Surface Triangulation and Voronoi Regions

CS334
Spring 2025

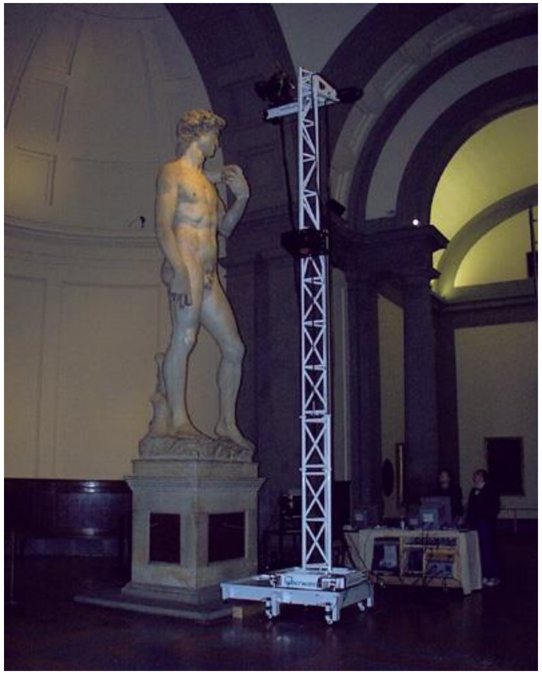
Daniel G. Aliaga
Department of Computer Science
Purdue University

[Slides with help from Michael Kazhdan @ JHU,
Ioannis Stamos @ CUNY, and
Profs. Shmuel Wimer and Andy Mirzaian]

