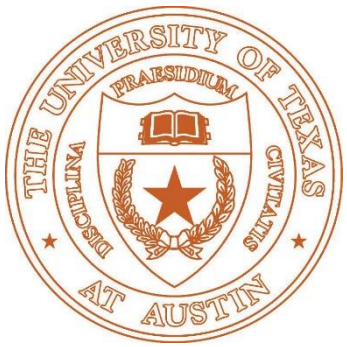
CS354 Computer Graphics
Surface Representation V

Qixing Huang

March 19th 2018



Today's Topic

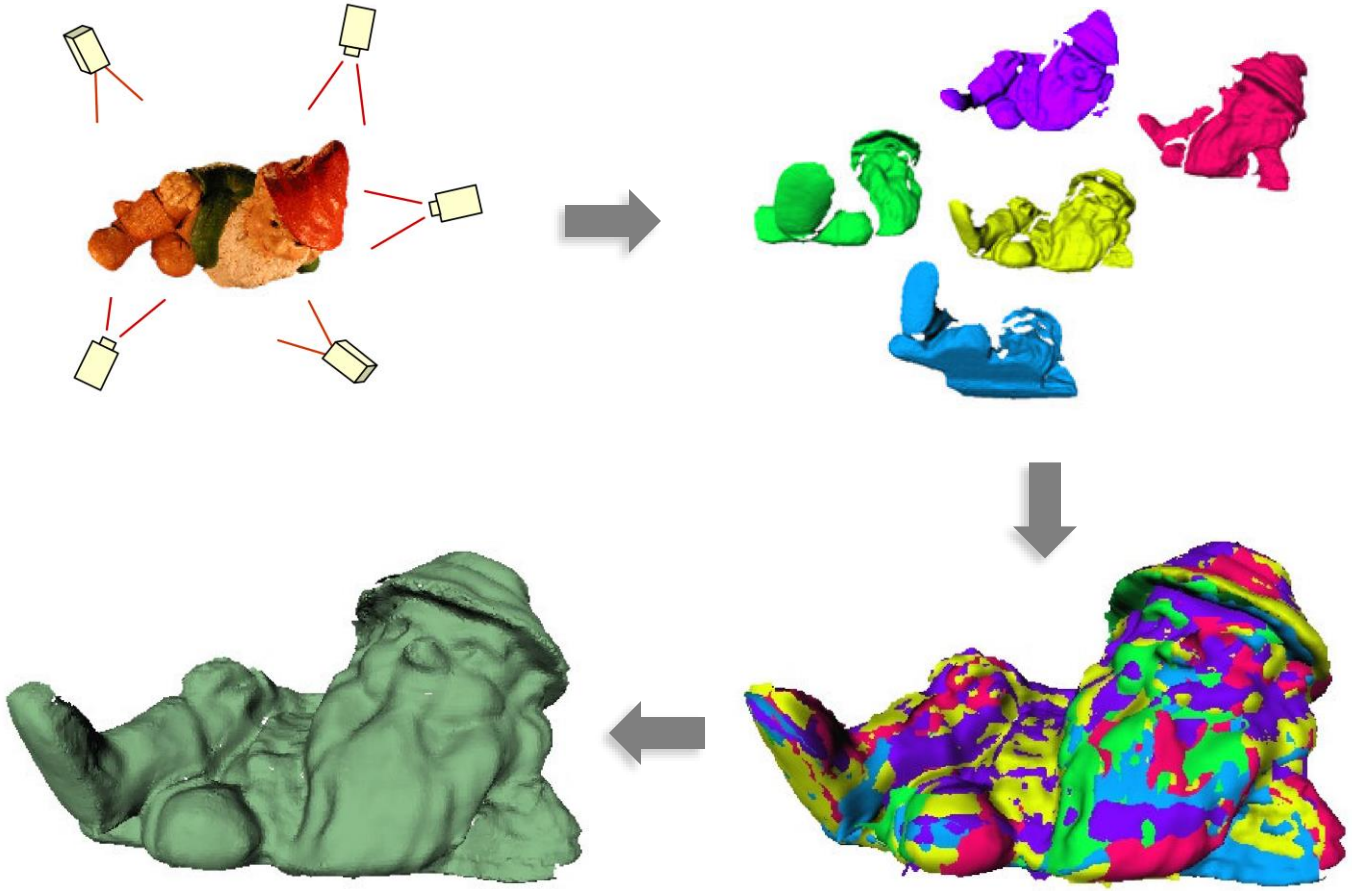
- Geometry Reconstruction Pipeline

- Marching cube for implicit surface/mesh conversion

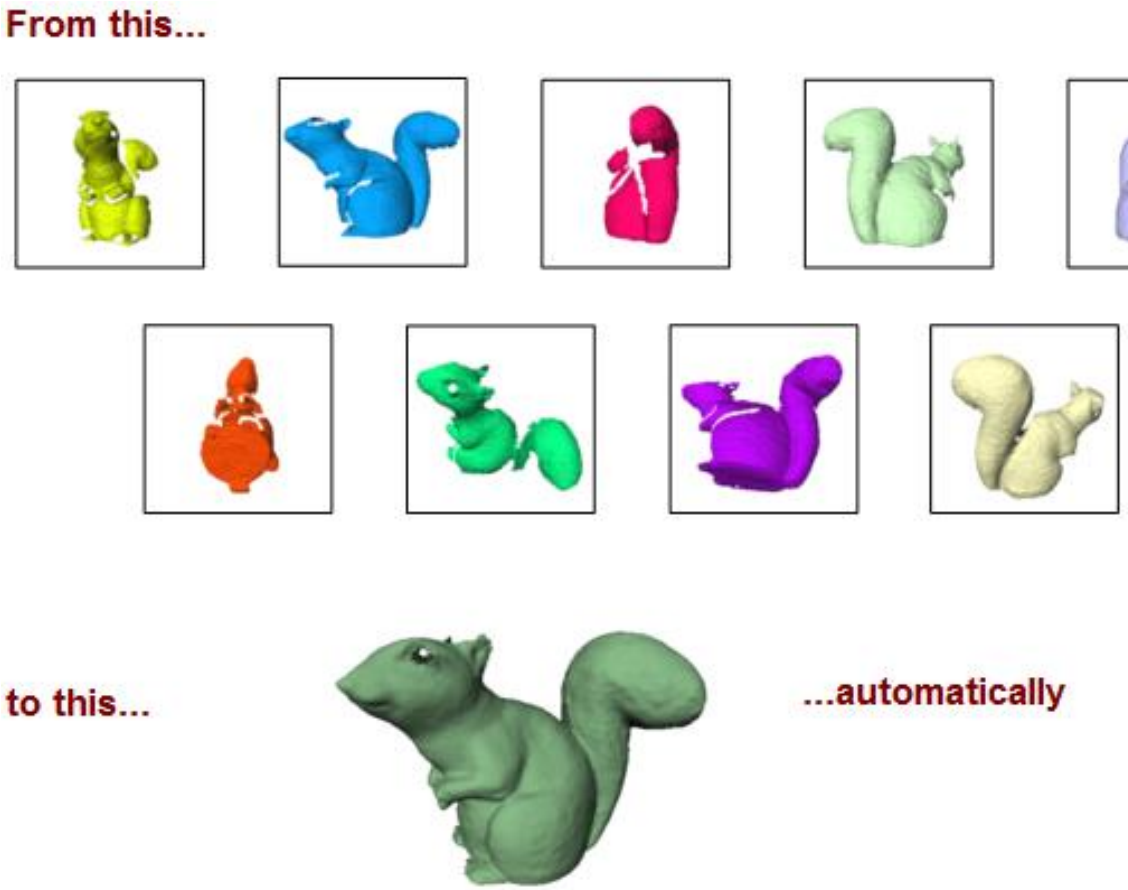
Next Lecture

- TA will give the next lecture
- Half-edge structure
- Mesh simplification

Geometry Reconstruction Pipeline

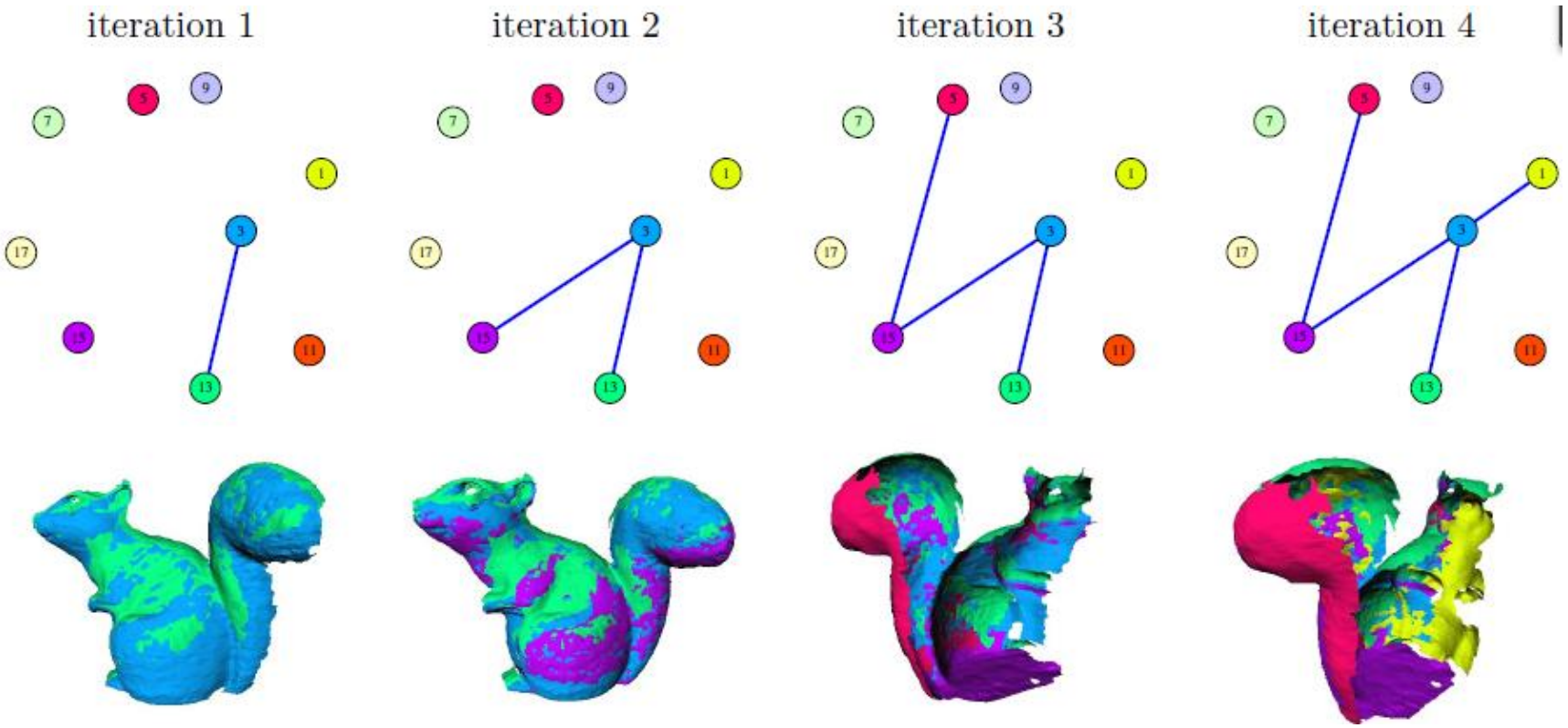


Spanning tree based

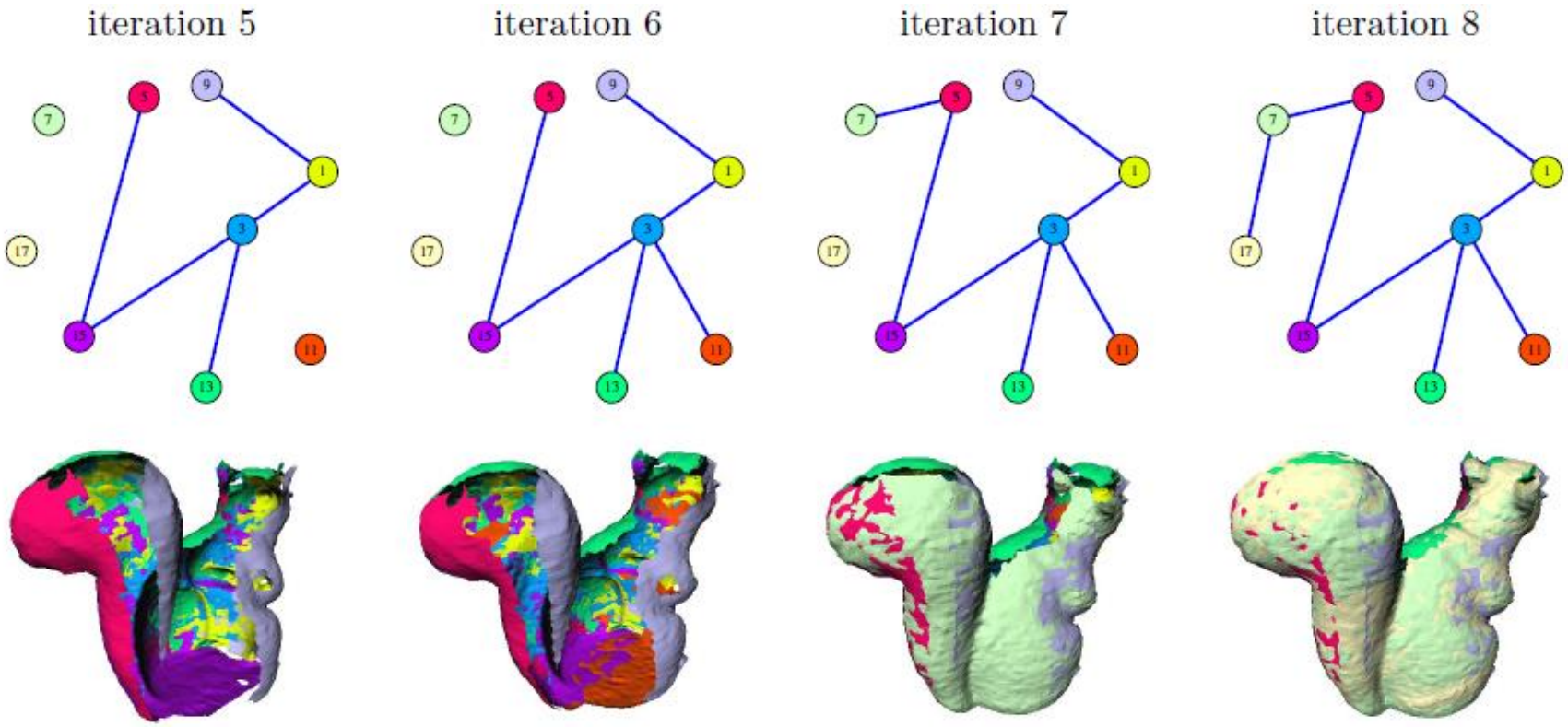


