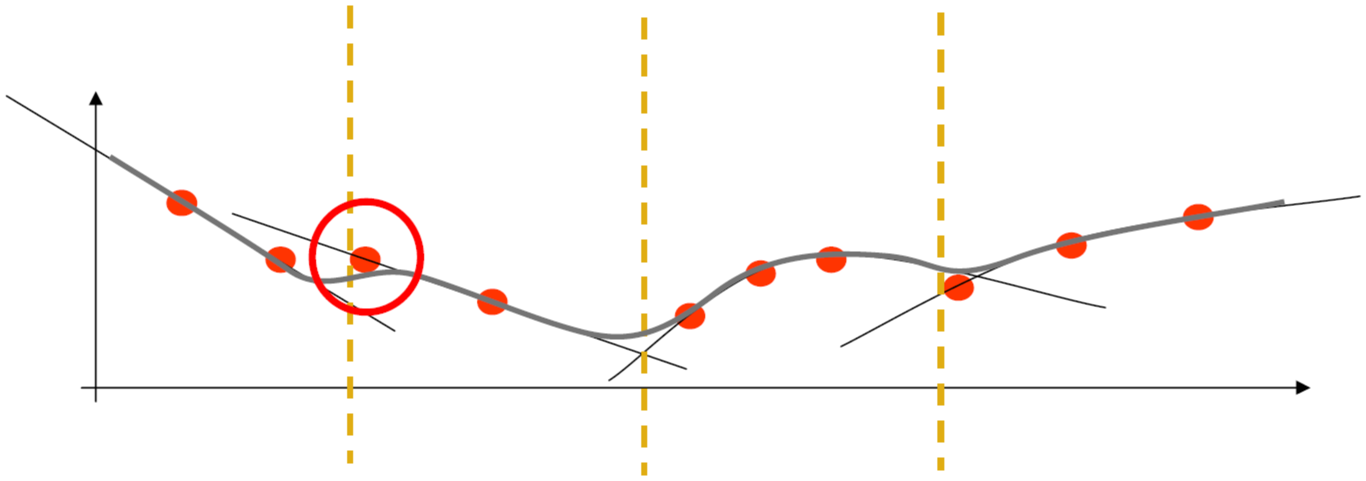
Partition of Unity Methods

- Compute local interpolation per cell



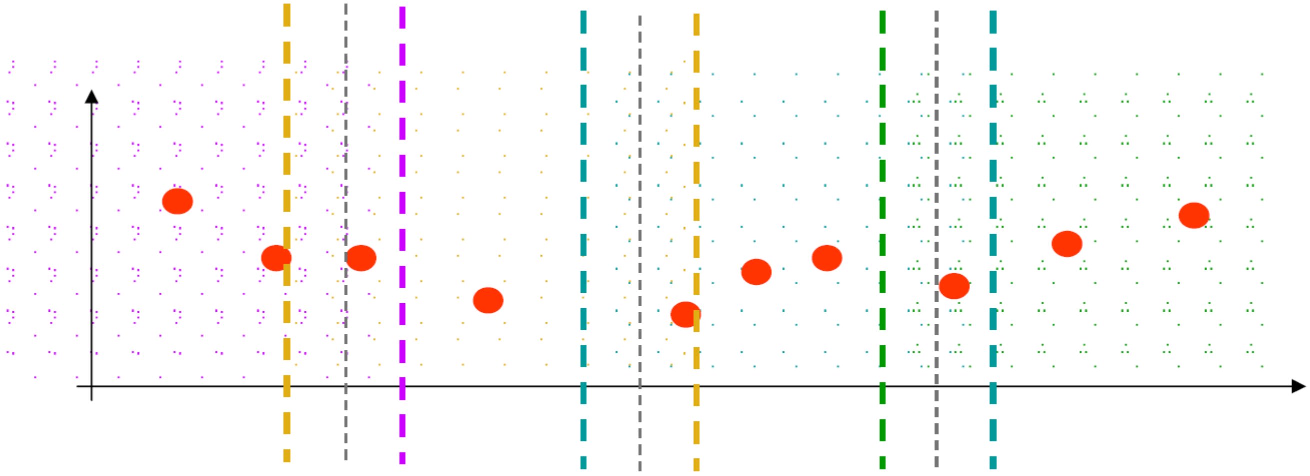
Partition of Unity Methods

- Blend local interpolations?



