CS395T Lecture 18: 3D Representations

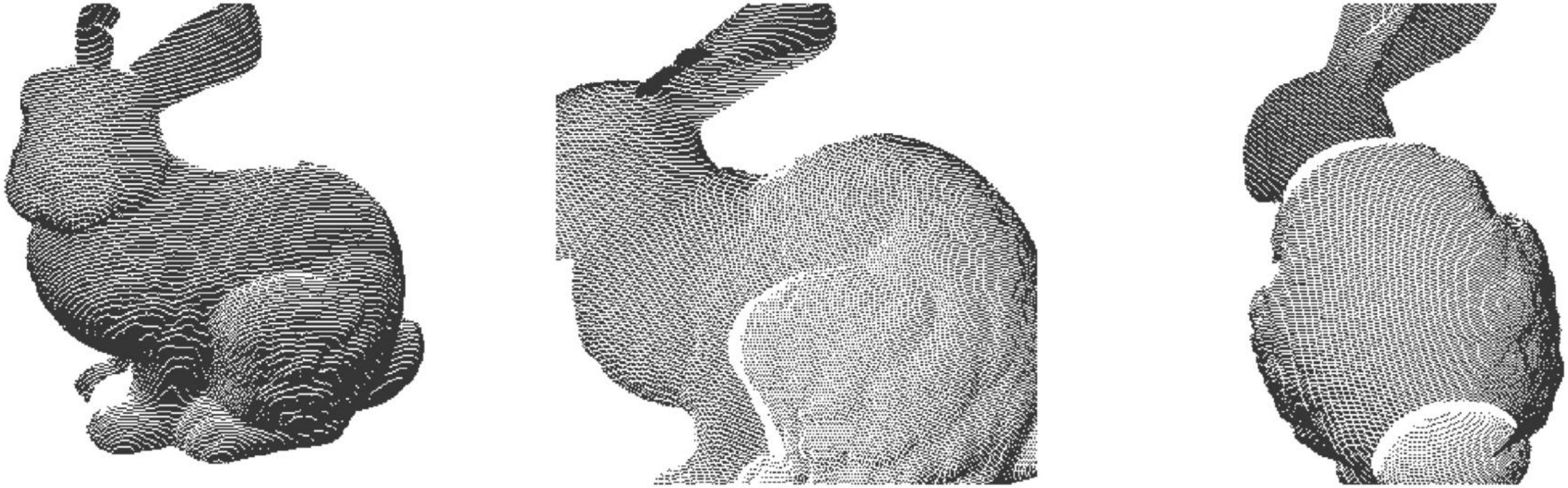
Qixing Huang

Oct. 31th 2018

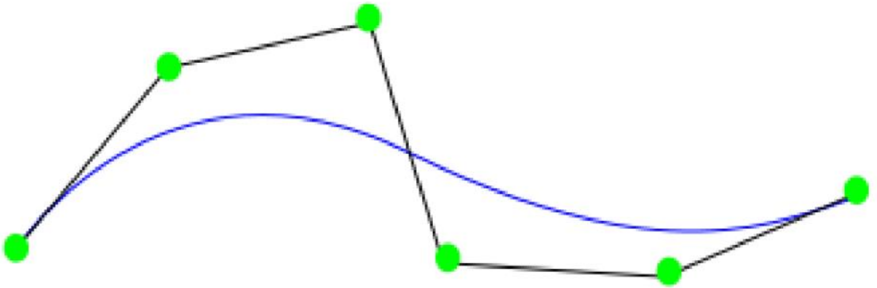
3D Representations

- Point clouds
- Parametric surfaces
- Implicit surfaces
- Triangular meshes
- Part-based models

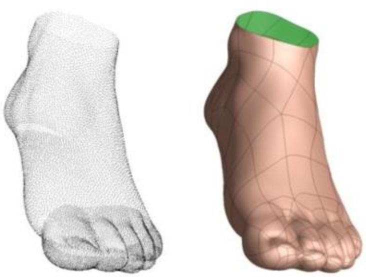
Point Cloud



Parametric surfaces



Bspline curve



Eck and Hoppe' 96

Implicit Surfaces

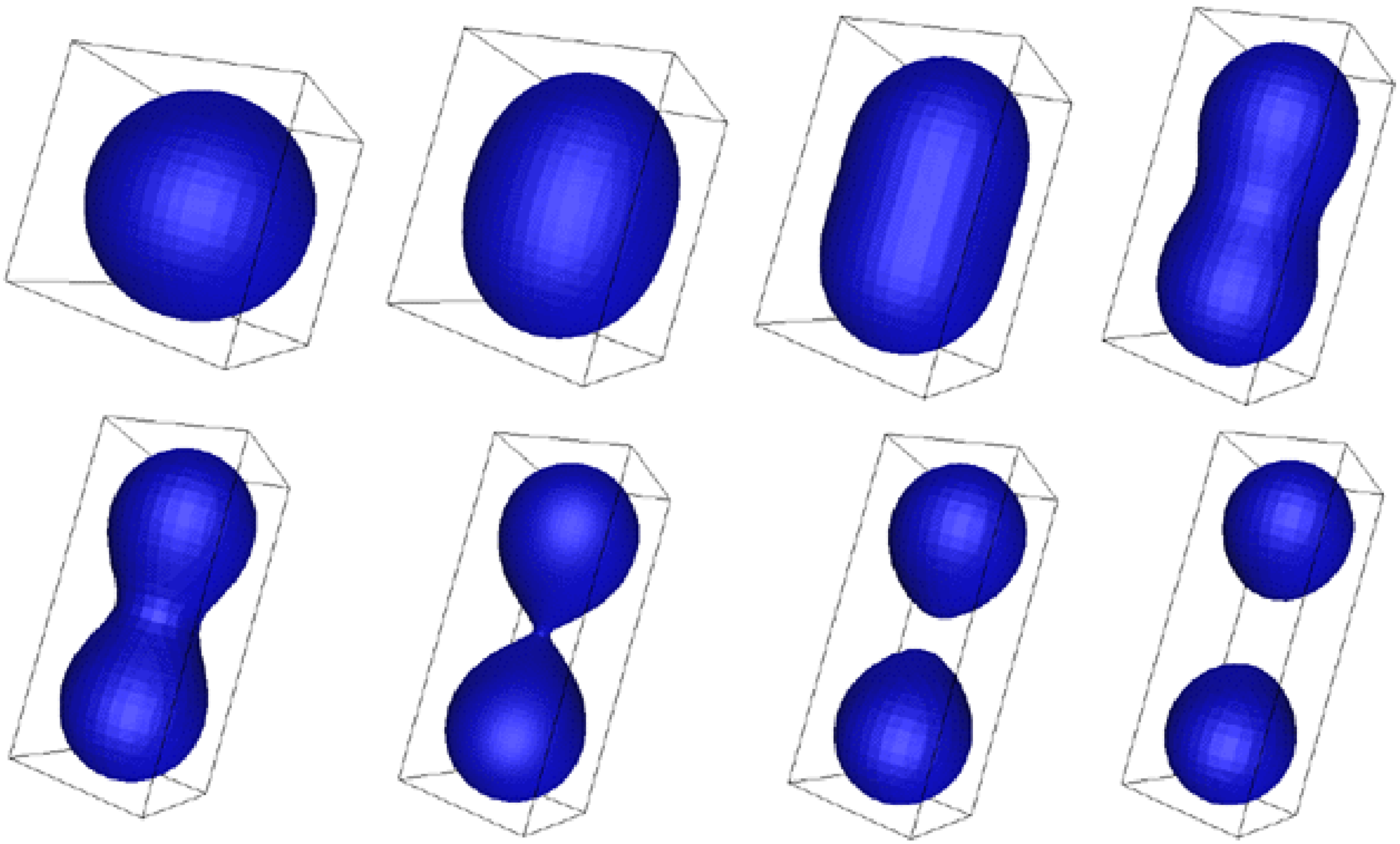
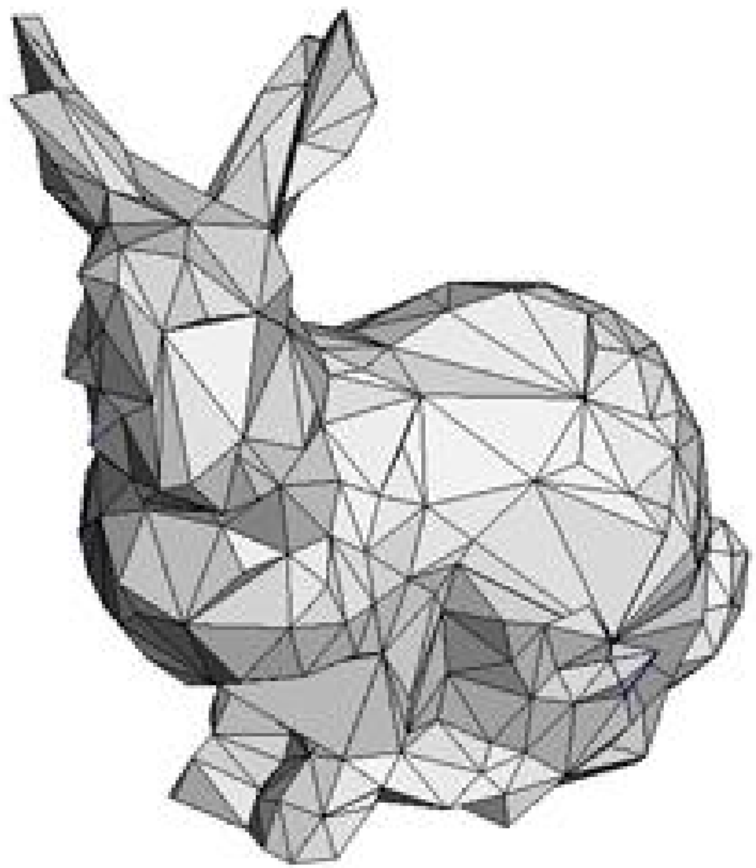