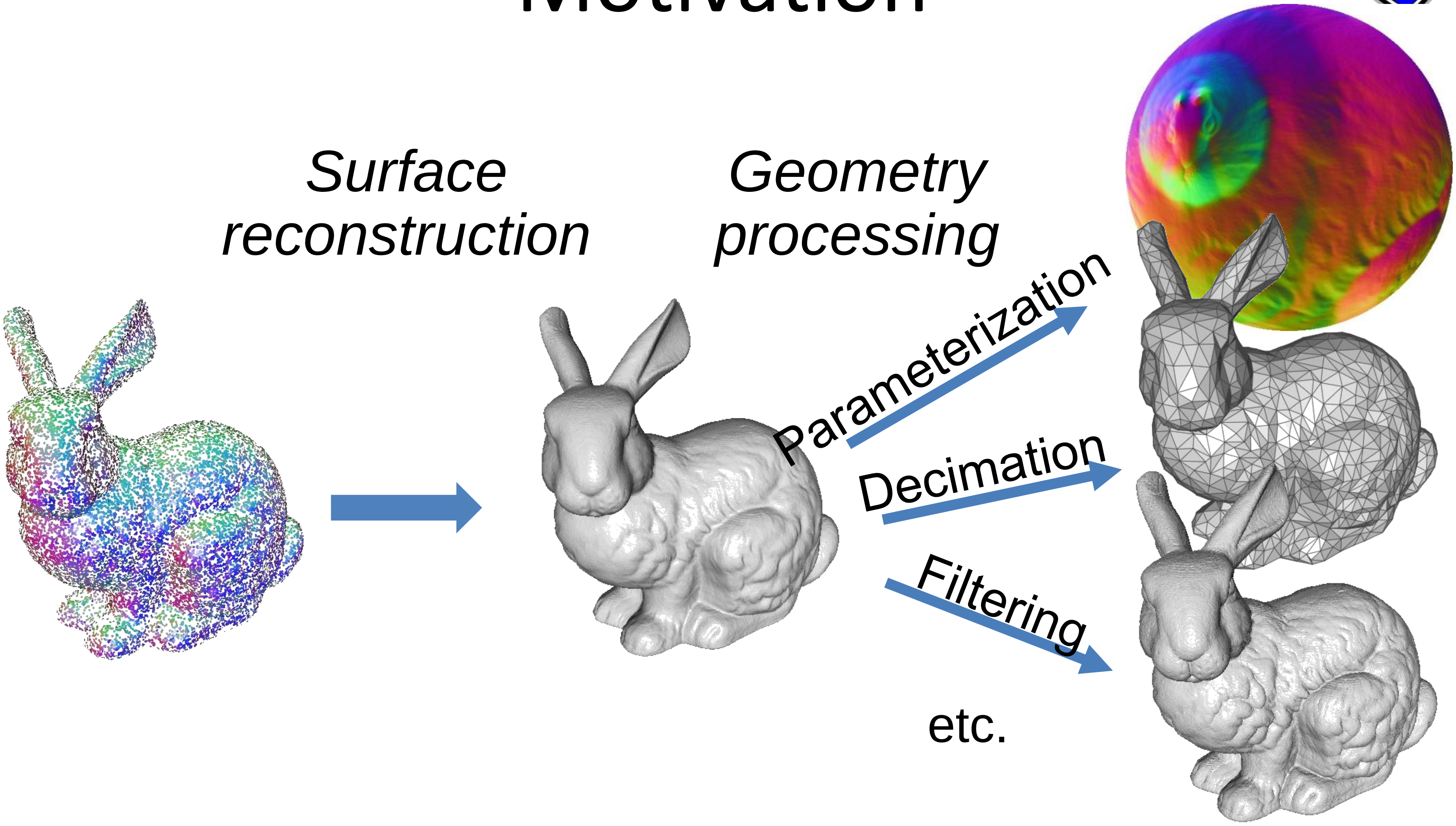
Motivation

- Time of flight
- Structured light
- Stereo images
- Shape from shading
- Etc.

http://graphics.stanford.edu/projects/mich/



Motivation

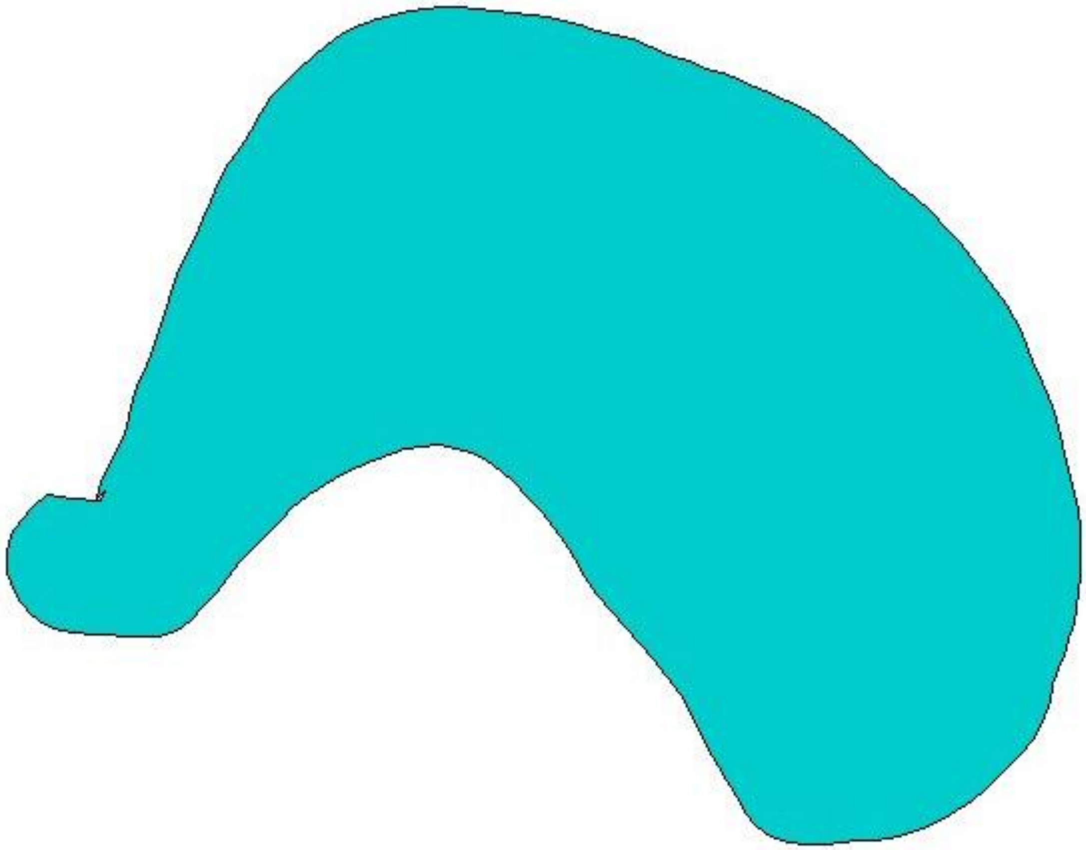


Surface reconstruction
Geometry processing
Parameterization
Decimation
Filtering
etc.