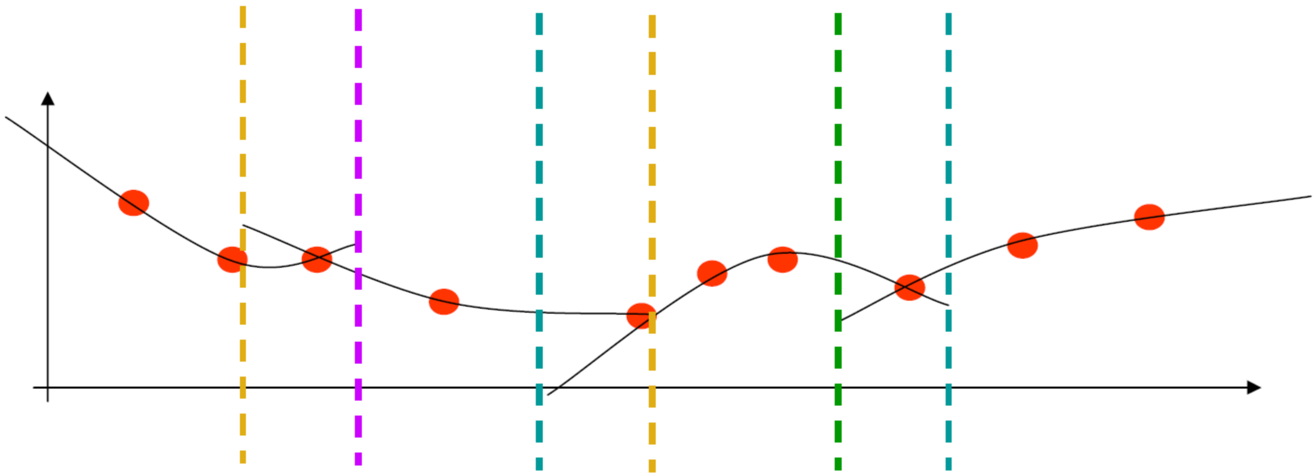
Partition of Unity Methods

- Subdivide domain into *overlapping* cells



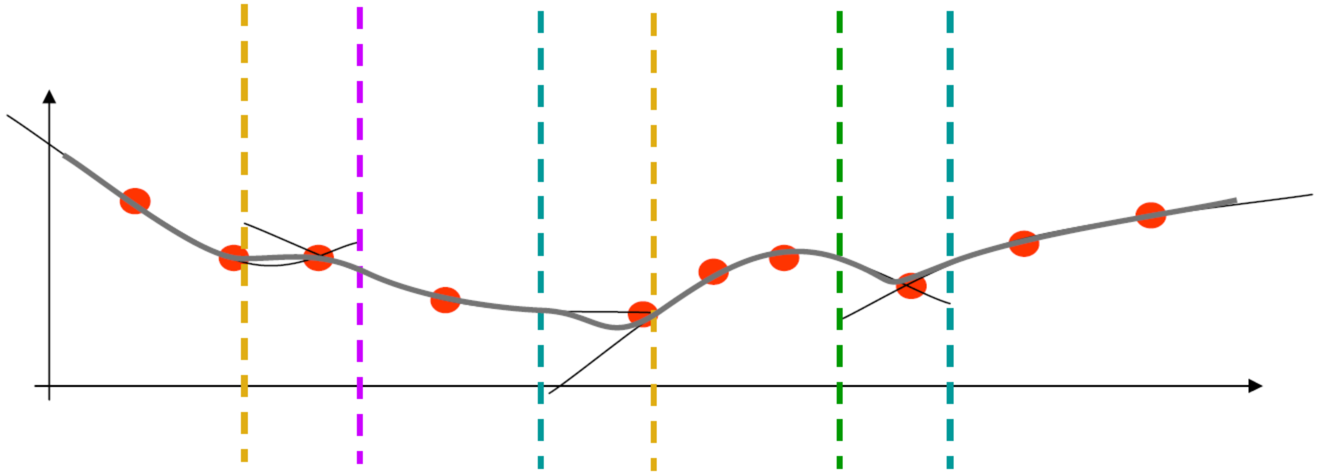
Partition of Unity Methods

- Compute local interpolations



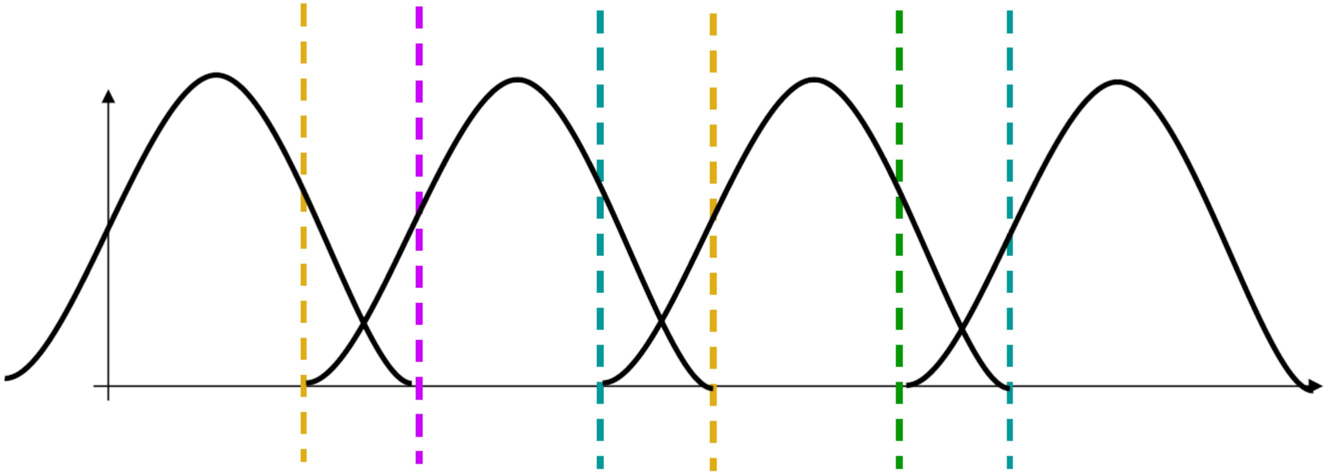
Partition of Unity Methods

- Blend local interpolations



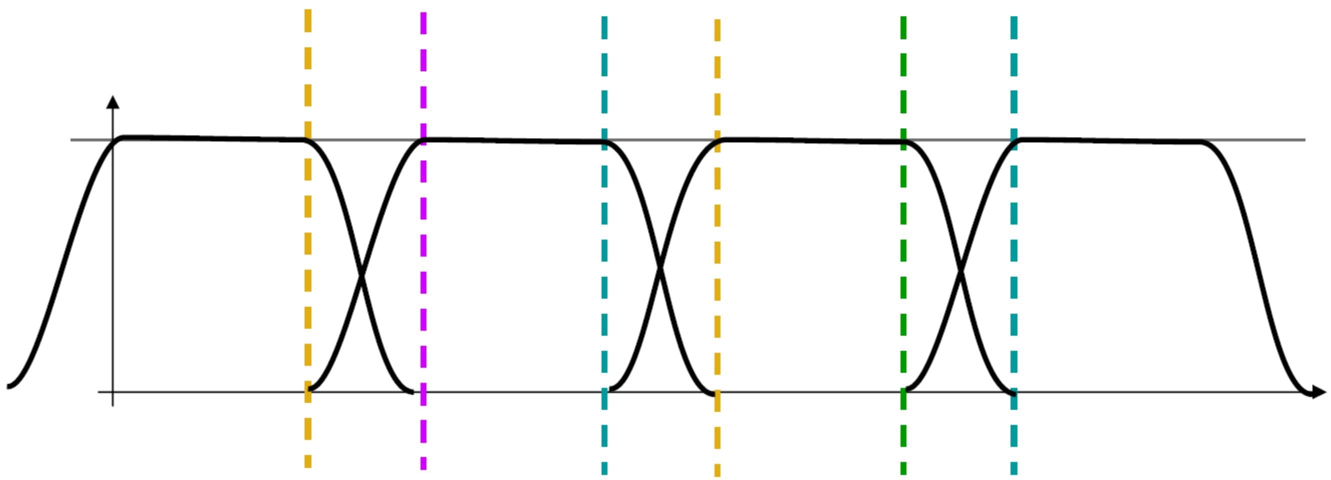
Partition of Unity Methods

- Weights should
  - have the (local) support of the cell



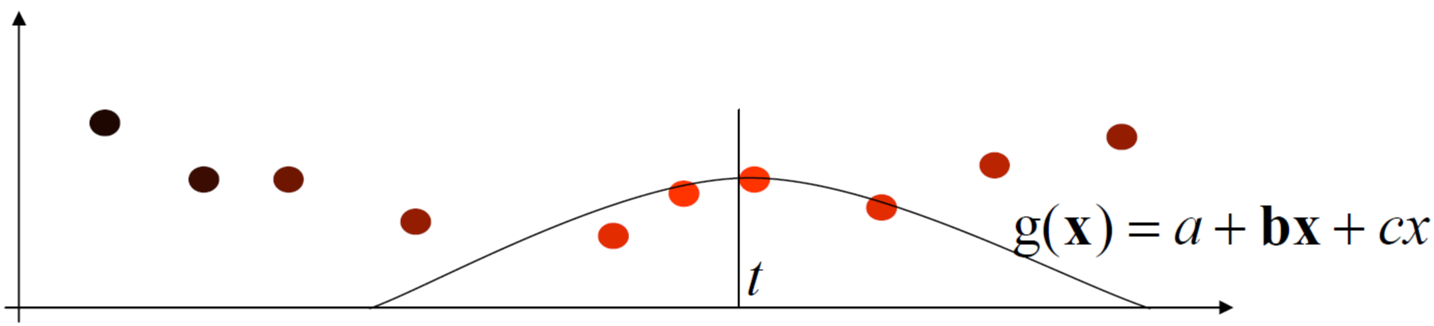
Partition of Unity Methods

- 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 $\mathbf{x}$
- Weight data points based on distance to $\mathbf{x}$