Image from http://paulbourke.net/geometry/implicitsurf/implicitsurf4.gif

Triangular Mesh



Scene Graph

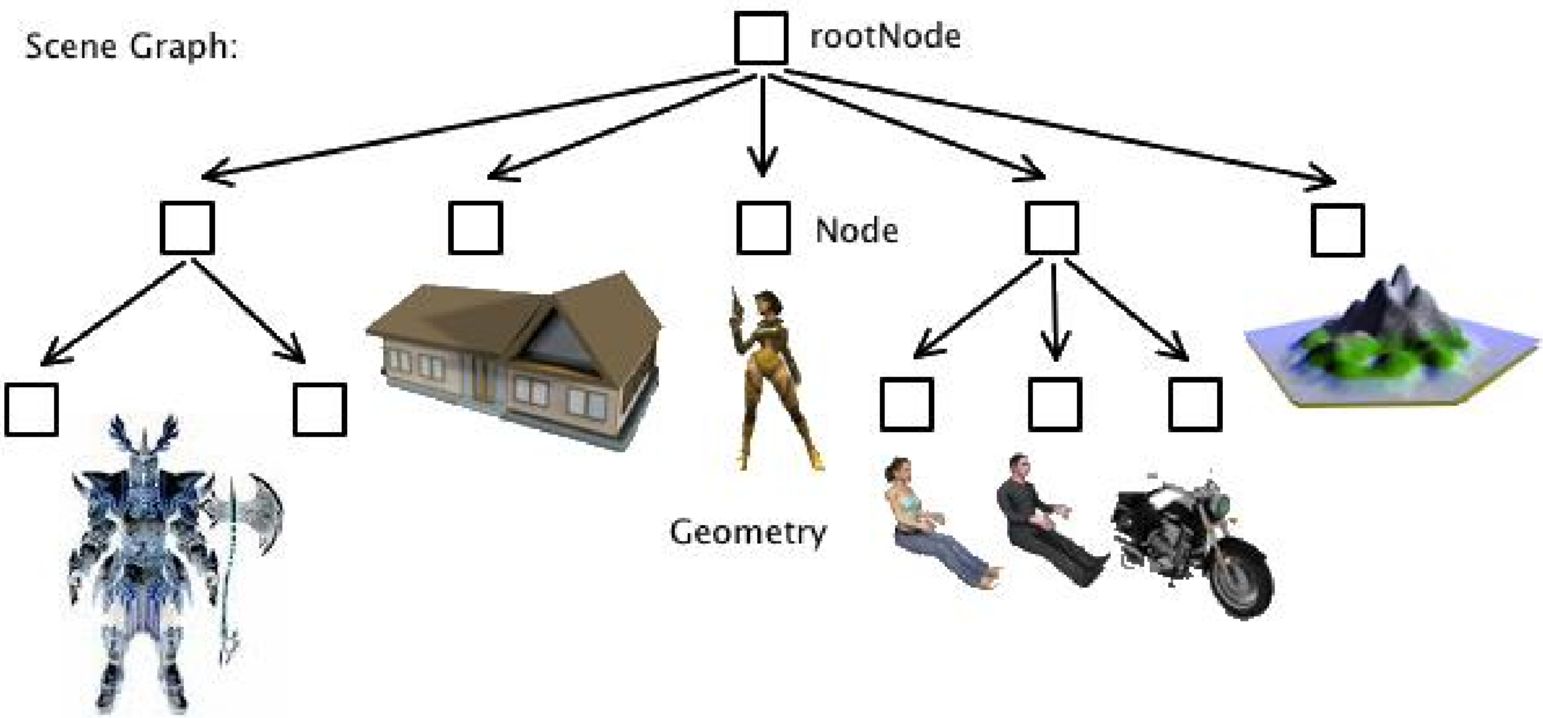


Image from https://gamedev.stackexchange.com/tags/scene-graph/info

What to Learn?

- Pros and cons of each representation
- Conversions between different representations

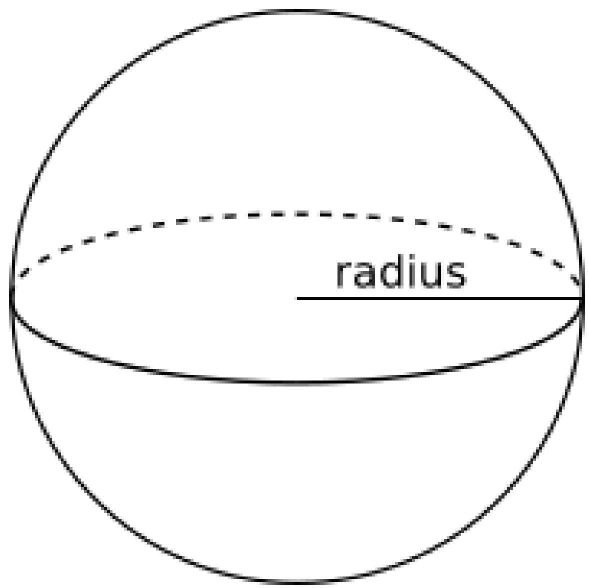
This Lecture

- Implicit surface
- Point cloud -> Implicit Surface
- Implicit surface -> triangular mesh

Implicit Surfaces

What is implicit surface?

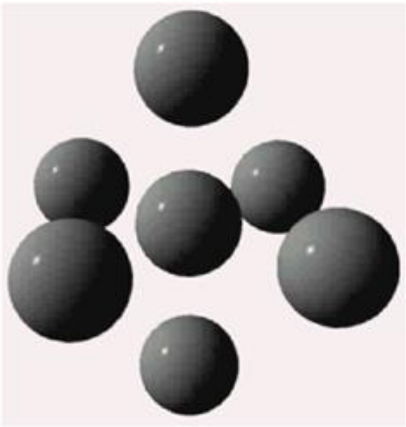
- A sphere $x^2 + y^2 + z^2 = \text{radius}^2$ is an implicit surface



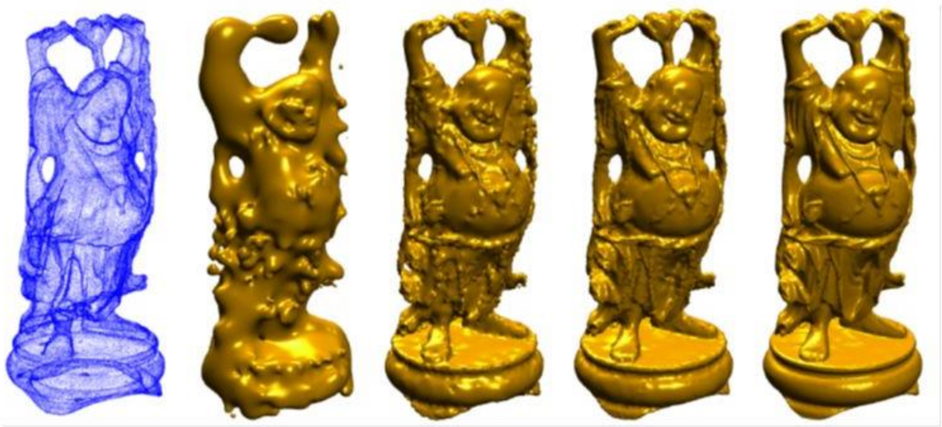
What is implicit surface?

- Implicit surfaces are two-dimensional, geometric shapes that exist in three dimensional space
  - Defined in R^3
  - 2D Manifold if no singular points
  - A surface embedded in R^3

Examples of implicit surfaces



Metaball



Radial Basis Function
[Carr et al. 01]

Definition of implicit surface

- Definition

$$\{p=(x,y,z): f(p)=0,\ p\in\mathbf{R}^3\}$$

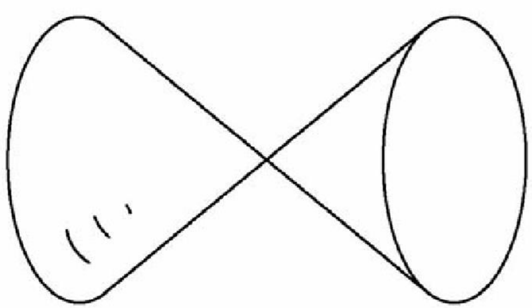
- When *f* is algebraic function, i.e., polynomial function
  - Note that *f* and *c*f* specify the same curve
  - Algebraic distance: the value of *f(p)* is the approximation of distance from *p* to the algebraic surface *f=0*

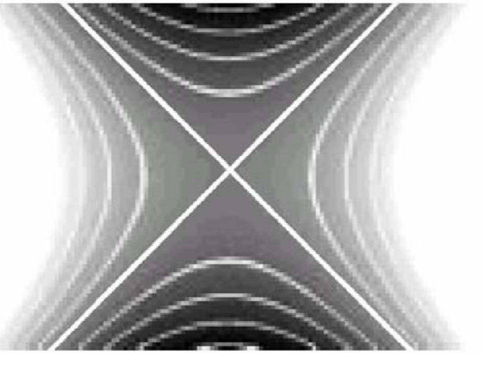
Definition of implicit surface

- Regular point *p* on the surface

$$\nabla f(p) = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}) \neq 0$$

- Consider cone $z^2=x^2+y^2$
  - (0,0,0) is not a regular point





Implicit function theorem

Let f: $\mathbf{R}^{n+m} \to \mathbf{R}^m$ be a continuously differentiable function, and let $\mathbf{R}^{n+m}$ have coordinates $(\mathbf{x}, \mathbf{y})$. Fix a point $(\mathbf{a}, \mathbf{b}) = (a_1, ..., a_n, b_1, ..., b_m)$ with $f(\mathbf{a}, \mathbf{b}) = \mathbf{0}$, where $\mathbf{0} \in \mathbf{R}^m$ is the zero vector. If the Jacobian matrix $J_{f,\mathbf{y}}(\mathbf{a}, \mathbf{b}) = [(\partial f_i / \partial y_j)(\mathbf{a}, \mathbf{b})]$ is invertible, then there exists an open set U of $\mathbf{R}^n$ containing $\mathbf{a}$, and such that there exists a unique continuously differentiable function g: $U \to \mathbf{R}^m$ such that