Automatic Three-dimensional Modeling from Reality, PhD thesis, D. Huber, Robotics Institute, Carnegie Mellon University, 2002

Spanning tree based

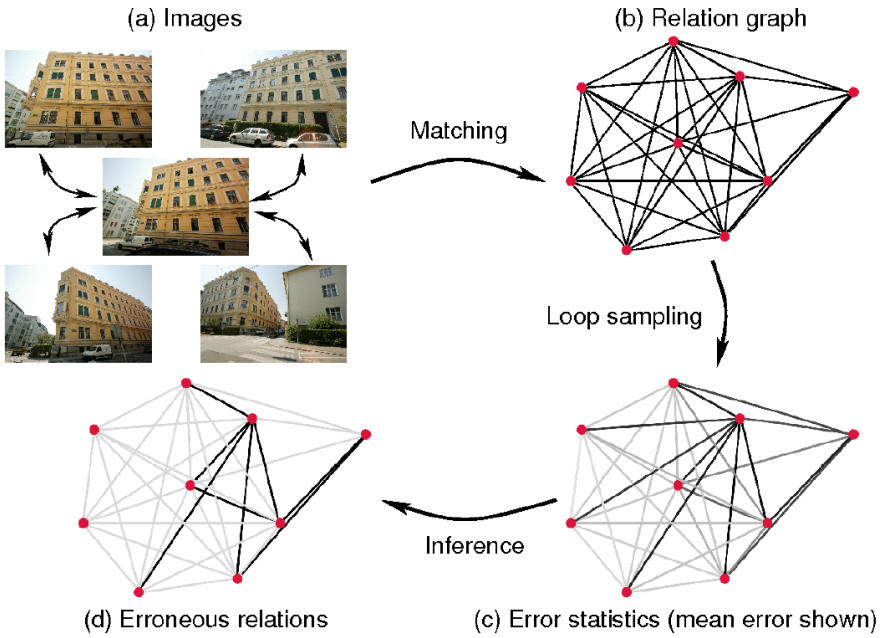


Spanning tree based



Issue: A single incorrect match can destroy everything

Detecting inconsistent cycles



large for inconsistent cycles

maximize $\sum_L \rho_L x_L$

subject to $x_L \geq x_e \; \forall e \in L,$

$$x_L \leq \sum_{e \in L} x_e,$$

$$x_L \in [0, 1],$$

$$x_e \in [0, 1].$$