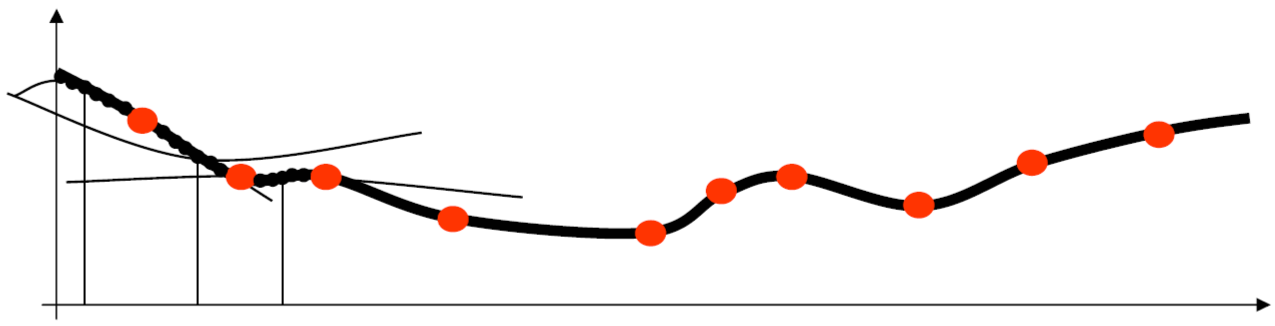
$$\min \sum_i \left(f_i - g(\mathbf{p}_i)\right)^2 \theta\left(\left\|\mathbf{x} - \mathbf{p}_i\right\|\right)$$

Moving Least Squares

- The set

$$f(\mathbf{x}) = g_{\mathbf{x}}(\mathbf{x}), g_{\mathbf{x}} : \min_{g} \sum_{i} \left(f_i - g(\mathbf{p}_i)\right)^2 \theta(\|\mathbf{x} - \mathbf{p}_i\|)$$

is a smooth curve, iff θ is smooth



Moving Least Squares

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

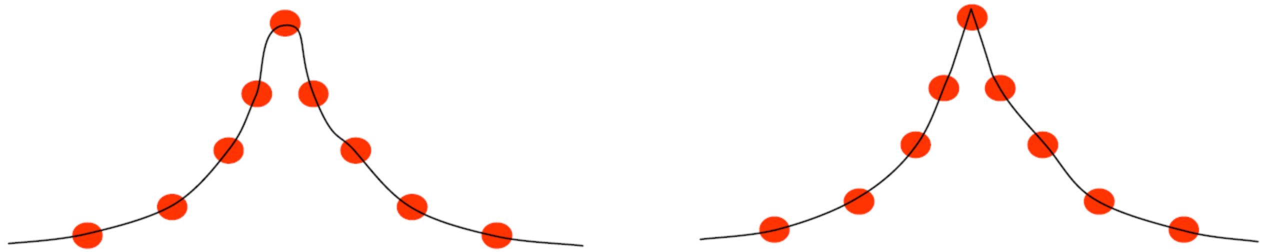
- Note: $\theta_i = \theta(\|\mathbf{x} - \mathbf{p}_i\|)$ is fixed
- For each $\mathbf{x}$
  - Standard weighted LS problem
  - Linear iff corresponding LS is linear

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

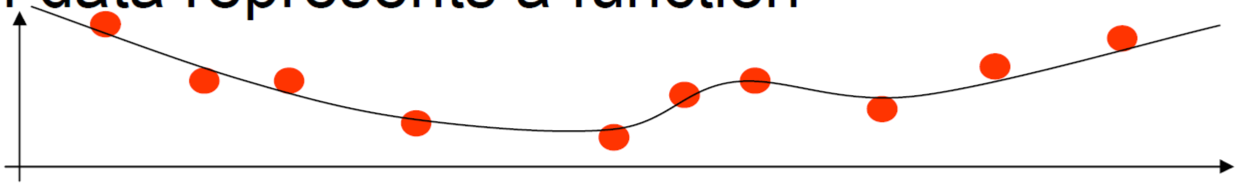
- Sharp corners/edges



- 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

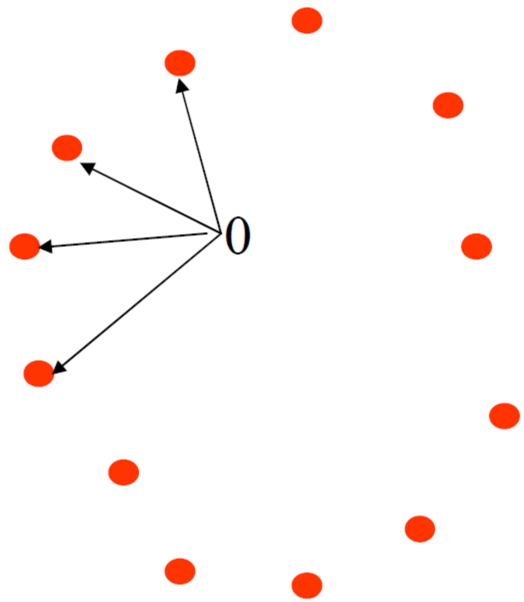
- Introduction & Basics
- **Fitting Implicit Surfaces**
- Surfaces from Local Frames

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: $f(\mathbf{x}) < 0$
  - On the manifold: $f(\mathbf{x}) = 0$
  - Outside: $f(\mathbf{x}) > 0$

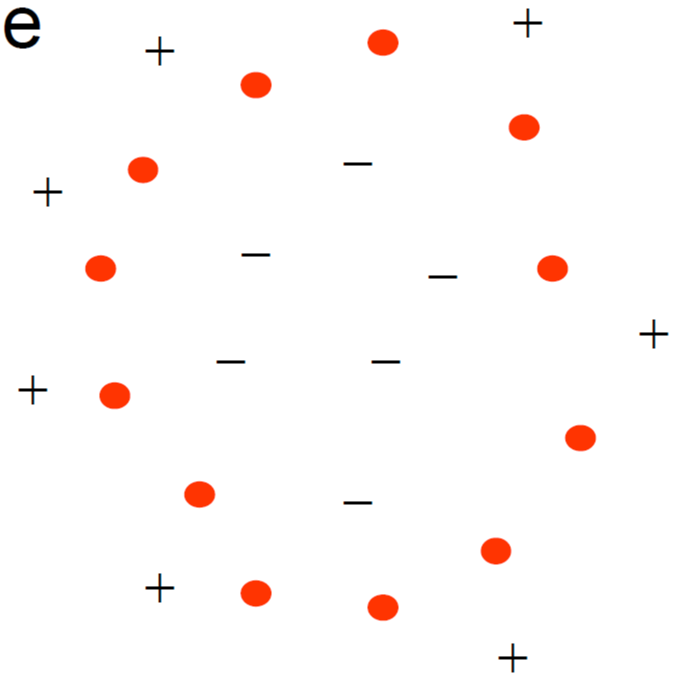
Implicits from point samples

- Function should be zero in data points
  - $\mathrm{f}(\mathbf{p}_i) = 0$
- Use standard approximation techniques to find f
- Trivial solution: $\mathrm{f} = 0$
- Additional constraints are needed



Implicits from point samples

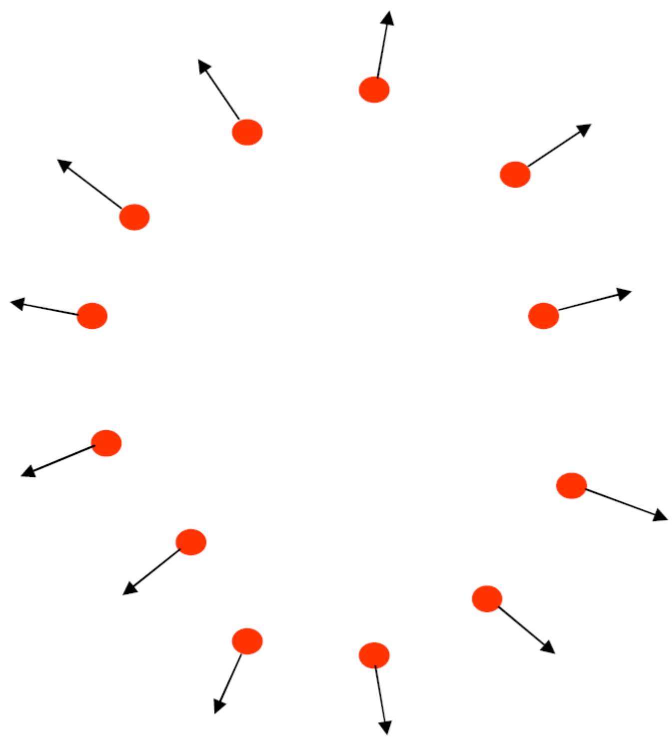
- Constraints define inside and outside
- Simple approach (Turk, O'Brien)
  - Sprinkle additional information manually
  - Make additional information soft constraints