$$g(\mathbf{a}) = \mathbf{b}$$

and

$$f(\mathbf{x}, g(\mathbf{x})) = \mathbf{0} \text{ for all } \mathbf{x} \in U.$$

Moreover, the partial derivatives of g in U are given by

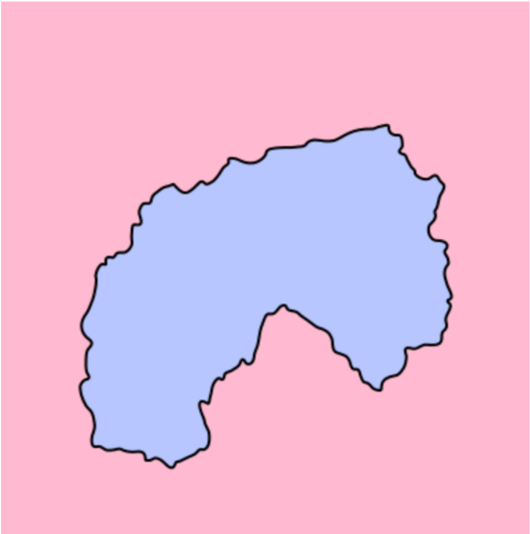
$$\frac{\partial g}{\partial x_j}(\mathbf{x}) = -\sum_i \left(J_{f,\mathbf{y}}(\mathbf{x}, g(\mathbf{x}))^{-1}\right)_{ji} \frac{\partial f}{\partial x_i}(\mathbf{x}, g(\mathbf{x})).$$

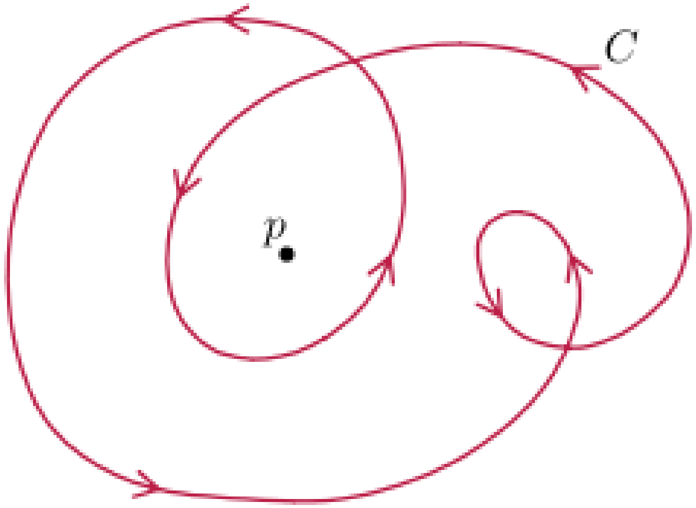
No singular points then an implicit surface is a manifold

From https://en.wikipedia.org/wiki/Implicit_function_theorem

Jordan-Brouwer Separation Theorem

- Any compact, connected hyper-surface X in R^n will divide R^n into two connected regions: the "outside" D_0 and the "inside" D_1. Furthermore, D_1 is itself a compact manifold with boundary X

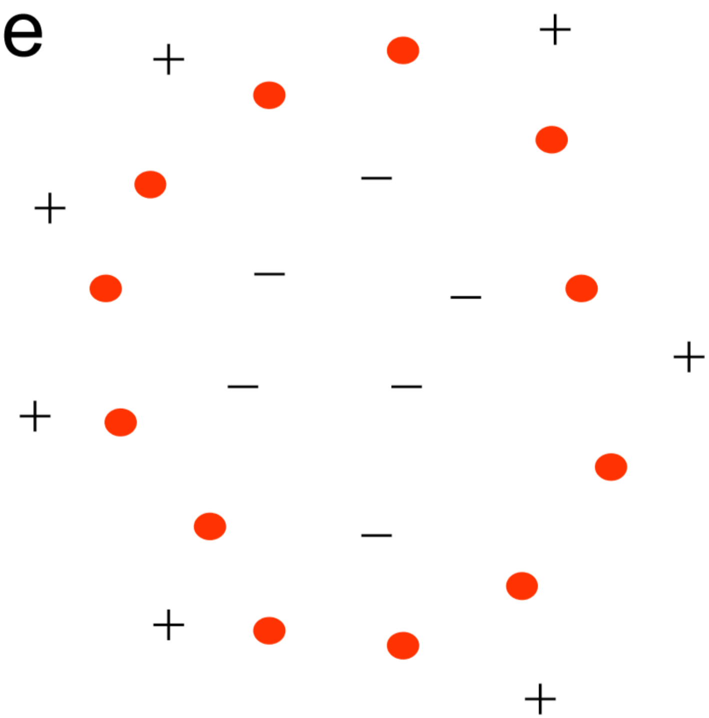




Point Cloud -> Implicit

Implicits from point samples

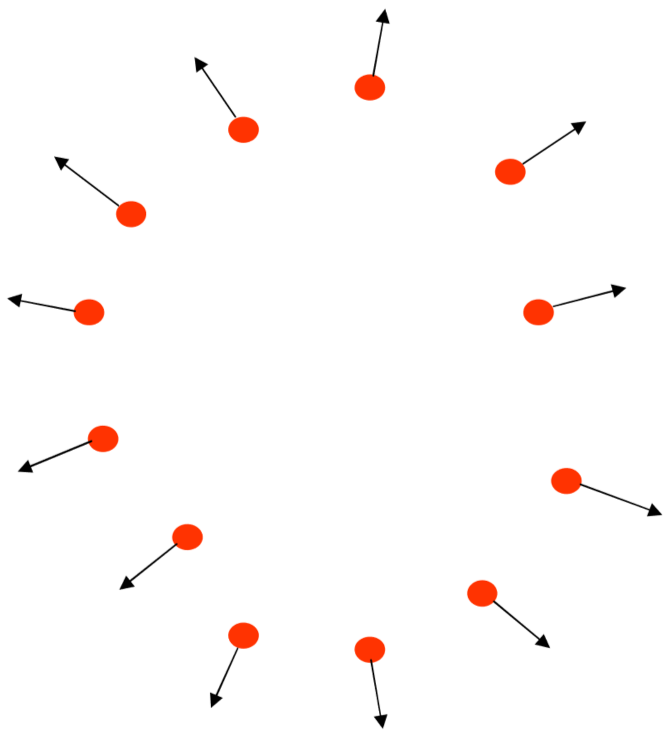
- Constraints define inside and outside
- Simple approach (Turk, O'Brien)
  - Sprinkle additional information manually
  - Make additional information soft constraints



http://graphics.stanford.edu/~mapauly/Pdfs/SigCourse04.pdf

Implicits from point samples

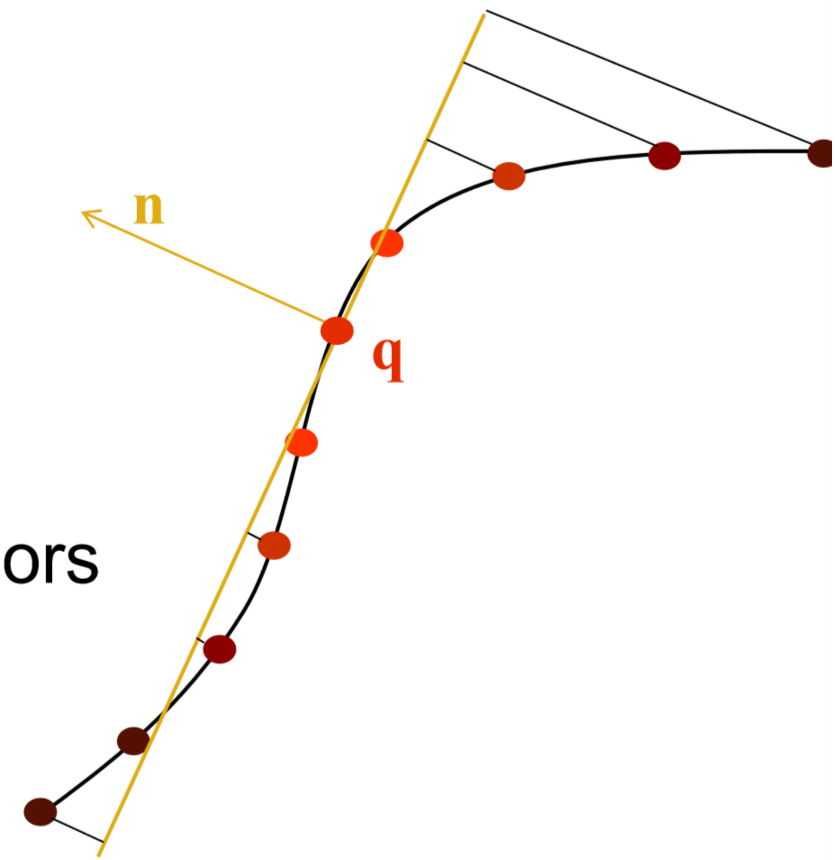
- Use normal information
- Normals could be computed from scan
- Or, normals have to be estimated