Disambiguating visual relations using loop constraints, C. Zach, M. Klopscjotz, and M. POLLEFEYS, *CVPR'10*

Rotation

[Wang and Singer'13]

$$\mathbf{R} = \begin{bmatrix} I_3 & \cdots & \mathbf{R}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{R}_{1n}^T & \cdots & I_3 \end{bmatrix}$$

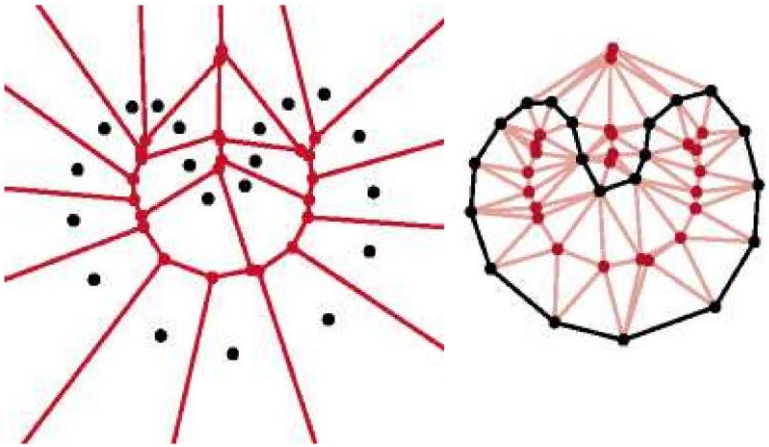
minimize $$\sum_{(i,j)\in\mathcal{G}} \|\mathbf{R}_{ij} - \mathbf{R}_{ij}^{init}\|_{\mathcal{F}}$$

subject to $$\mathbf{R} \succeq 0$$

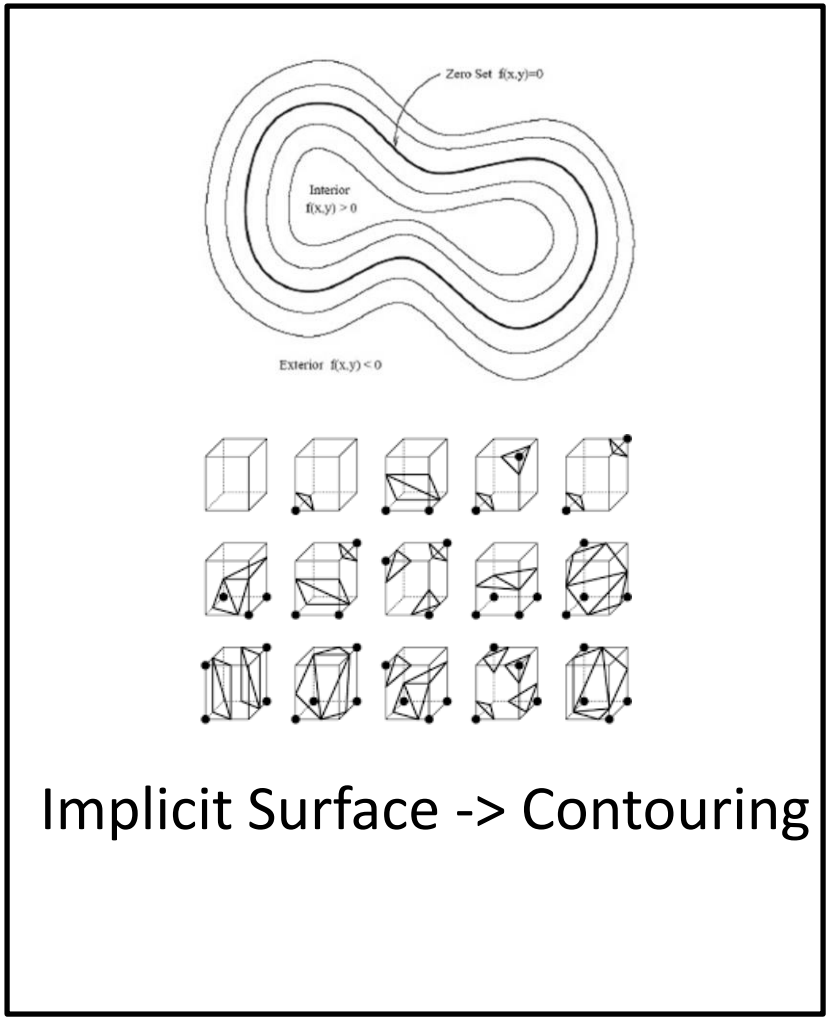
$$\mathbf{R}_{ii} = I_3, \quad 1 \le i \le n$$