Implicits from point samples

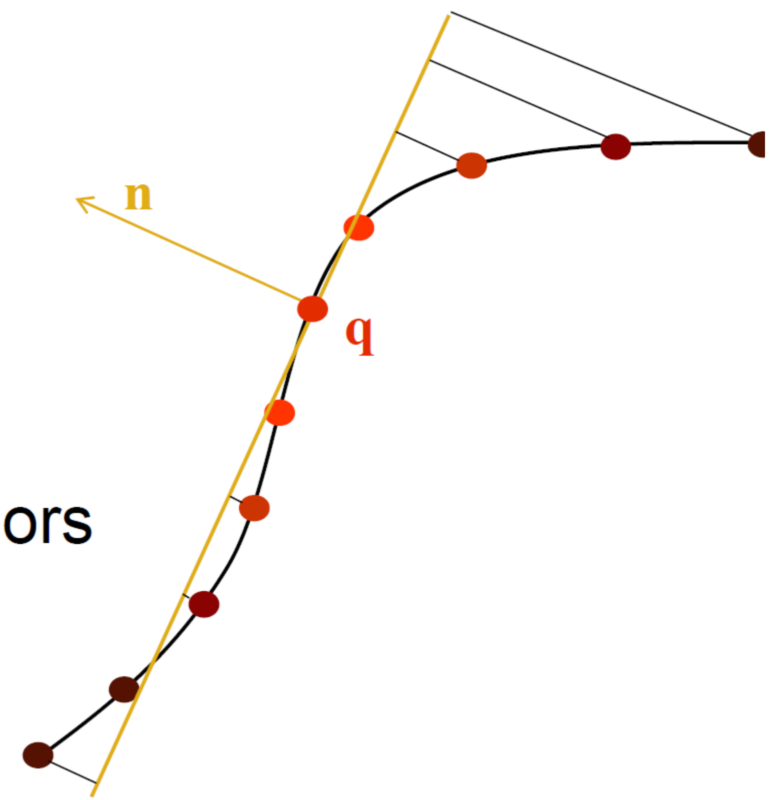
- Use normal information

- 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

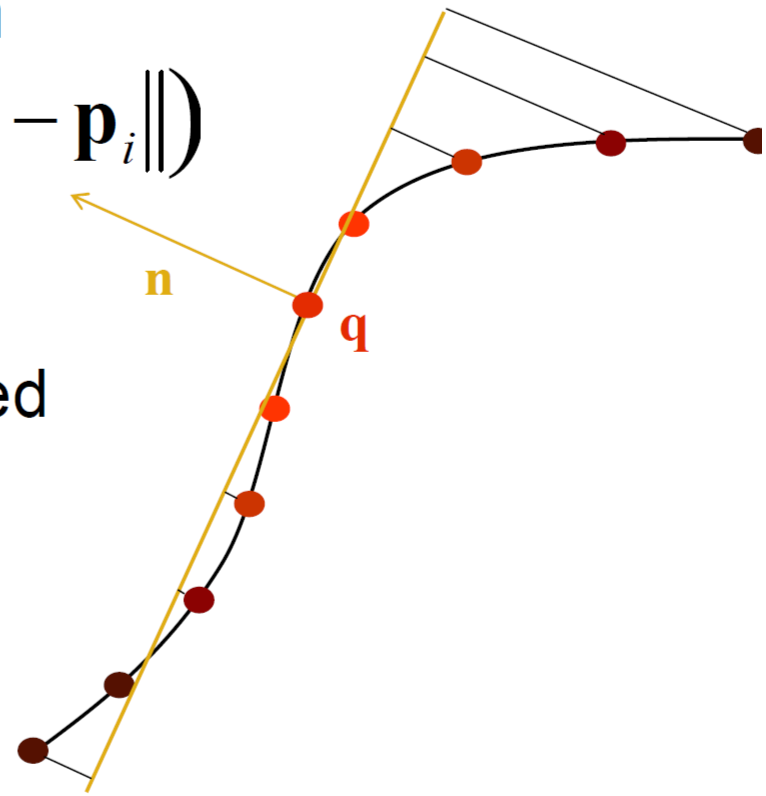


Estimating normals

- General fitting problem

$$\min_{\|\mathbf{n}\|=1} \sum_i \langle \mathbf{q}-\mathbf{p}_i, \mathbf{n} \rangle^2 \theta(\|\mathbf{q}-\mathbf{p}_i\|)$$

  - Problem is non-linear because $\mathbf{n}$ is constrained to unit sphere



Estimating normals

- The constrained minimization problem

$$\min_{\|\mathbf{n}\|=1} \sum_i \langle \mathbf{q} - \mathbf{p}_i, \mathbf{n} \rangle^2 \theta_i$$

is solved by the eigenvector corresponding to the smallest eigenvalue of the following co-variance matrix

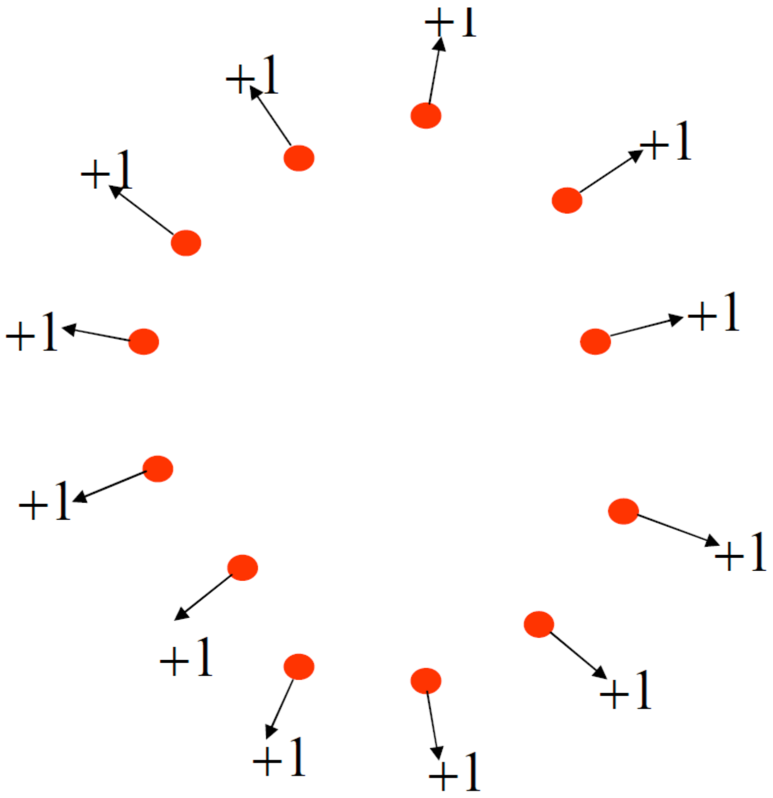
$$\sum_i (\mathbf{q} - \mathbf{p}_i) \cdot (\mathbf{q} - \mathbf{p}_i)^{\mathrm{T}} \theta_i$$