http://graphics.stanford.edu/~mapauly/Pdfs/SigCourse04.pdf

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



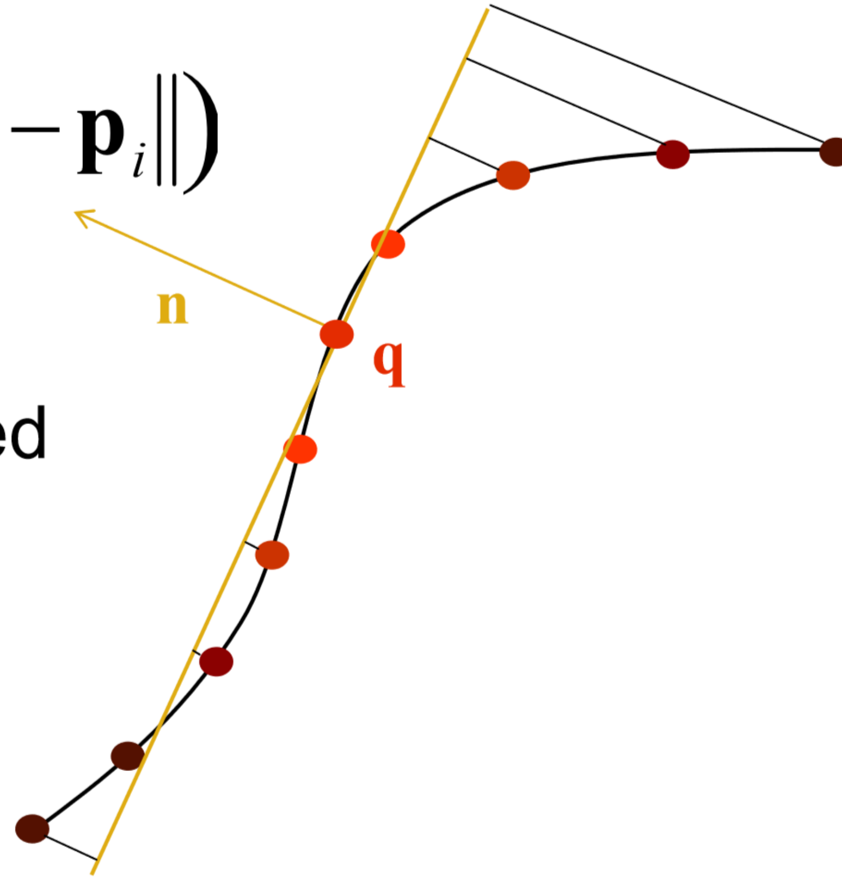
http://graphics.stanford.edu/~mapauly/Pdfs/SigCourse04.pdf

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



http://graphics.stanford.edu/~mapauly/Pdfs/SigCourse04.pdf

Estimating normals

- The constrained minimization problem

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

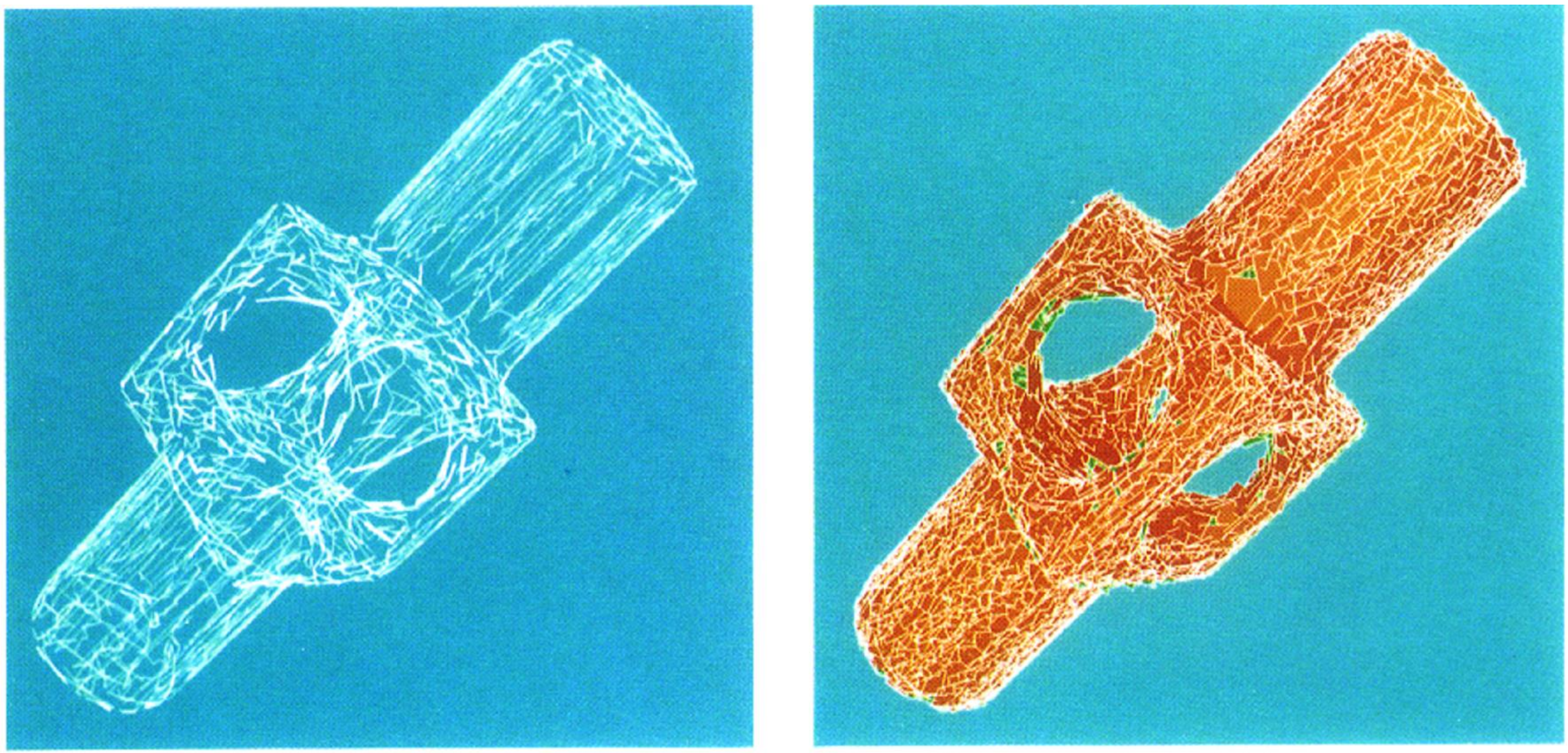
is solved by the eigenvector corresponding to the smallest eigenvalue of the following co-variance matrix

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

which is constructed as a sum of weighted outer products.

http://graphics.stanford.edu/~mapauly/Pdfs/SigCourse04.pdf

Normal orientation [Hoppe et al. 92]



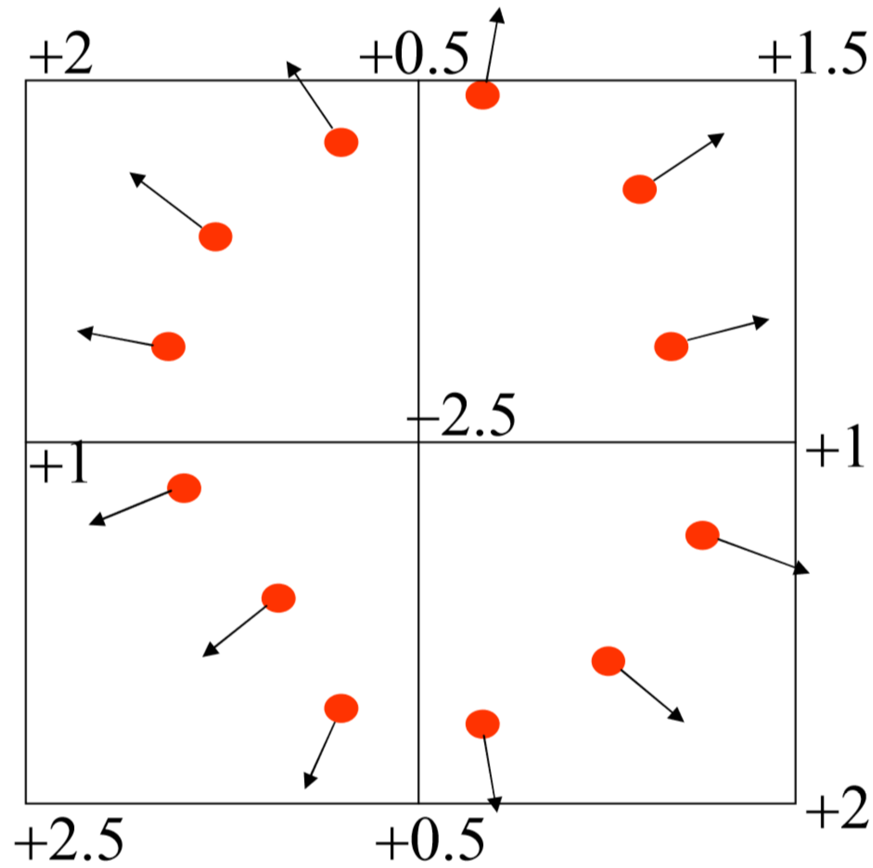
(a) Traversal order of orientation propagation

(b) Oriented tangent planes ($Tp(\mathbf{x}_i)$)

http://graphics.stanford.edu/~mapauly/Pdfs/SigCourse04.pdf

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



http://graphics.stanford.edu/~mapauly/Pdfs/SigCourse04.pdf

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

http://graphics.stanford.edu/~mapauly/Pdfs/SigCourse04.pdf

Computing Implicits

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

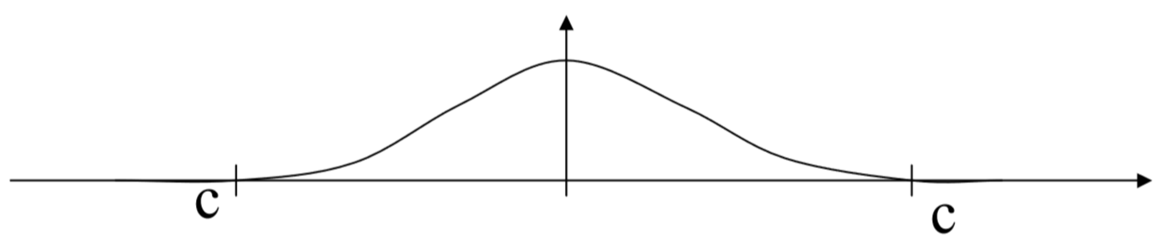
http://graphics.stanford.edu/~mapauly/Pdfs/SigCourse04.pdf