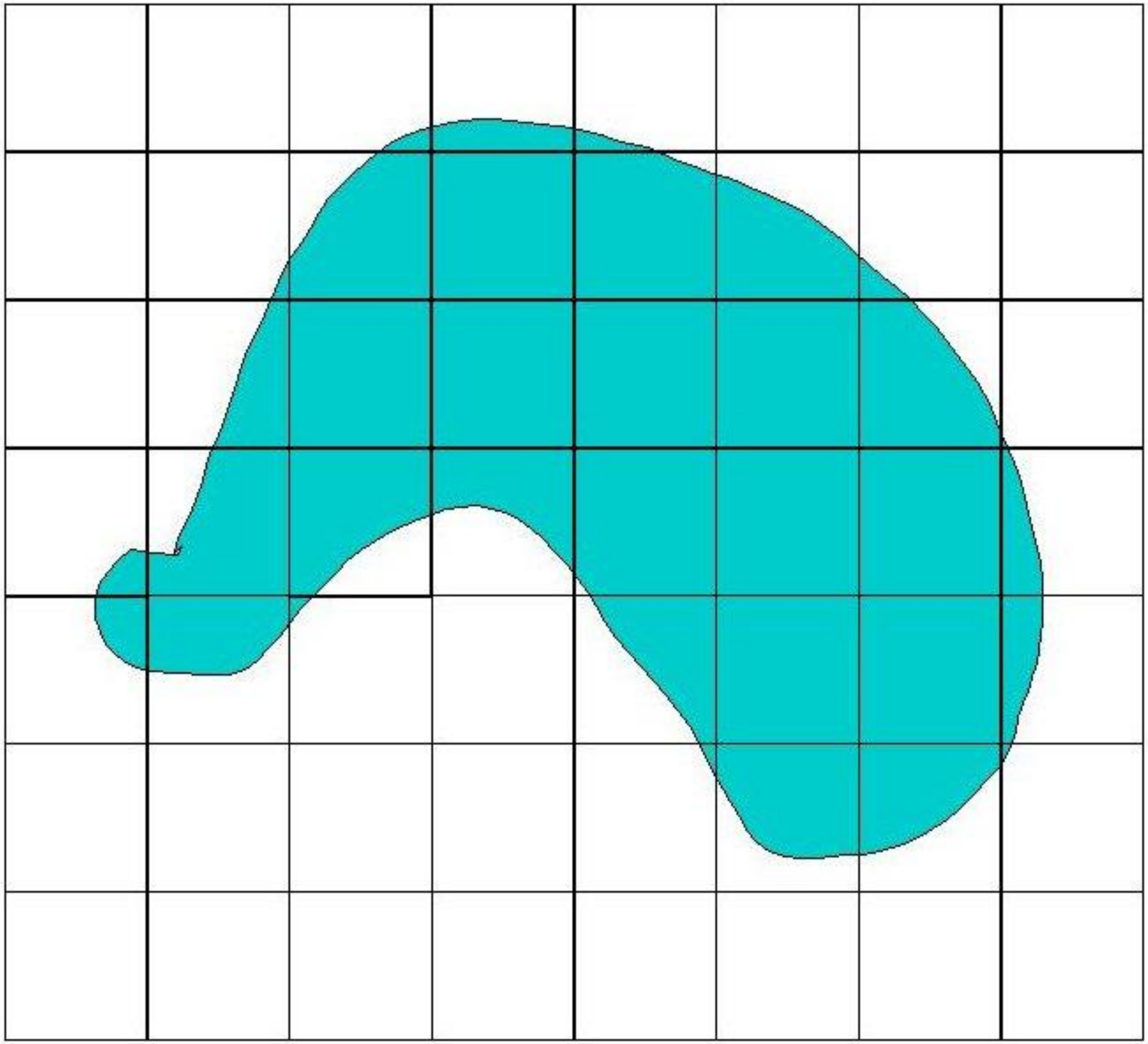
Marching Squares (2D)
Marching Cubes (3D)