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

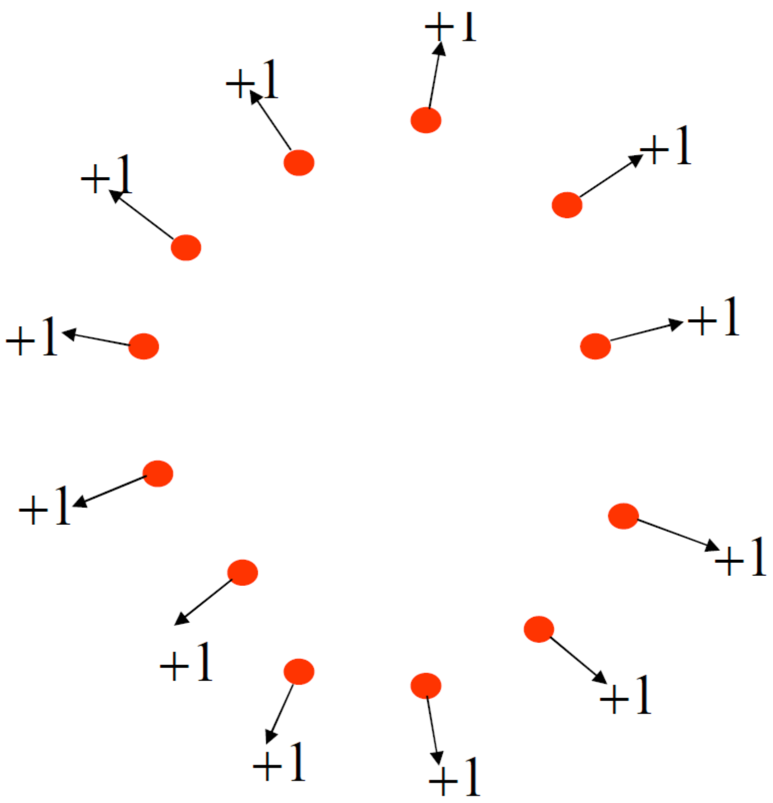
$$f(\mathbf{p}_i + \mathbf{n}_i) = 1$$



Implicits from point samples

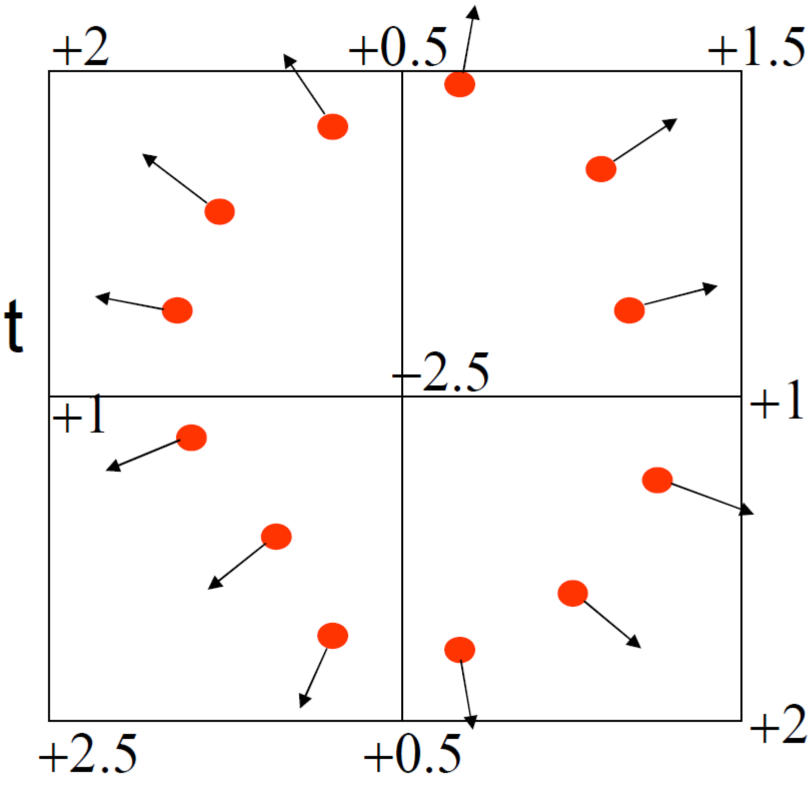
- Compute non-zero anchors in the distance field
- Use normal information directly as constraints

$$f(\mathbf{p}_i + \mathbf{n}_i) = 1$$



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 $\mathbf{p}_i, \mathbf{n}_i$ and constraints $\mathrm{f}(\mathbf{p}_i) = 0, \mathrm{f}(\mathbf{c}_i) = d_i$

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

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

leads to a system of linear equations

Computing Implicits

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

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

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

leads to a system of linear equations

Computing Implicits

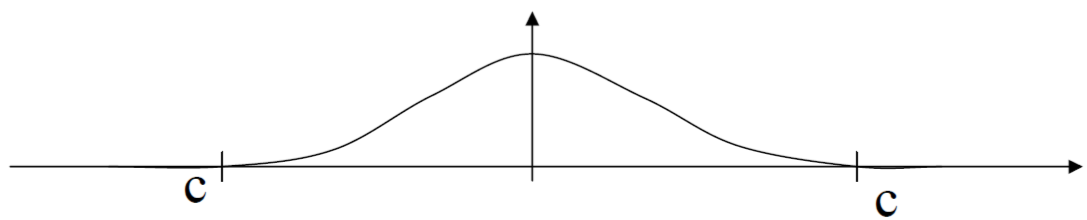
- Practical problems: $N > 10000$
- Matrix solution becomes difficult
- Two solutions
  - Sparse matrices allow iterative solution
  - Smaller number of RBFs

Computing Implicits

- Sparse matrices

$$\begin{pmatrix} \theta(0) & \theta(\|\mathbf{p}_0-\mathbf{p}_1\|) & \theta(\|\mathbf{p}_0-\mathbf{p}_2\|) & \cdots \\ \theta(\|\mathbf{p}_1-\mathbf{p}_0\|) & \theta(0) & \theta(\|\mathbf{p}_1-\mathbf{p}_2\|) & \\ \theta(\|\mathbf{p}-\mathbf{p}_0\|) & \theta(\|\mathbf{p}_2-\mathbf{p}_1\|) & \theta(0) & \\ \vdots & & & \ddots \end{pmatrix}$$

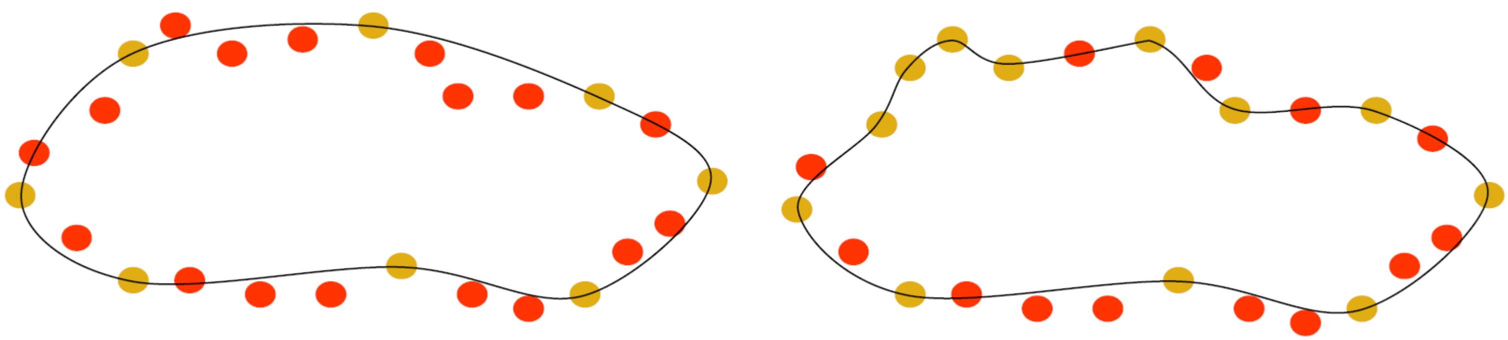
  - Needed: $d > c \rightarrow r(d) = 0, r'(c) = 0$