$$\mathbf{R}_{ij} \in convex - hull(SO(3)), \quad 1 \le i < j \le n$$

Two Approaches



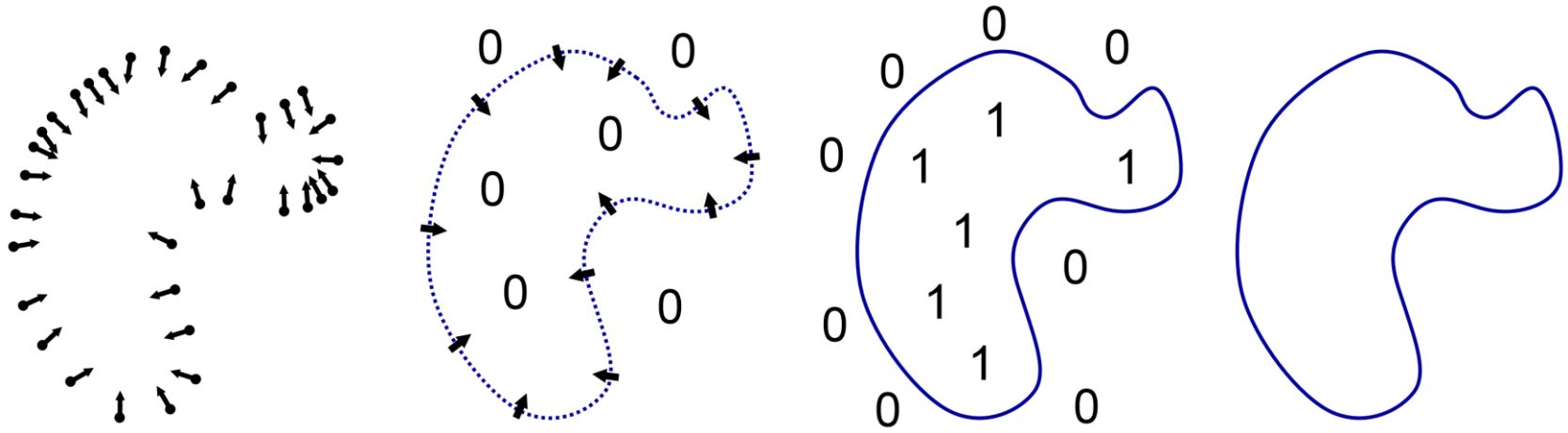
Computational Geometry Based



Implicit Surface -> Contouring

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:

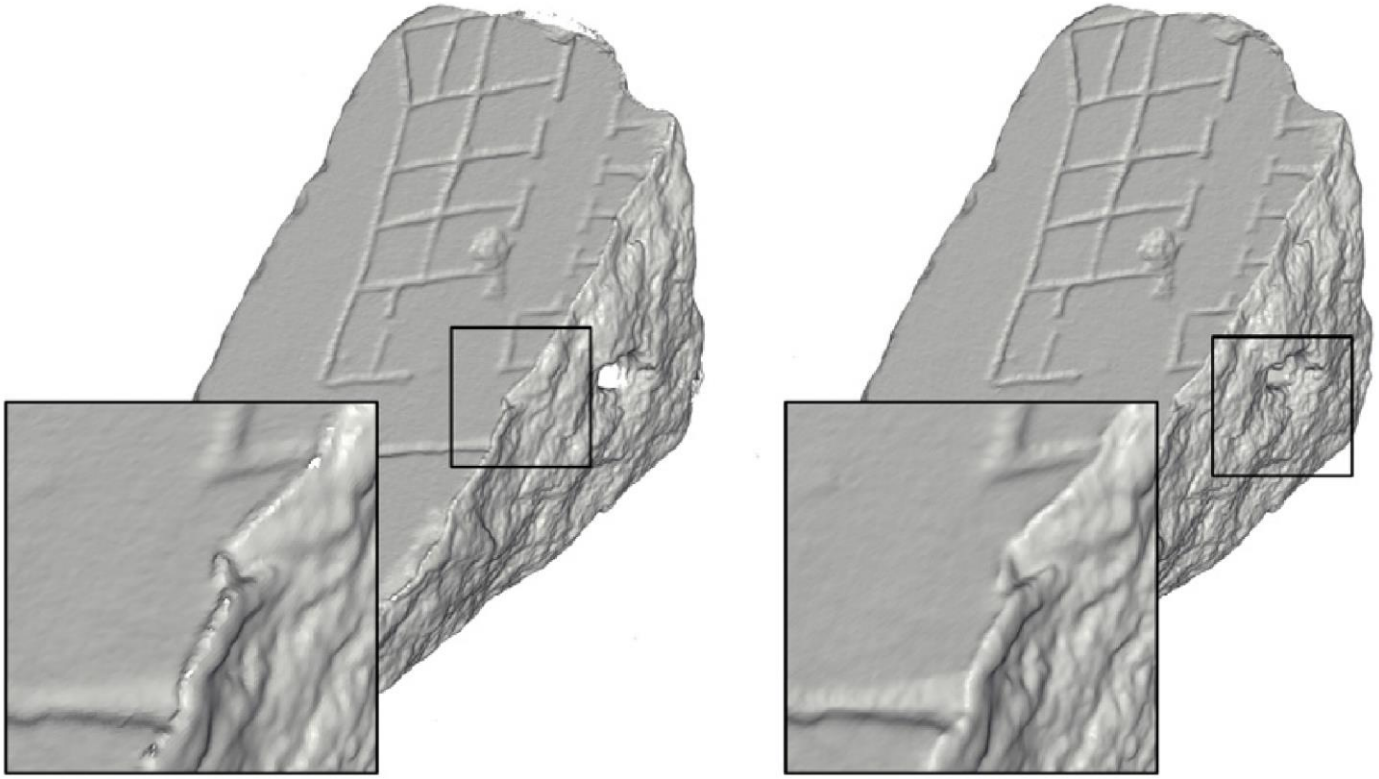
$$\nabla(\chi_M * \tilde{F})(q) = \sum_{s \in S} \int_{\mathscr{P}_s} \tilde{F}_p(q) \vec{N}_{\partial M}(p) dp$$