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

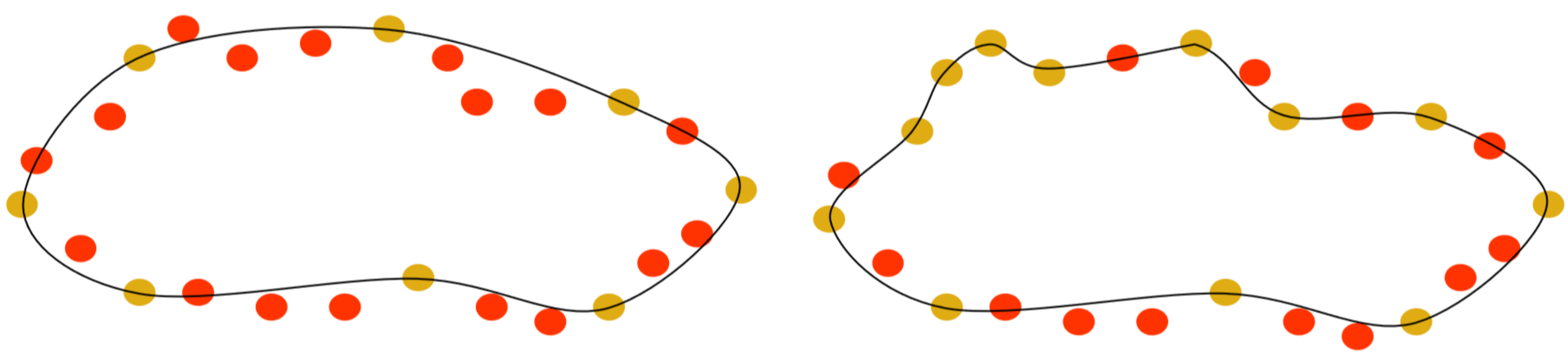


  – Compactly supported RBFs

http://graphics.stanford.edu/~mapauly/Pdfs/SigCourse04.pdf

Computing Implicits

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



http://graphics.stanford.edu/~mapauly/Pdfs/SigCourse04.pdf

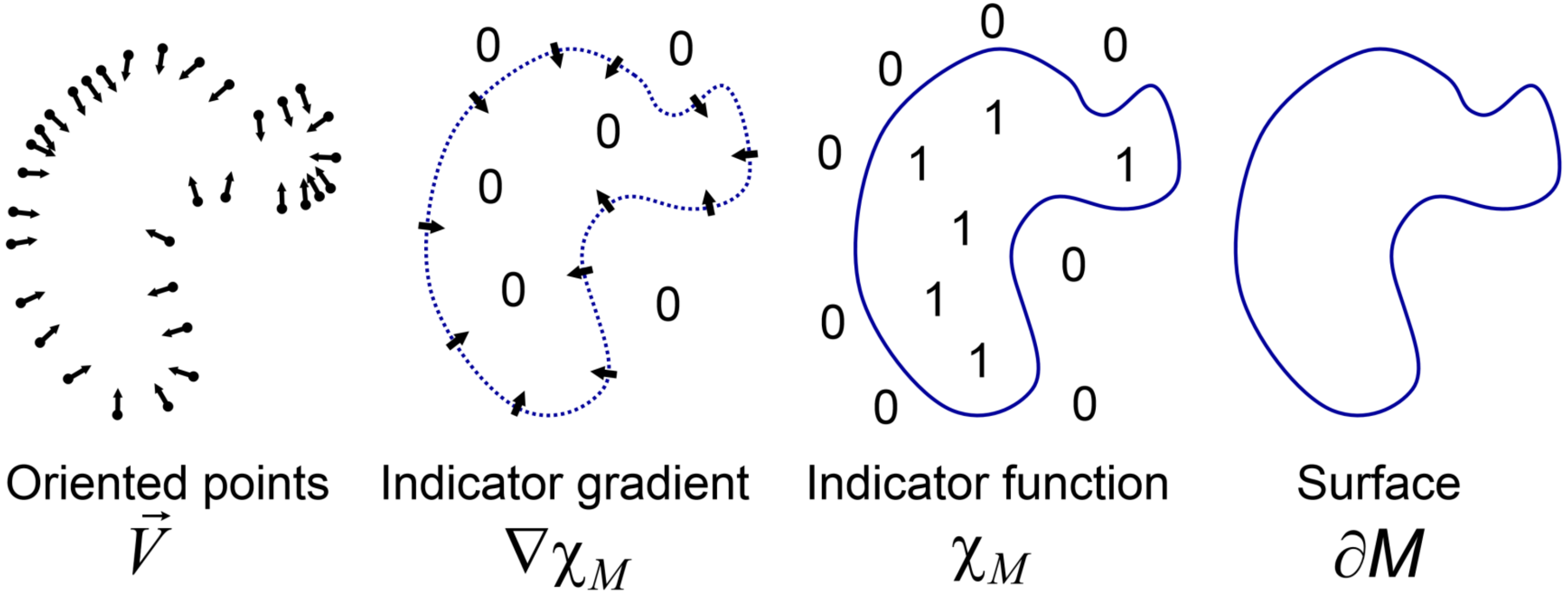
RBF Implicits - Results

- Images courtesy Greg Turk

http://graphics.stanford.edu/~mapauly/Pdfs/SigCourse04.pdf

Defining point-set surfaces [Amenta et al. 05]

Poisson surface reconstruction [Kazhdan et al. 06]



Poisson surface reconstruction [Kazhdan et al. 06]

Define the vector field:

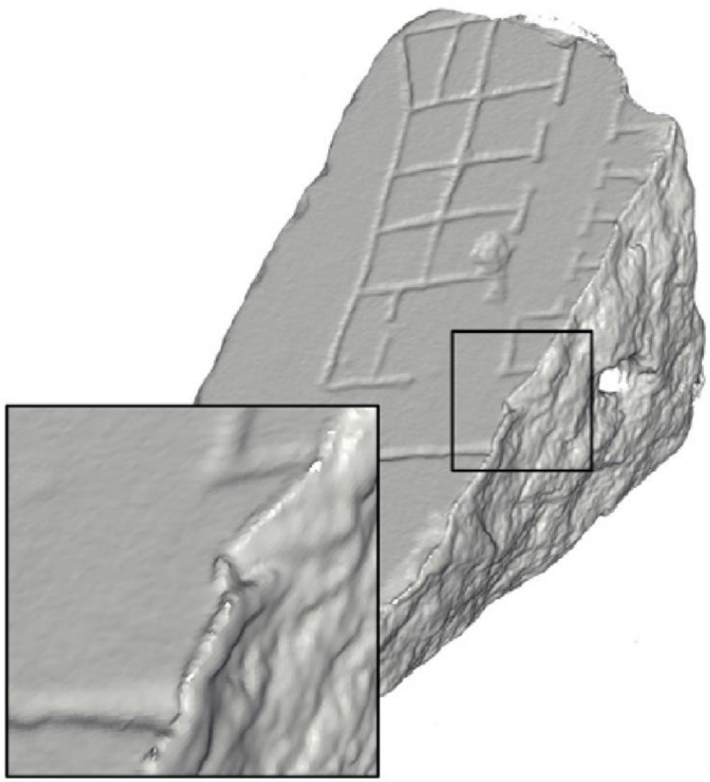
$$\nabla(\chi_M * \tilde{F})(q) = \sum_{s \in S} \int_{\mathscr{P}_s} \tilde{F}_p(q) \vec{N}_{\partial M}(p) dp$$

$$\approx \sum_{s \in S} |\mathscr{P}_s| \, \tilde{F}_{s.p}(q) \, s.\vec{N} \;\equiv\; \vec{V}(q)$$

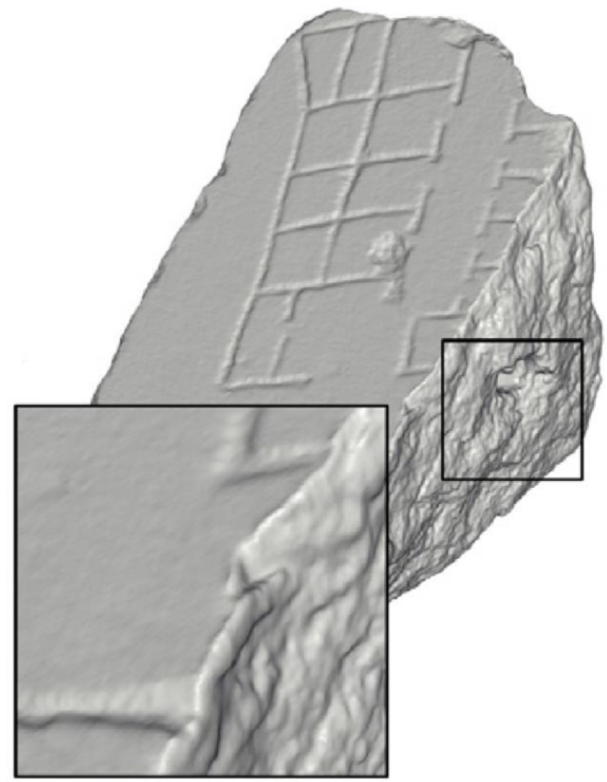
Solve the Poisson equation:

$$\Delta\tilde{\chi} = \nabla \cdot \vec{V}.$$

Poisson surface reconstruction [Kazhdan et al. 06]



VRIP

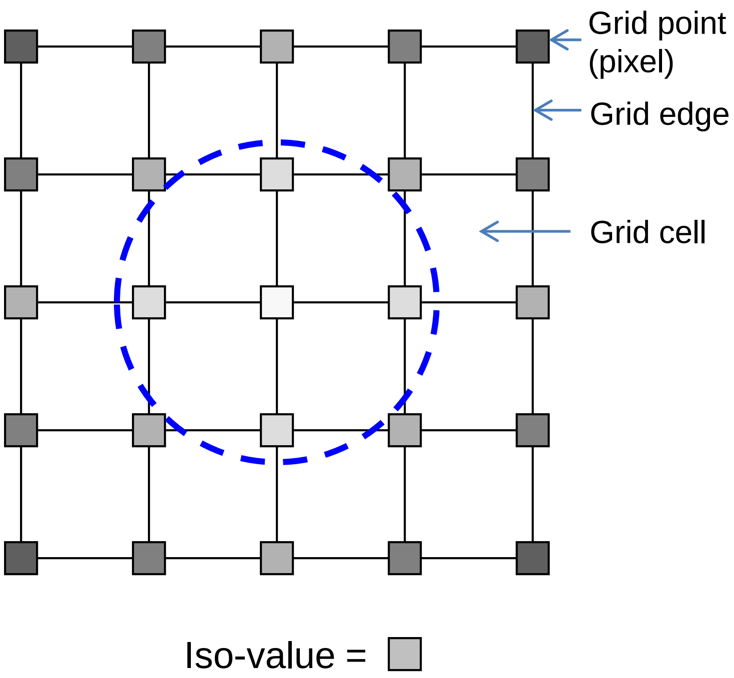


Poisson Surface Reconstruction

Implicit Surface -> Mesh

Contouring (On A Grid)