  - Compactly supported RBFs

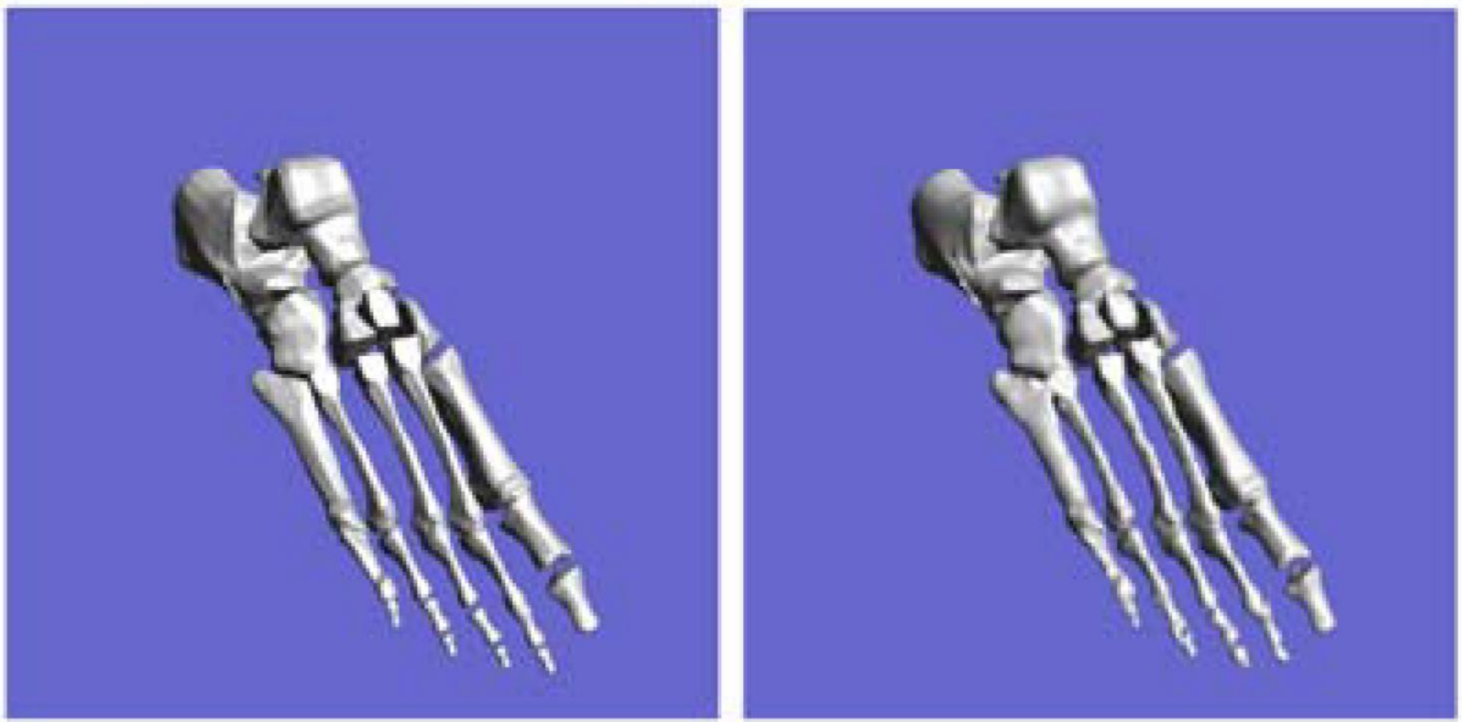
Computing Implicits

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



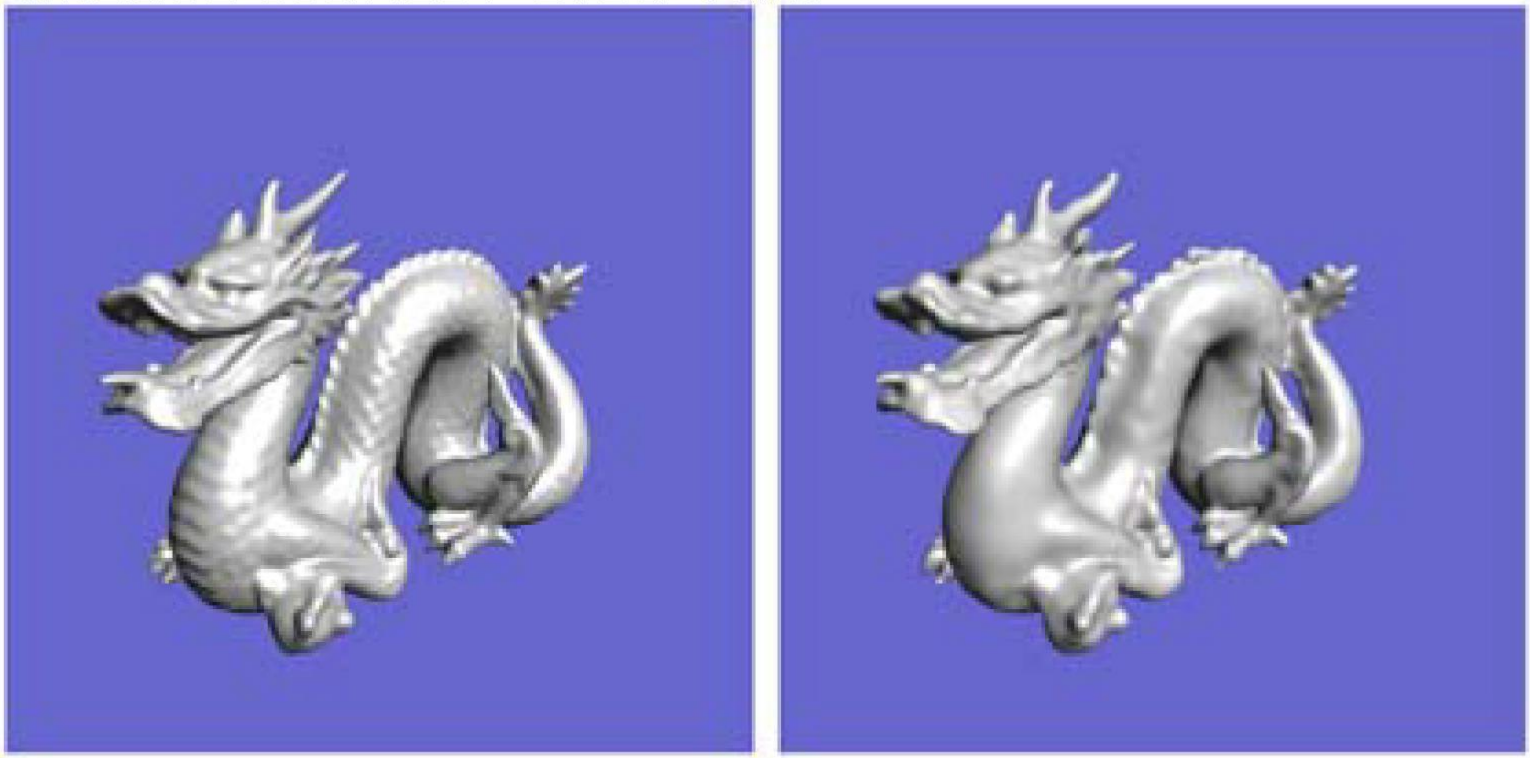
RBF Implicits - Results

- Images courtesy Greg Turk



RBF Implicits - Results

- Images courtesy Greg Turk

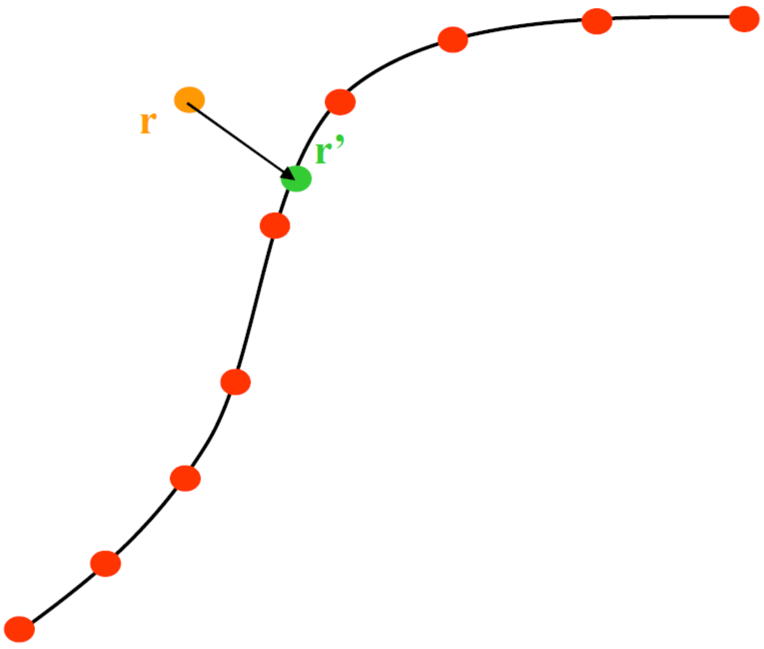


Overview