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

Solve the Poisson equation:

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

Poisson surface reconstruction [Kazhdan et al. 06]



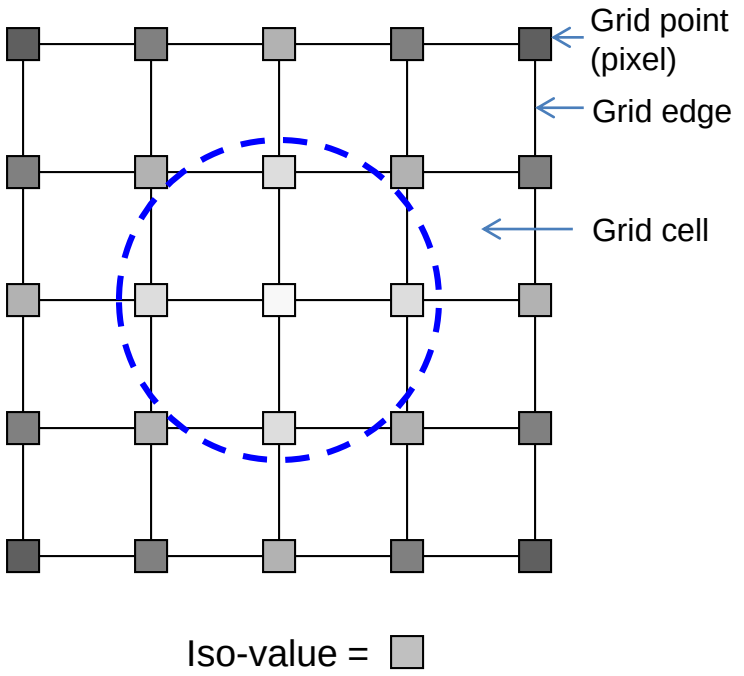
VRIP

Poisson Surface Reconstruction

Contouring

Contouring (On A Grid)

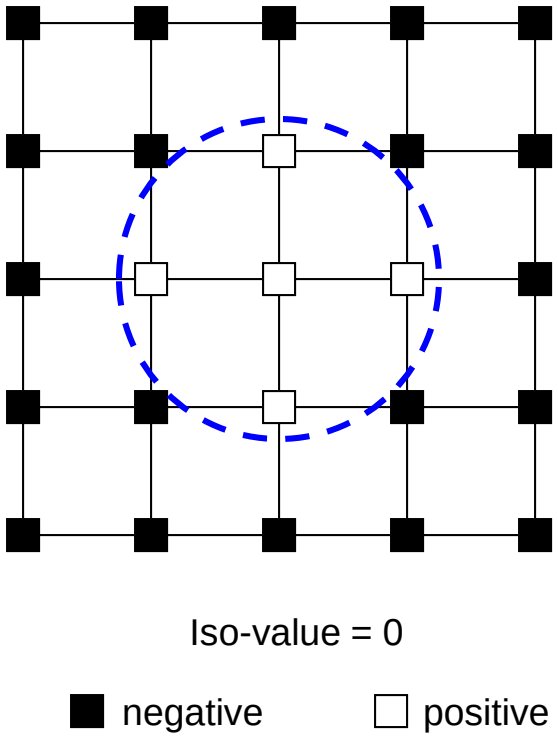
- Input
  - A grid where each grid point (pixel or voxel) has a value (color)
  - An iso-value (threshold)
- Output
  - A closed, manifold, non-intersecting polyline (2D) or mesh (3D) that separates grid points above the iso-value from those that are below the iso-value.



Slide Credit: Tao Ju

Contouring (On A Grid)

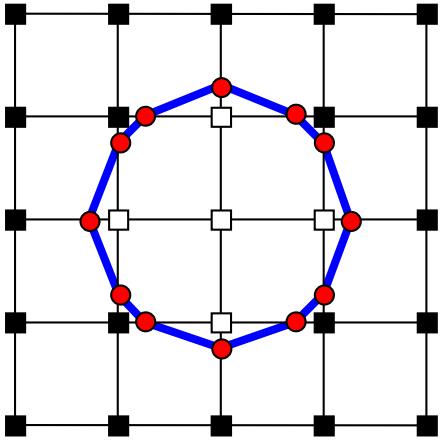
- Input
  - A grid where each grid point (pixel or voxel) has a value (color)
  - An iso-value (threshold)
- Output
  - Equivalently, we extract the zero-contour (separating negative from positive) after subtracting the iso-value from the grid points



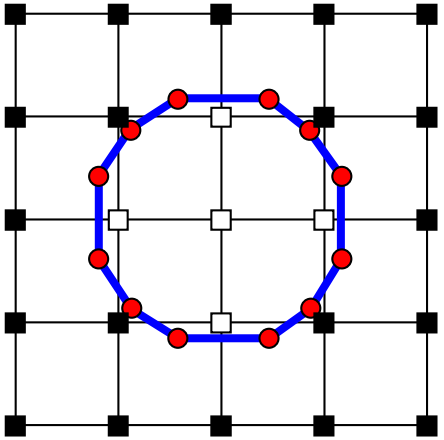
Slide Credit: Tao Ju

Algorithms

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



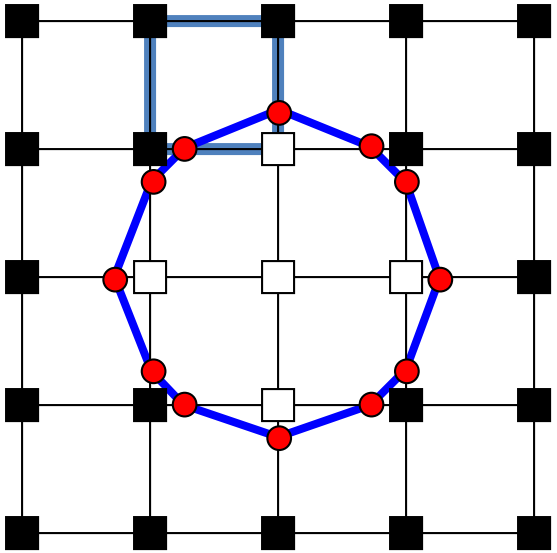
- Dual methods
  - Dual Contouring (2D,3D)
  - Placing vertices in grid cells



Slide Credit: Tao Ju

Marching Squares (2D)

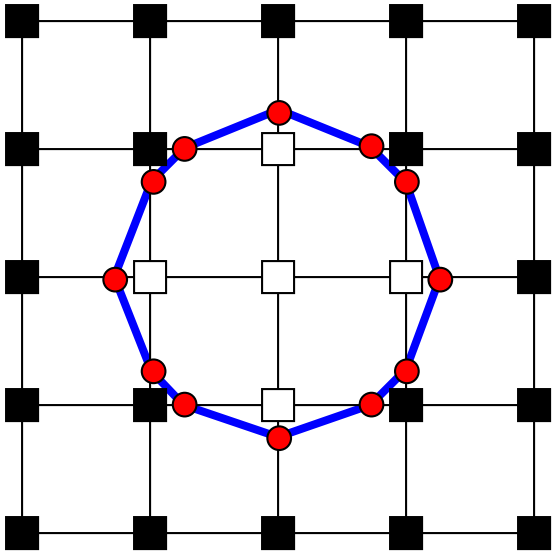
- For each grid cell with a sign change
  - Create one vertex on each grid edge with a sign change
  - Connect vertices by lines