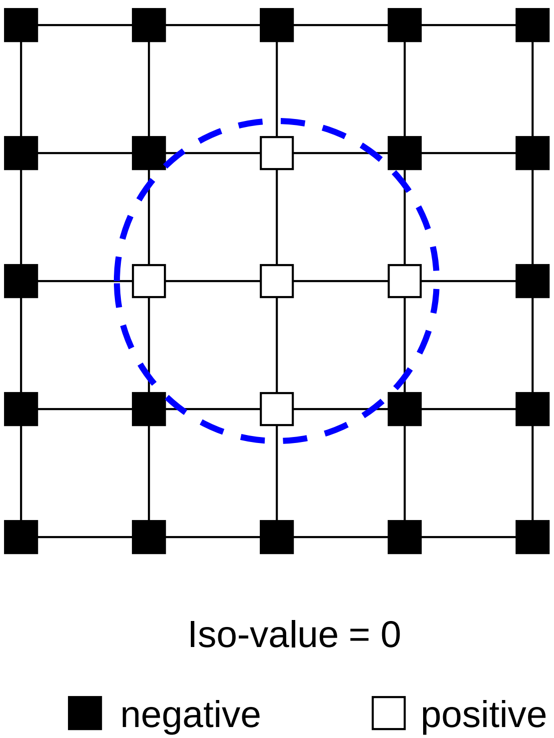
- Input
  - A grid where each grid point (pixel or voxel) has a value (color)
  - An iso-value (threshold)
- Output
  - A closed, manifold, non-intersecting polyline (2D) or mesh (3D) that separates grid points above the iso-value from those that are below the iso-value.



Slide Credit: Tao Ju

Contouring (On A Grid)

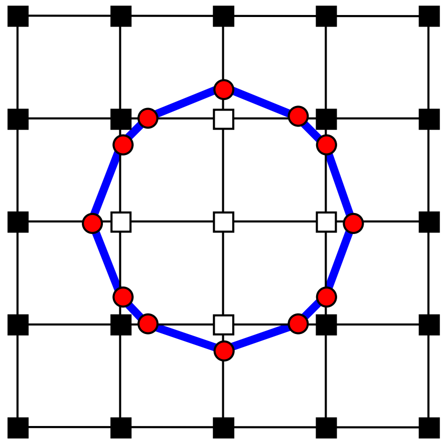
- Input
  - A grid where each grid point (pixel or voxel) has a value (color)
  - An iso-value (threshold)
- Output
  - Equivalently, we extract the zero-contour (separating negative from positive) after subtracting the iso-value from the grid points



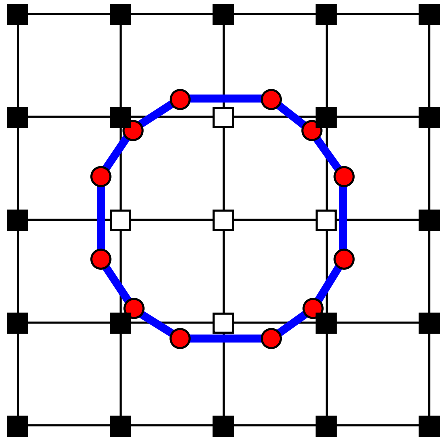
Slide Credit: Tao Ju

Algorithms

- Primal methods
  - Marching Squares (2D), Marching Cubes (3D)
  - Placing vertices on grid edges



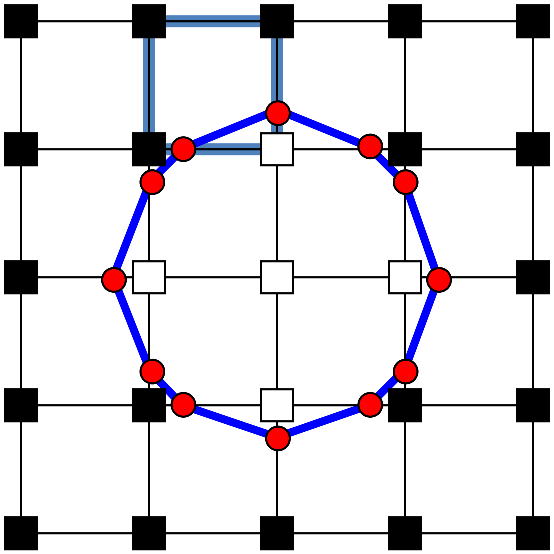
- Dual methods
  - Dual Contouring (2D,3D)
  - Placing vertices in grid cells



Slide Credit: Tao Ju

Marching Squares (2D)

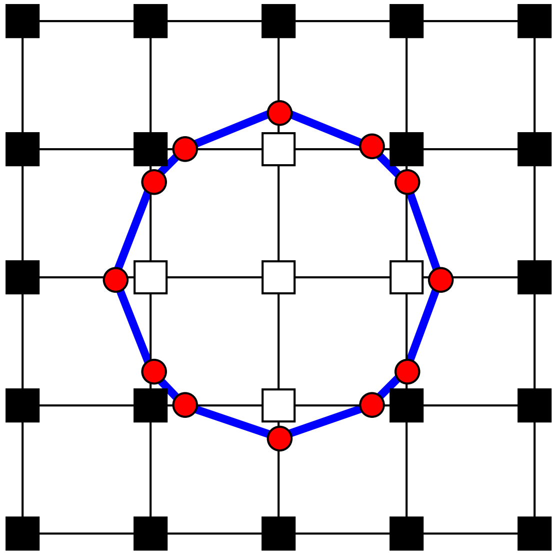
- For each grid cell with a sign change
  - Create one vertex on each grid edge with a sign change
  - Connect vertices by lines



Slide Credit: Tao Ju

Marching Squares (2D)

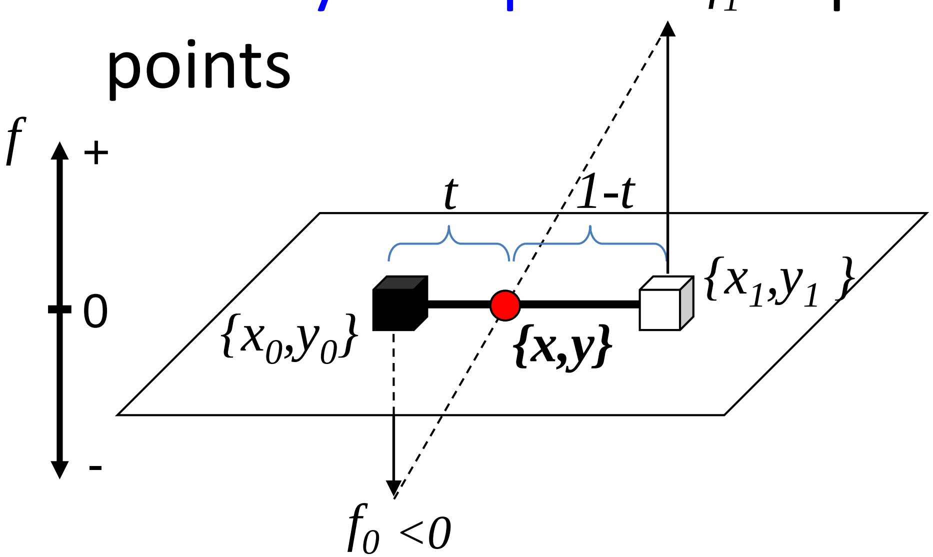
- For each grid cell with a sign change
  - Create one vertex on each grid edge with a sign change
  - Connect vertices by lines



Slide Credit: Tao Ju

Marching Squares (2D)

- Creating vertices: linear interpolation
  - Assuming the underlying, continuous function is linear on the grid edge
  - Linearly interpolate the positions of the two grid points



$$t = \frac{f_0}{f_0 - f_1}$$

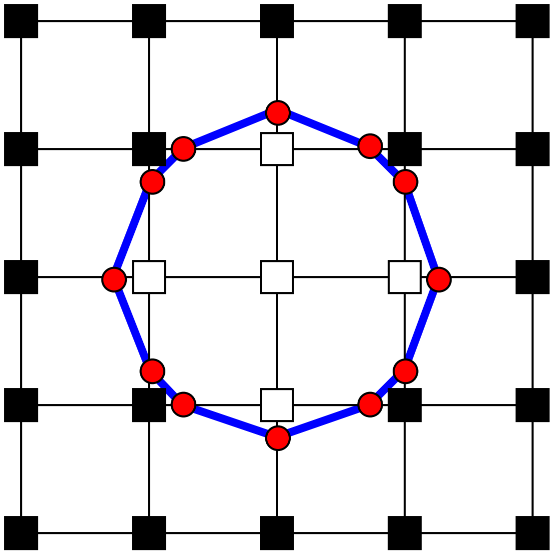
$$x = x_0 + t(x_1 - x_0)$$

$$y = y_0 + t(y_1 - y_0)$$

Slide Credit: Tao Ju

Marching Squares (2D)

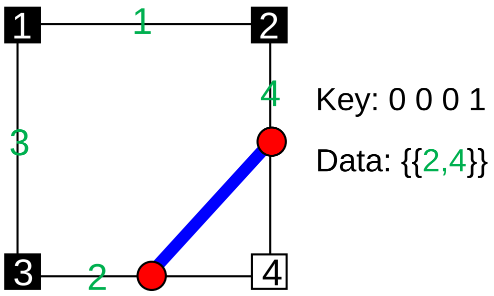
- For each grid cell with a sign change
  - Create one vertex on each grid edge with a sign change
  - Connect vertices by lines