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

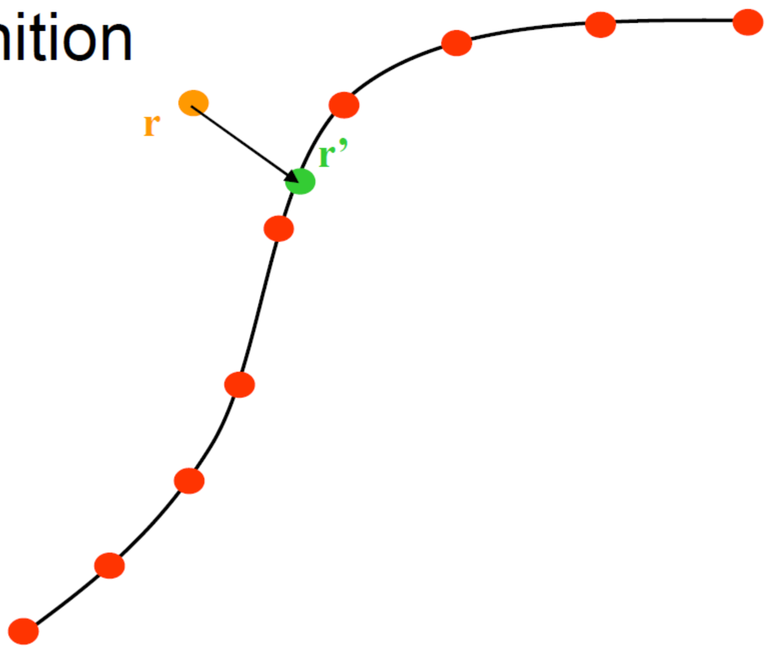
Projection

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



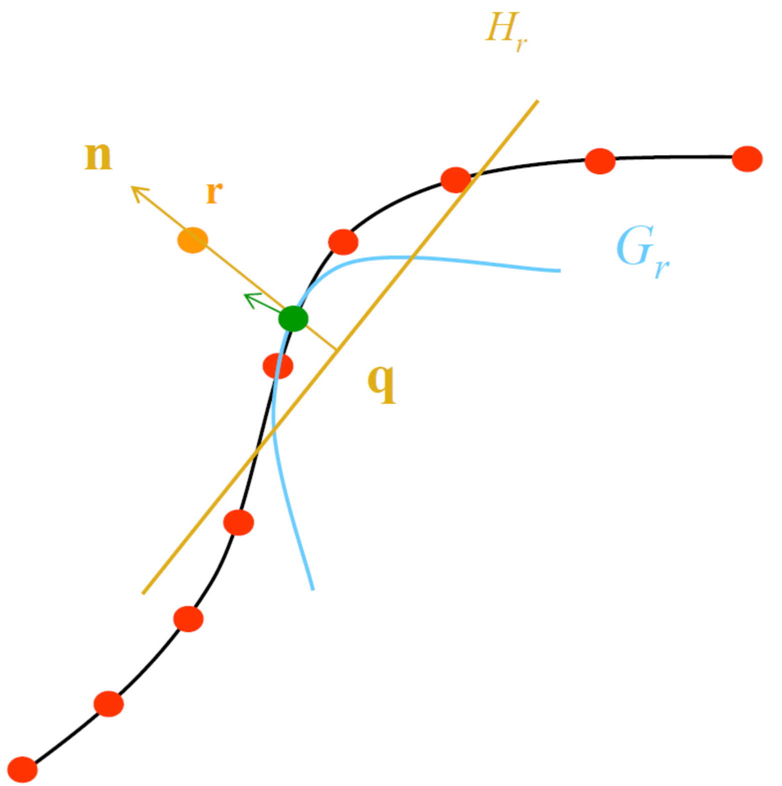
Projection

- Projection procedure (Levin)
  - Local polyonmial approximation
    - Inspired by differential geometry
  - “Implicit” surface definition
  - Infinitely smooth &
  - Manifold surface