Slide Credit: Tao Ju

Marching Squares (2D)

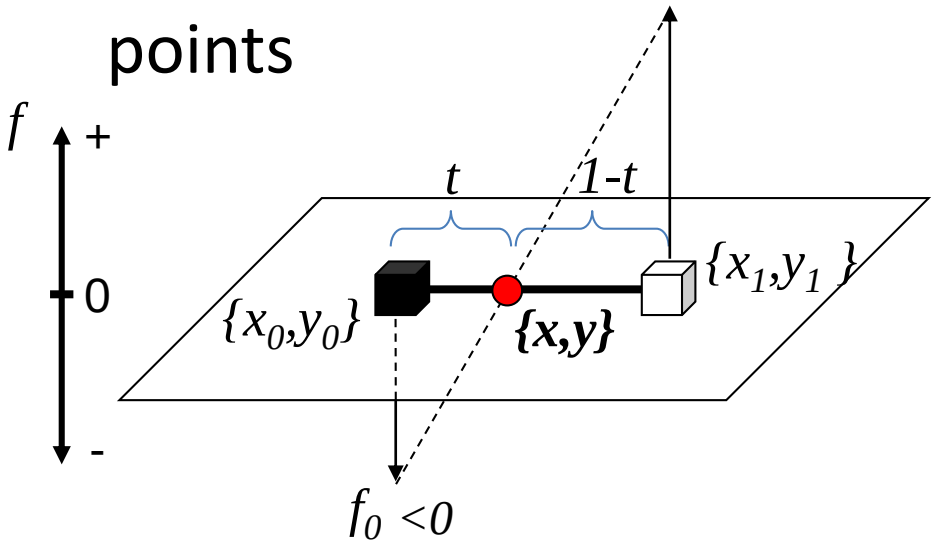
- For each grid cell with a sign change
  - Create one vertex on each grid edge with a sign change
  - Connect vertices by lines



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



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

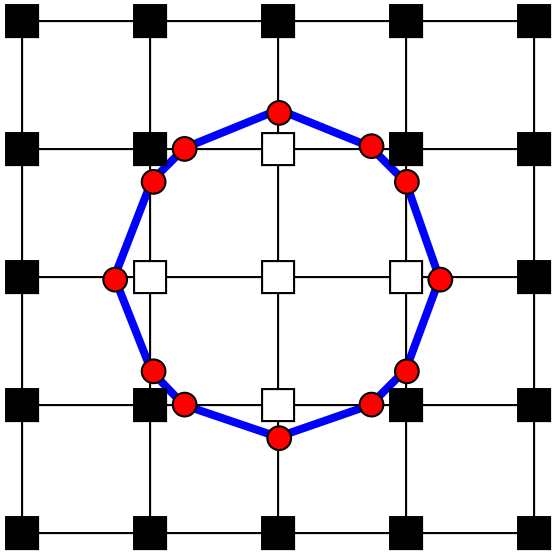
$$x = x_0 + t(x_1 - x_0)$$

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

Slide Credit: Tao Ju

Marching Squares (2D)

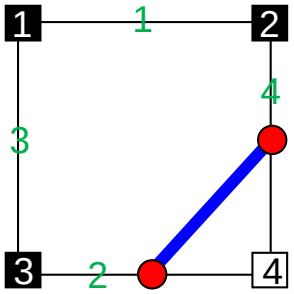
- For each grid cell with a sign change
  - Create one vertex on each grid edge with a sign change
  - Connect vertices by lines



Slide Credit: Tao Ju

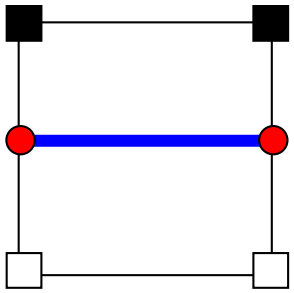
Marching Squares (2D)

- 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.



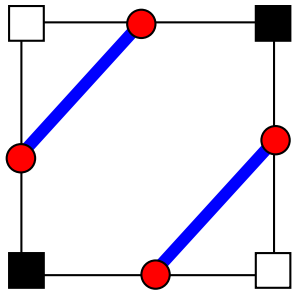
Key: 0 0 0 1

Data: {{2,4}}



Key: 0 0 1 1

Data: {{3,4}}



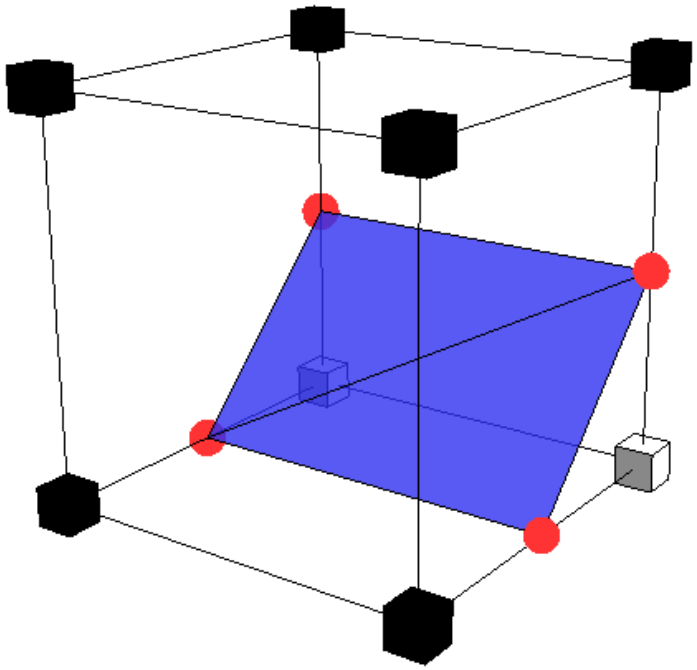
Key: 1 0 0 1

Data: {{1,3}, {2,4}}

Slide Credit: Tao Ju

Marching Cubes (3D)

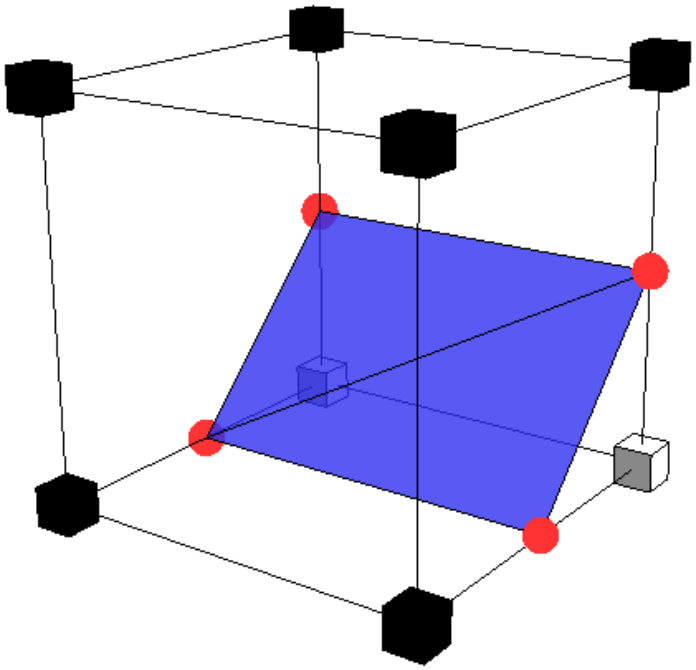
- For each grid cell with a sign change
  - Create one vertex on each grid edge with a sign change (using linear interpolation)
  - Connect vertices into triangles