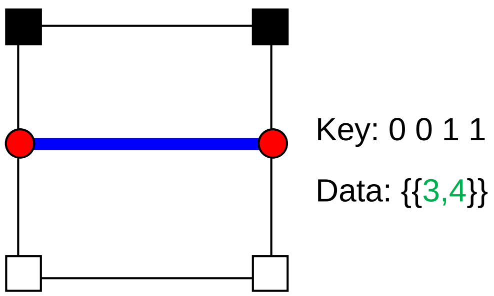


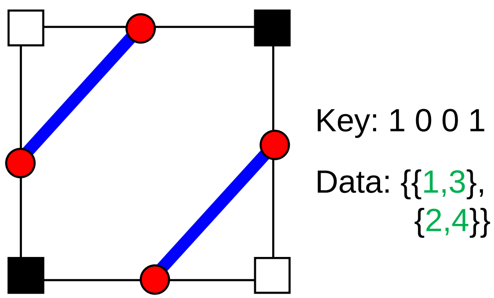
Slide Credit: Tao Ju

Marching Squares (2D)

- Connecting vertices by lines
  - Lines shouldn't intersect
  - Each vertex is used once
    - So that it will be used exactly twice by the two cells incident on the edge
- Two approaches
  - Do a walk around the grid cell
    - Connect consecutive pair of vertices
  - Or, using a pre-computed look-up table
    - 2^4=16 sign configurations
    - For each sign configuration, it stores the indices of the grid edges whose vertices make up the lines.



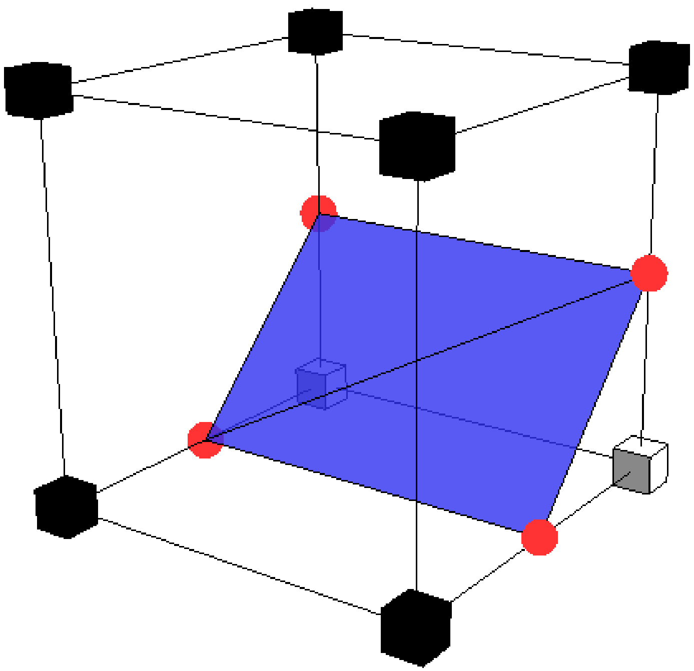




Slide Credit: Tao Ju

Marching Cubes (3D)

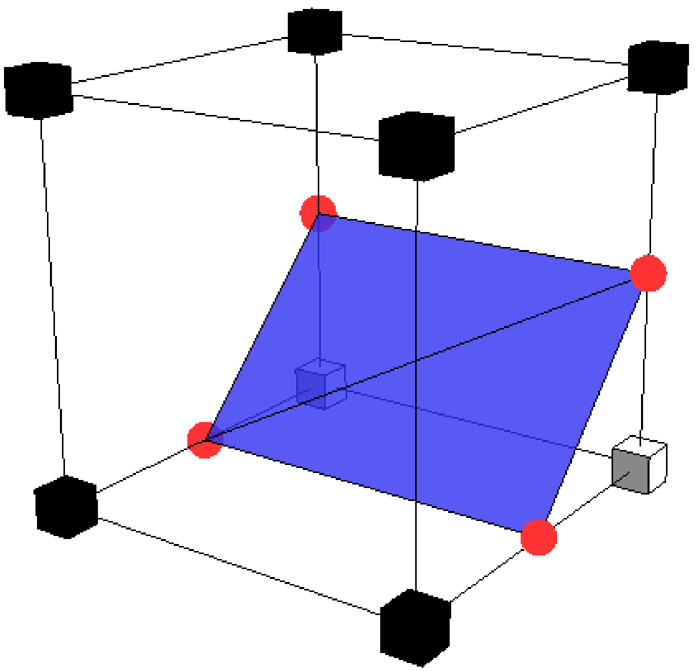
- For each grid cell with a sign change
  - Create one vertex on each grid edge with a sign change (using linear interpolation)
  - Connect vertices into triangles



Slide Credit: Tao Ju

Marching Cubes (3D)

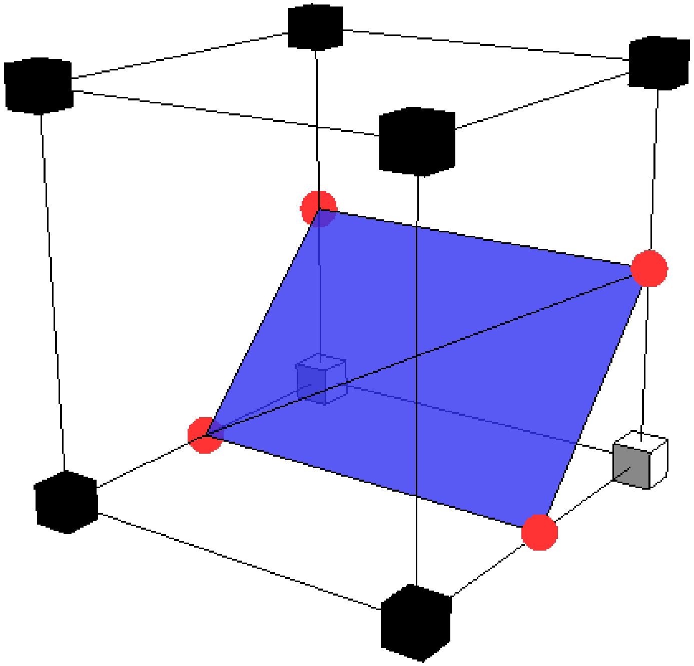
- For each grid cell with a sign change
  - Create one vertex on each grid edge with a sign change (using linear interpolation)
  - Connect vertices into triangles



Slide Credit: Tao Ju

Marching Cubes (3D)

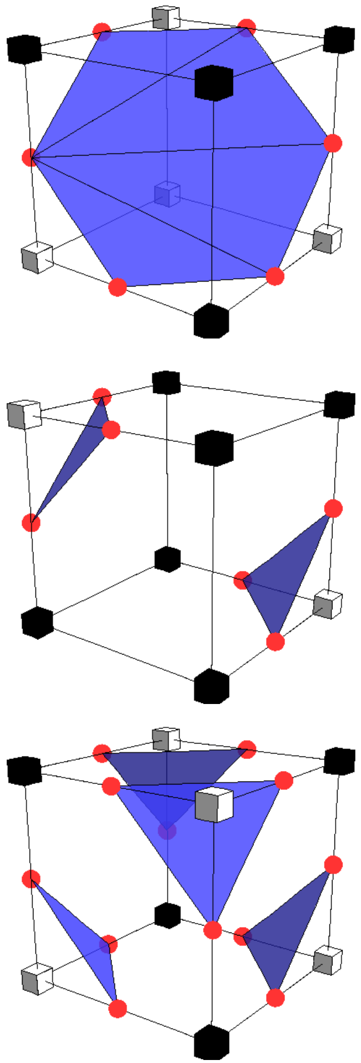
- Connecting vertices by triangles
  - Triangles shouldn't intersect
  - To be a closed manifold:
    - Each vertex used by a triangle "fan"
    - Each mesh edge used by 2 triangles (if inside grid cell) or 1 triangle (if on a grid face)



Slide Credit: Tao Ju

Marching Cubes (3D)

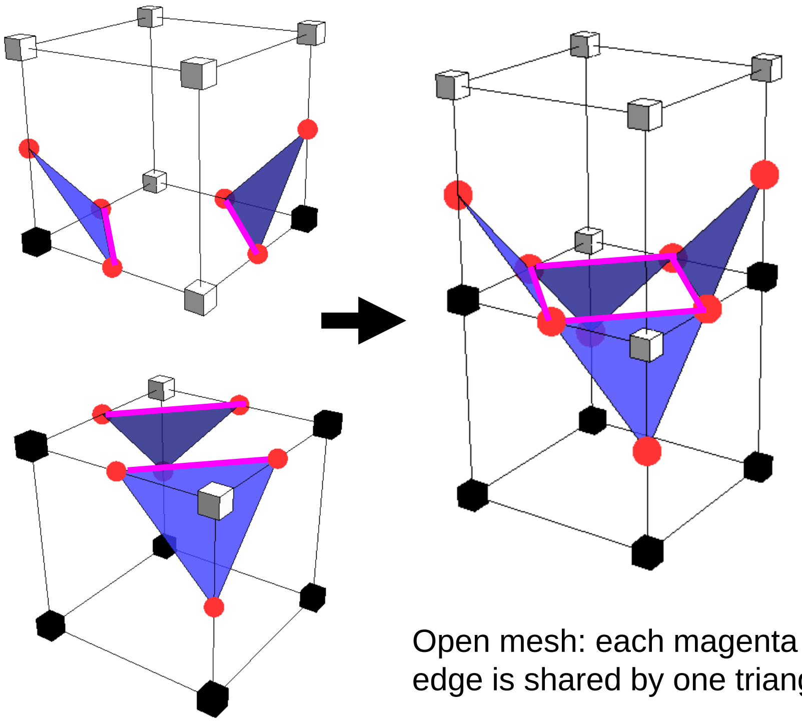
- Connecting vertices by triangles
  - Triangles shouldn't intersect
  - To be a closed manifold:
    - Each vertex used by a triangle "fan"
    - Each mesh edge used by 2 triangles (if inside grid cell) or 1 triangle (if on a grid face)