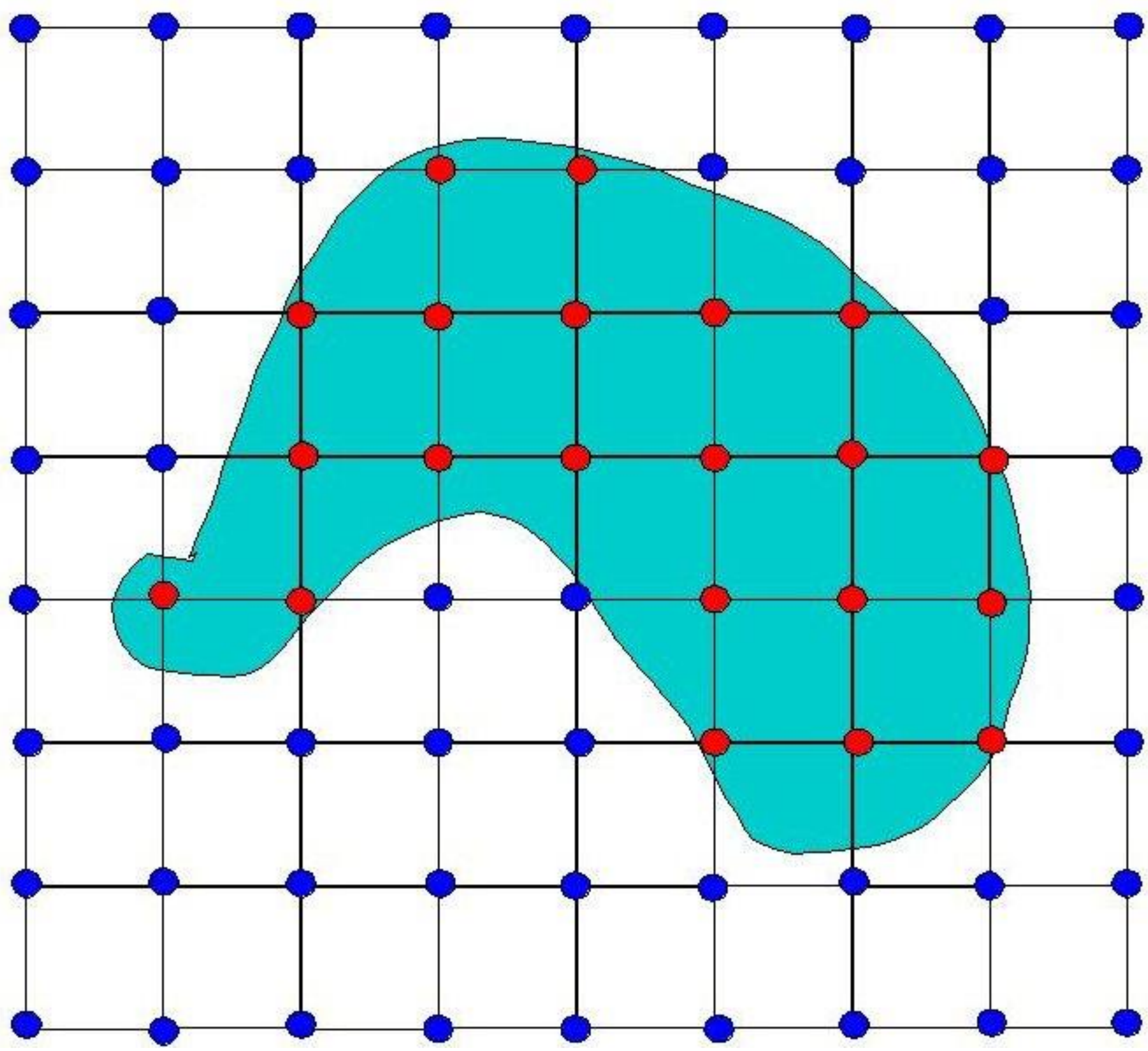
- The premise of the algorithm is to divide the input volume into a discrete set of squares/cubes. By assuming linear reconstruction filtering, each square/cube, which contains a piece of a given isosurface, can be easily represented with lines/triangles.