Slide Credit: Tao Ju

Marching Cubes (3D)

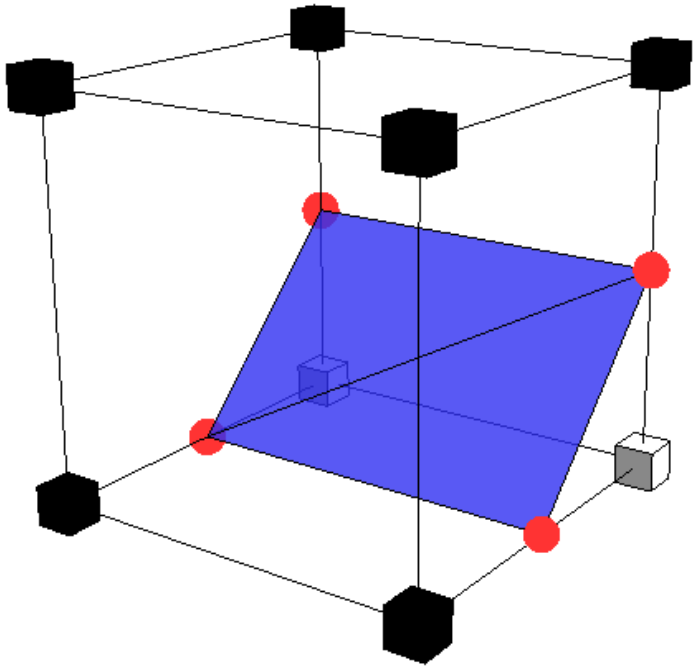
- For each grid cell with a sign change
  - Create one vertex on each grid edge with a sign change (using linear interpolation)
  - Connect vertices into triangles



Slide Credit: Tao Ju

Marching Cubes (3D)

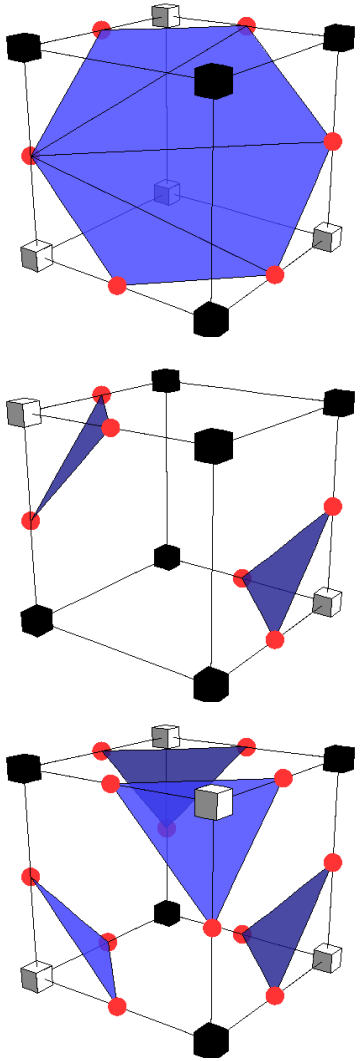
- Connecting vertices by triangles
  - Triangles shouldn't intersect
  - To be a closed manifold:
    - Each vertex used by a triangle "fan"
    - Each mesh edge used by 2 triangles (if inside grid cell) or 1 triangle (if on a grid face)



Slide Credit: Tao Ju

Marching Cubes (3D)

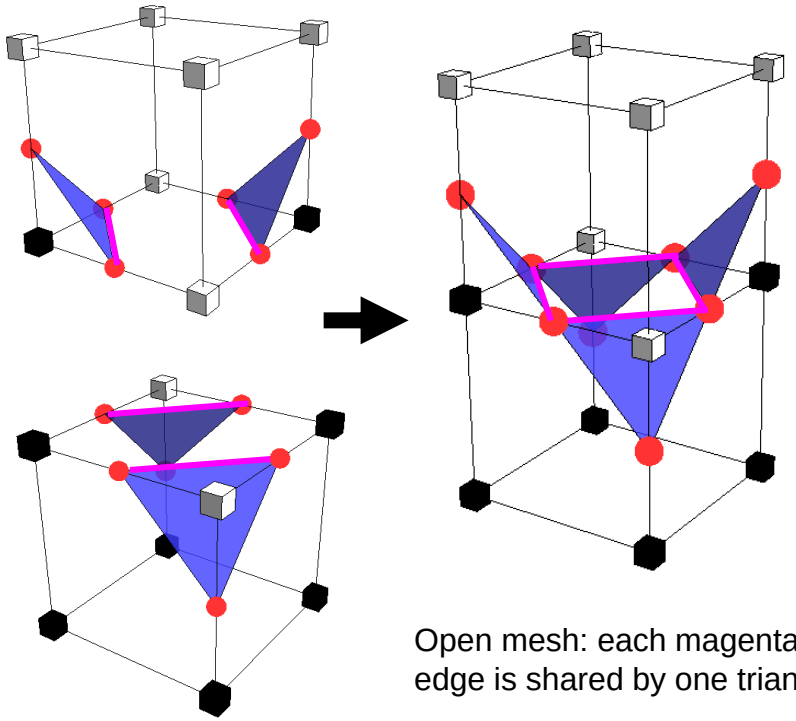
- Connecting vertices by triangles
  - Triangles shouldn't intersect
  - To be a closed manifold:
    - Each vertex used by a triangle "fan"
    - Each mesh edge used by 2 triangles (if inside grid cell) or 1 triangle (if on a grid face)



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)