Slide Credit: Tao Ju

Marching Cubes (3D)

- Connecting vertices by triangles
  - Triangles shouldn't intersect
  - To be a closed manifold:
    - Each vertex used by a triangle "fan"
    - Each mesh edge used by 2 triangles (if inside grid cell) or 1 triangle (if on a grid face)

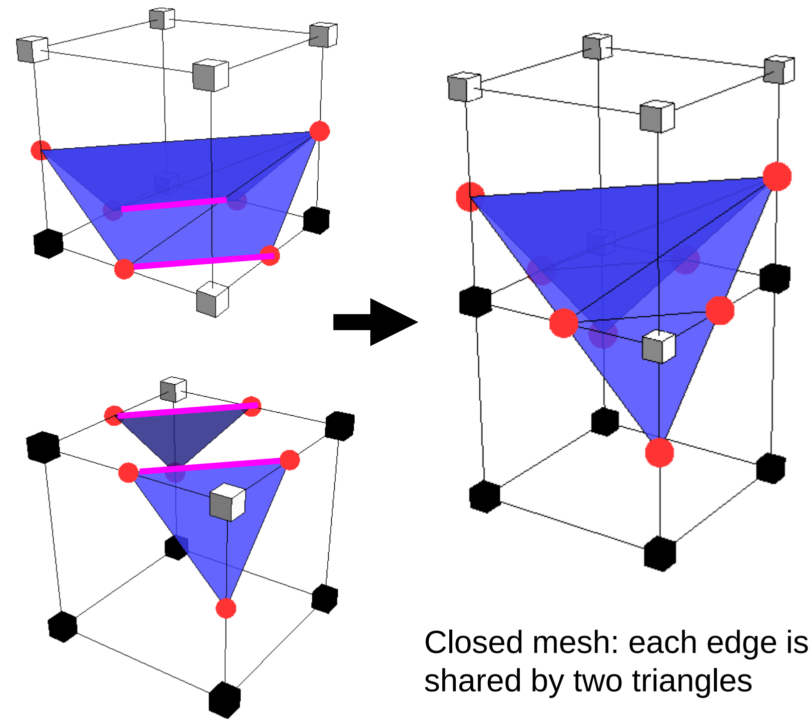


Open mesh: each magenta edge is shared by one triangle

Slide Credit: Tao Ju

Marching Cubes (3D)

- Connecting vertices by triangles
  - Triangles shouldn't intersect
  - To be a closed manifold:
    - Each vertex used by a triangle "fan"
    - Each mesh edge used by 2 triangles (if inside grid cell) or 1 triangle (if on a grid face)
    - Each mesh edge on the grid face is shared between adjacent cells

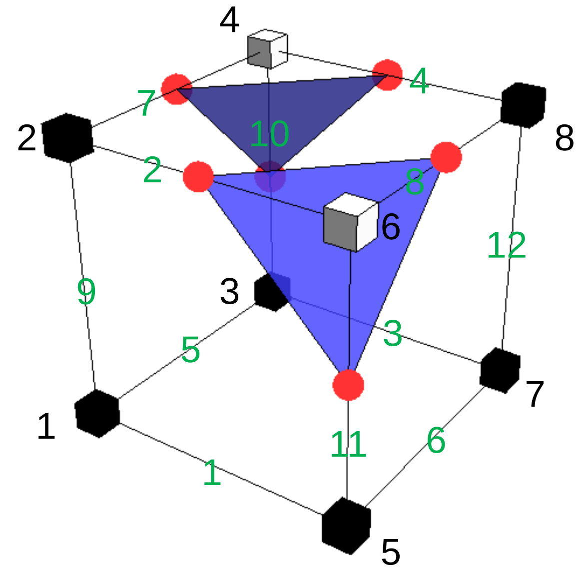


Closed mesh: each edge is shared by two triangles

Slide Credit: Tao Ju

Marching Cubes (3D)

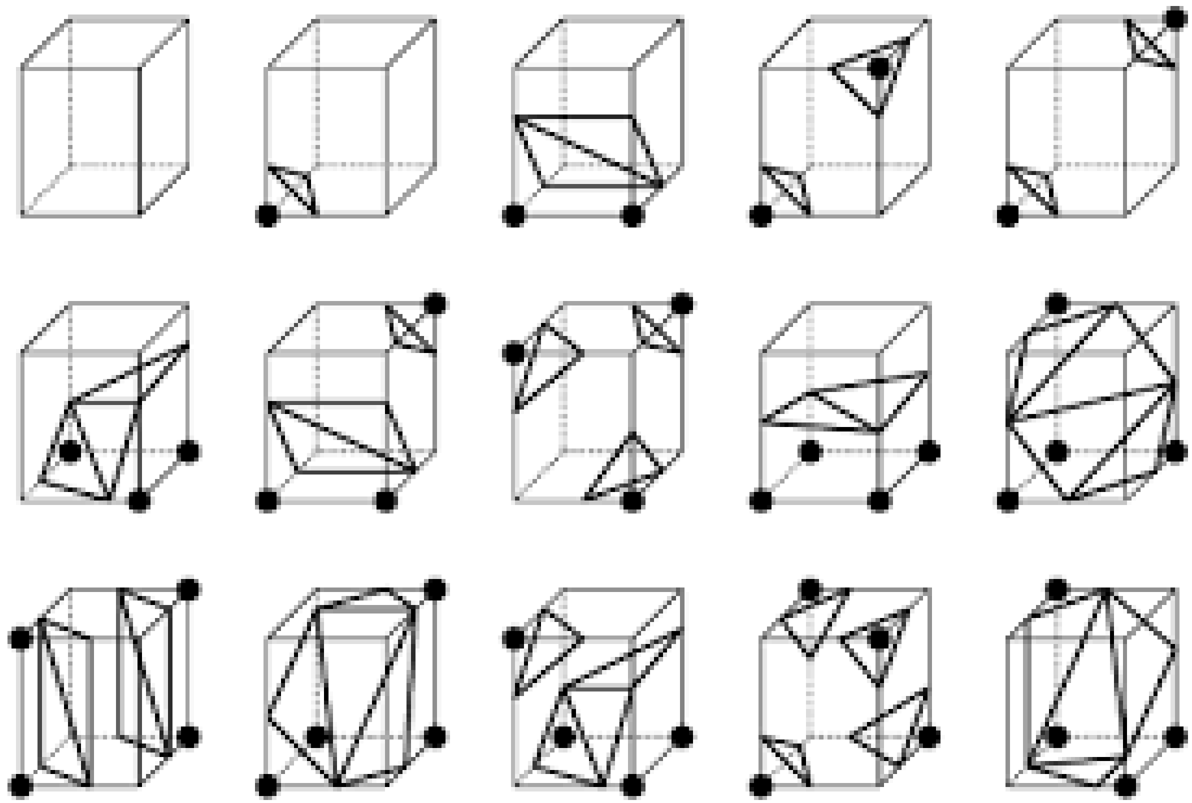
- Connecting vertices by triangles
  - Triangles shouldn't intersect
  - To be a closed manifold:
    - Each vertex used by a triangle "fan"
    - Each mesh edge used by 2 triangles (if inside grid cell) or 1 triangle (if on a grid face)
    - Each mesh edge on the grid face is shared between adjacent cells
- Look-up table
  - 2^8=256 sign configurations
  - For each sign configuration, it stores indices of the grid edges whose vertices make up the triangles



Sign: "0 0 0 1 0 1 0 0"
Triangles: {{2,8,11},{4,7,10}}

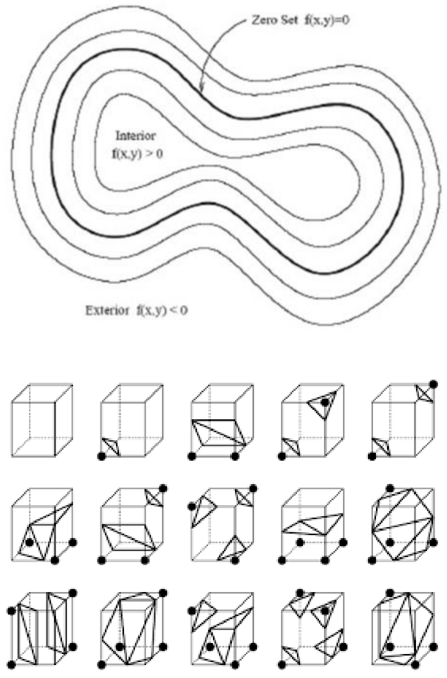
Slide Credit: Tao Ju

Lookup Table

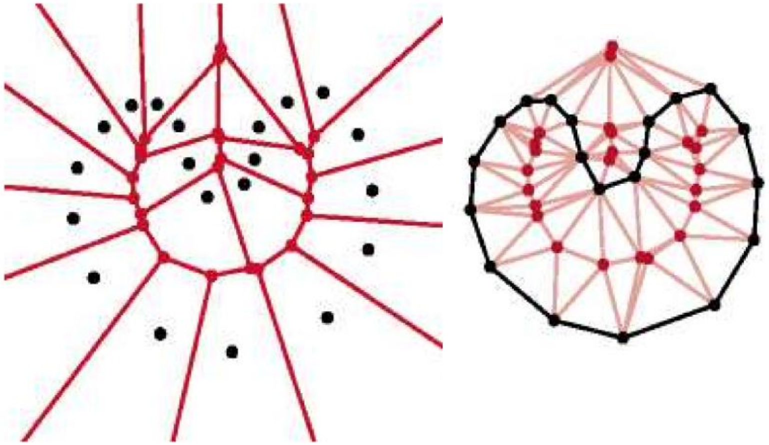


Slide Credit: Tao Ju

Two Approaches



Implicit Surface -> Contouring



Computational Geometry Based