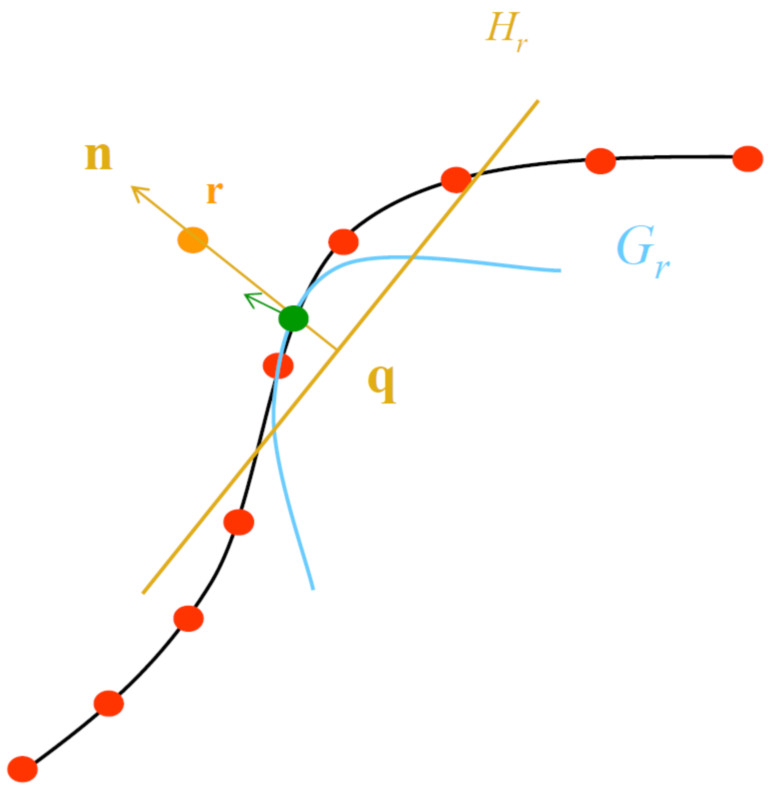
Surface Definition

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



Surface Definition

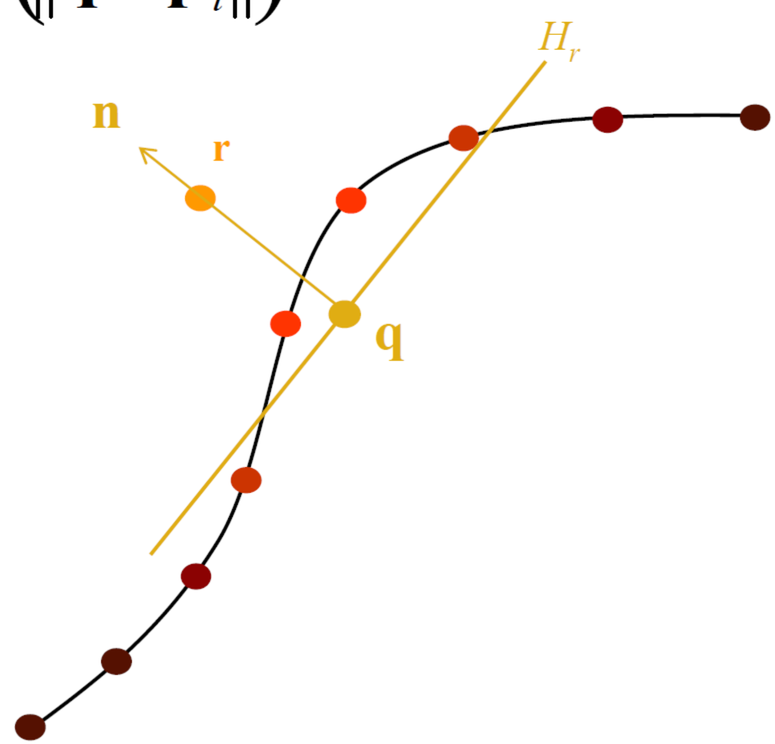
- Constructive definition
  - Input point $\mathbf{r}$
  - Compute a local reference plane $H_{\mathbf{r}}=\langle \mathbf{q},\mathbf{n} \rangle$
  - Compute a local polynomial over the plane $G_{\mathbf{r}}$
  - Project point $\mathbf{r}'=G_{\mathbf{r}}(0)$
  - Estimate normal



Local Reference Plane

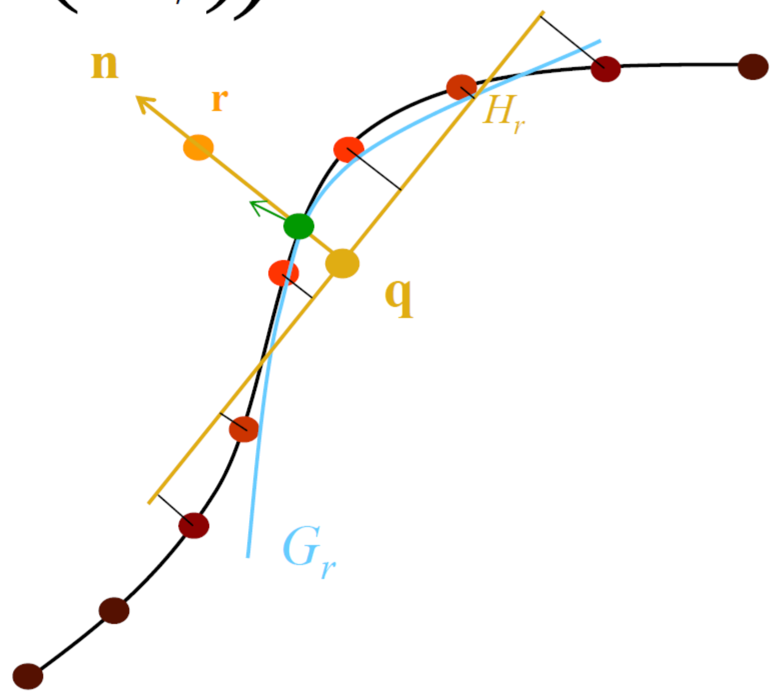
- Find plane $H_r = \langle \mathbf{q}, \mathbf{n} \rangle + D$
  - $\min_{\mathbf{q}, \|\mathbf{n}\|=1} \sum_i \langle \mathbf{q} - \mathbf{p}_i, \mathbf{n} \rangle^2 \theta(\|\mathbf{q} - \mathbf{p}_i\|)$
  - $\theta(\mathrm{d}) = e^{d^2/h^2}$
    - h is feature size/ point spacing
  - H_r is independent of r's distance
  - Manifold property

Weight function based on distance to q, not r



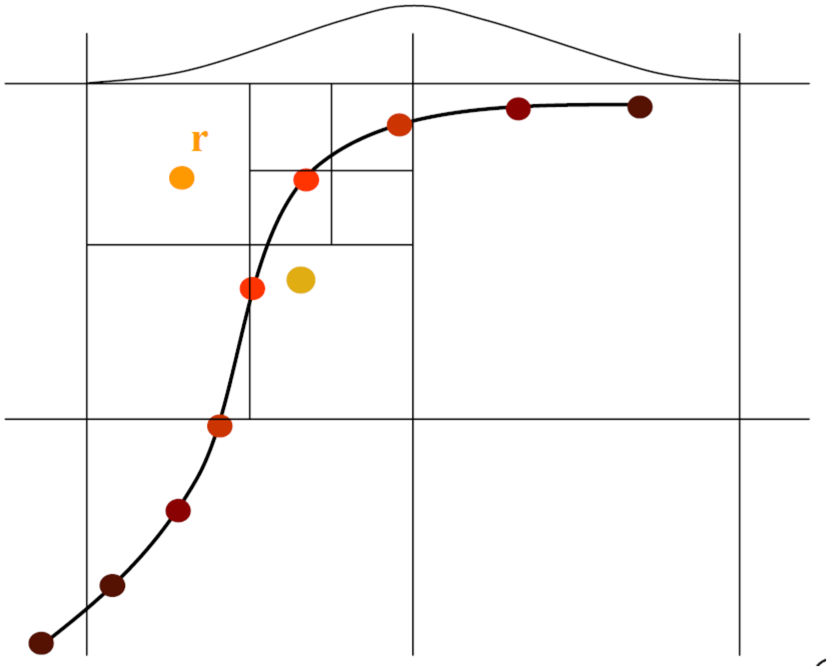
Projecting the point

- MLS polyonomial over H_r
  - $\min_{G \in \Pi_d} \sum_i \left(\langle \mathbf{q} - \mathbf{p}_i, \mathbf{n} \rangle - G\left(\mathbf{p}_i |_{H_r} \right) \right)^2 \theta\left(\|\mathbf{q} - \mathbf{p}_i\| \right)$
  - LS problem
  - $\mathbf{r}' = G_r(0)$
  - Estimate normal



Spatial data structure

- Regular grid based on support of θ
  - 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