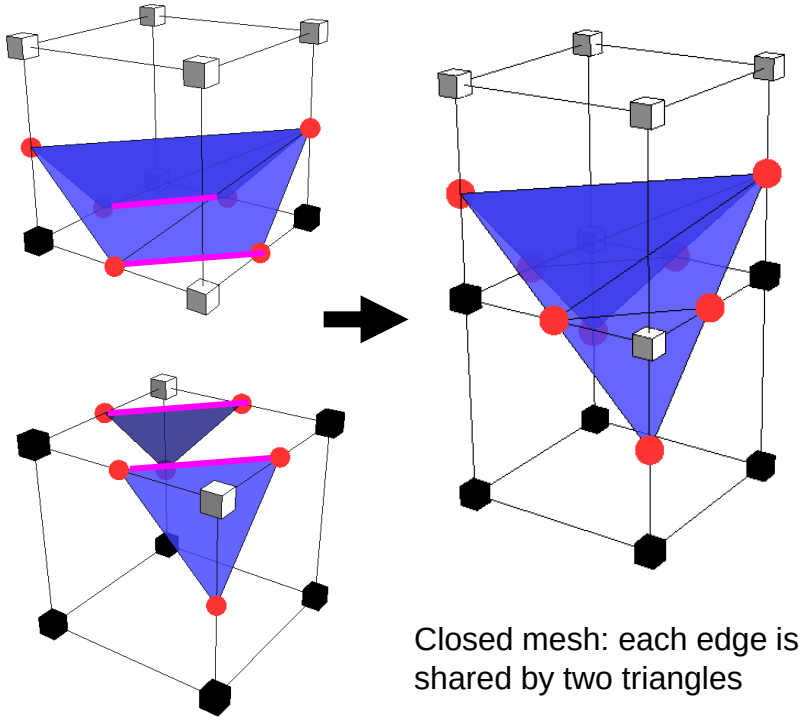


Open mesh: each magenta edge is shared by one triangle

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

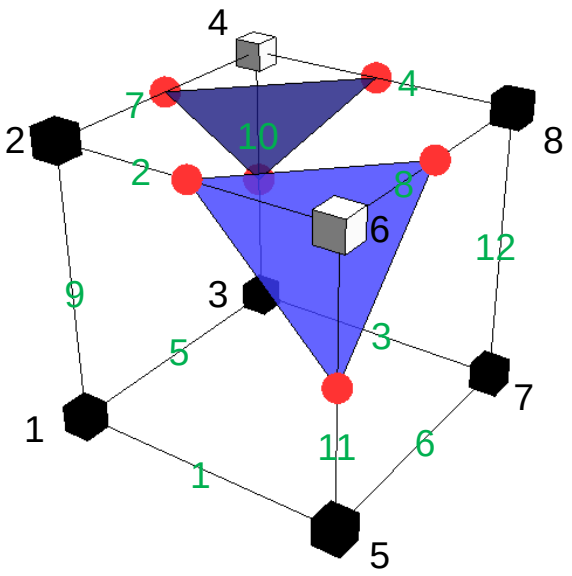


Closed mesh: each edge is shared by two triangles

Slide Credit: Tao Ju

Marching Cubes (3D)

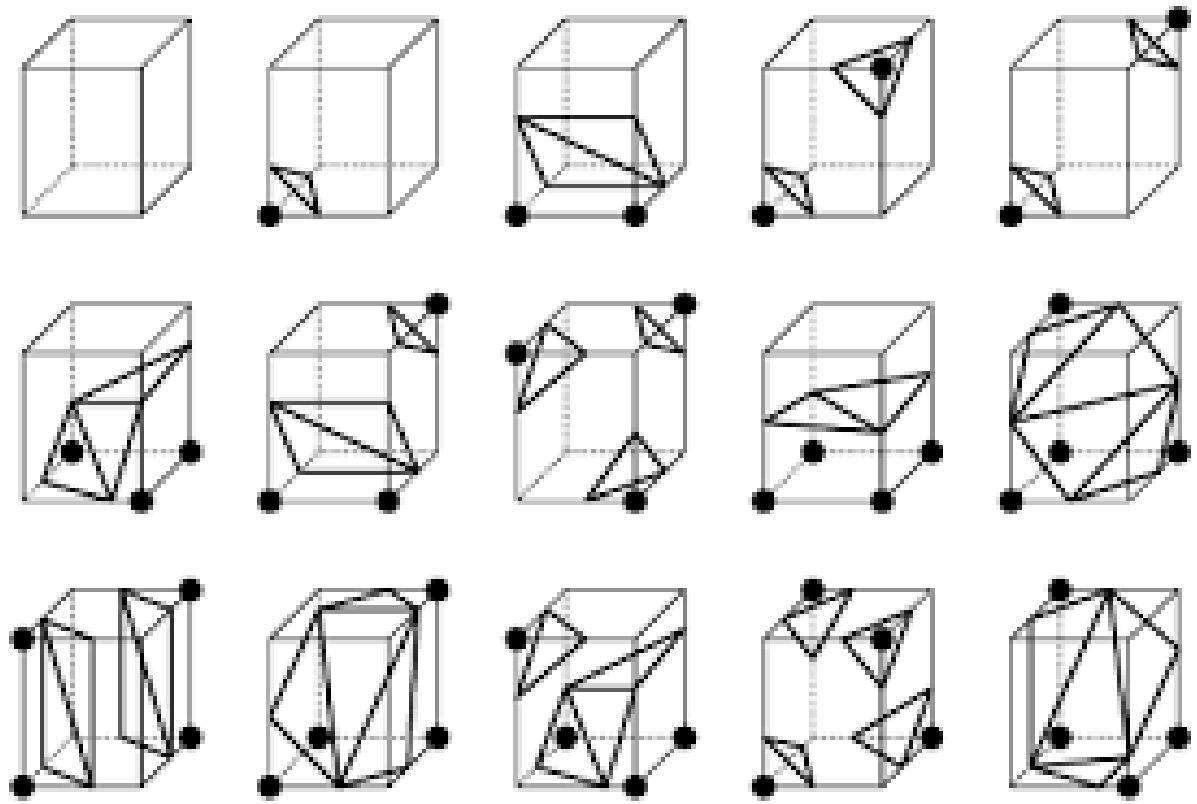
- Connecting vertices by triangles
  - Triangles shouldn't intersect
  - To be a closed manifold:
    - Each vertex used by a triangle "fan"
    - Each mesh edge used by 2 triangles (if inside grid cell) or 1 triangle (if on a grid face)
    - Each mesh edge on the grid face is shared between adjacent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

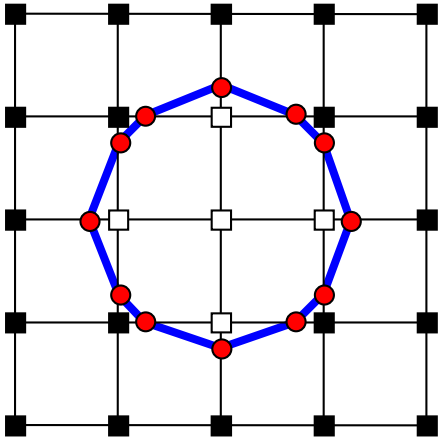
Lookup Table



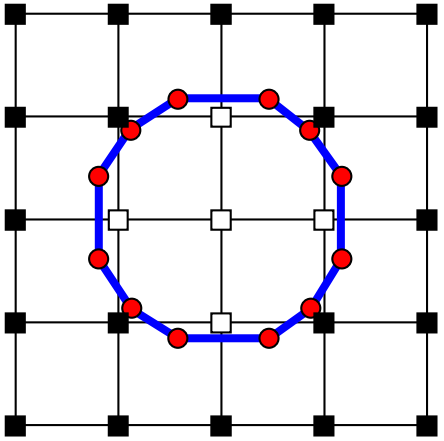
Slide Credit: Tao Ju

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

Discussion