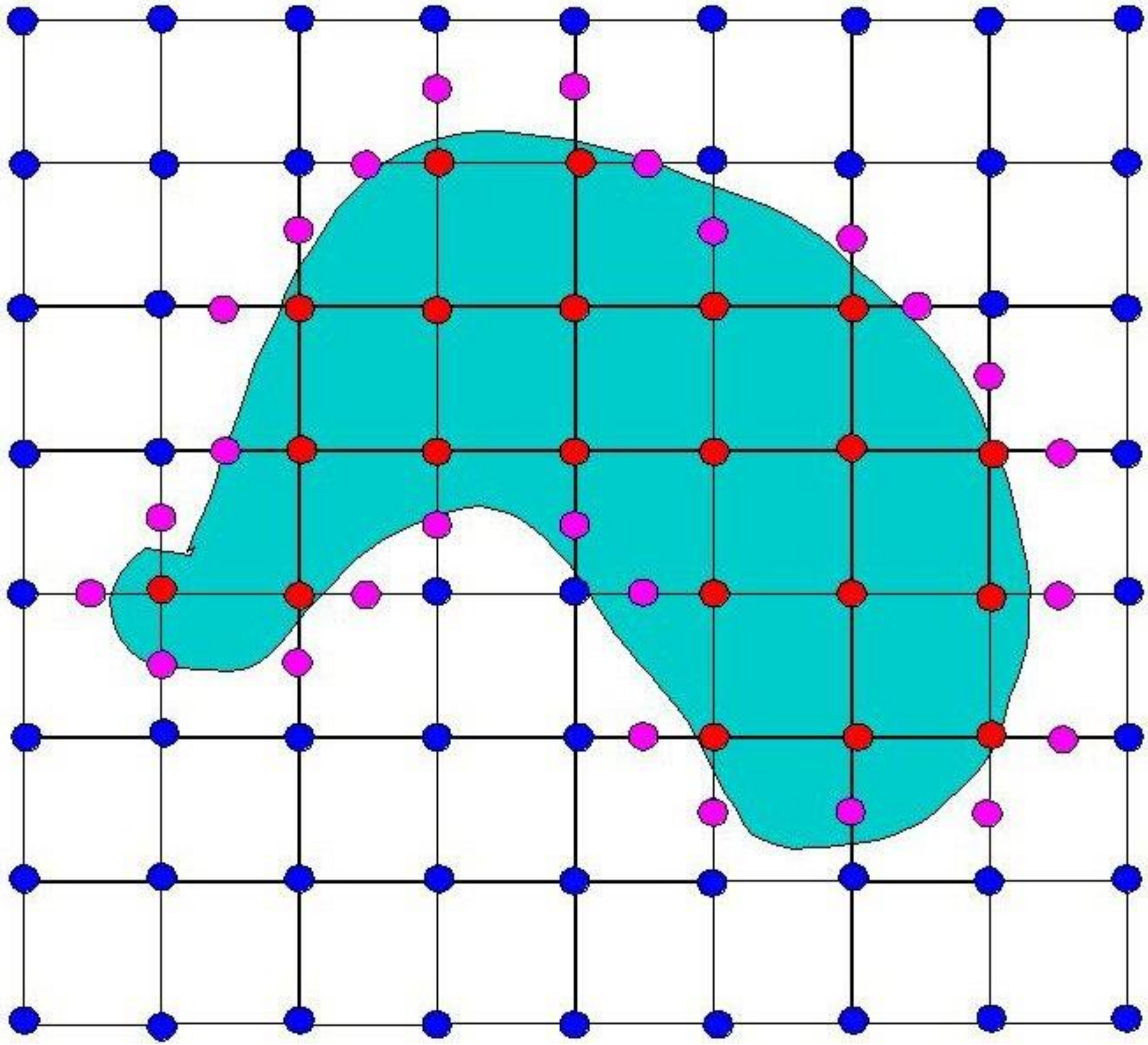
Marching Squares

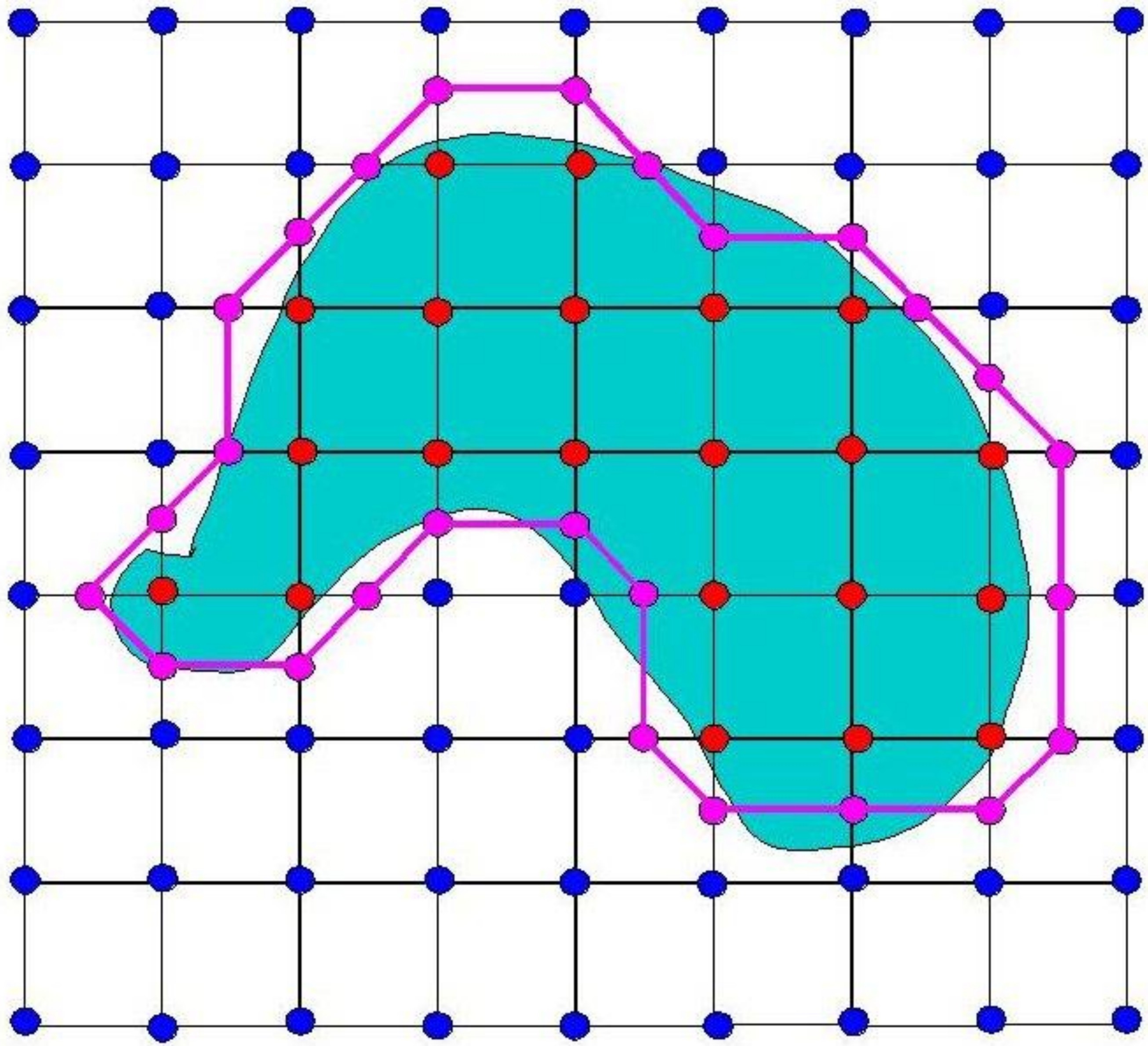


[https://www.cs.carleton.edu/cs_comps/0405/shape/marching_cubes.html]



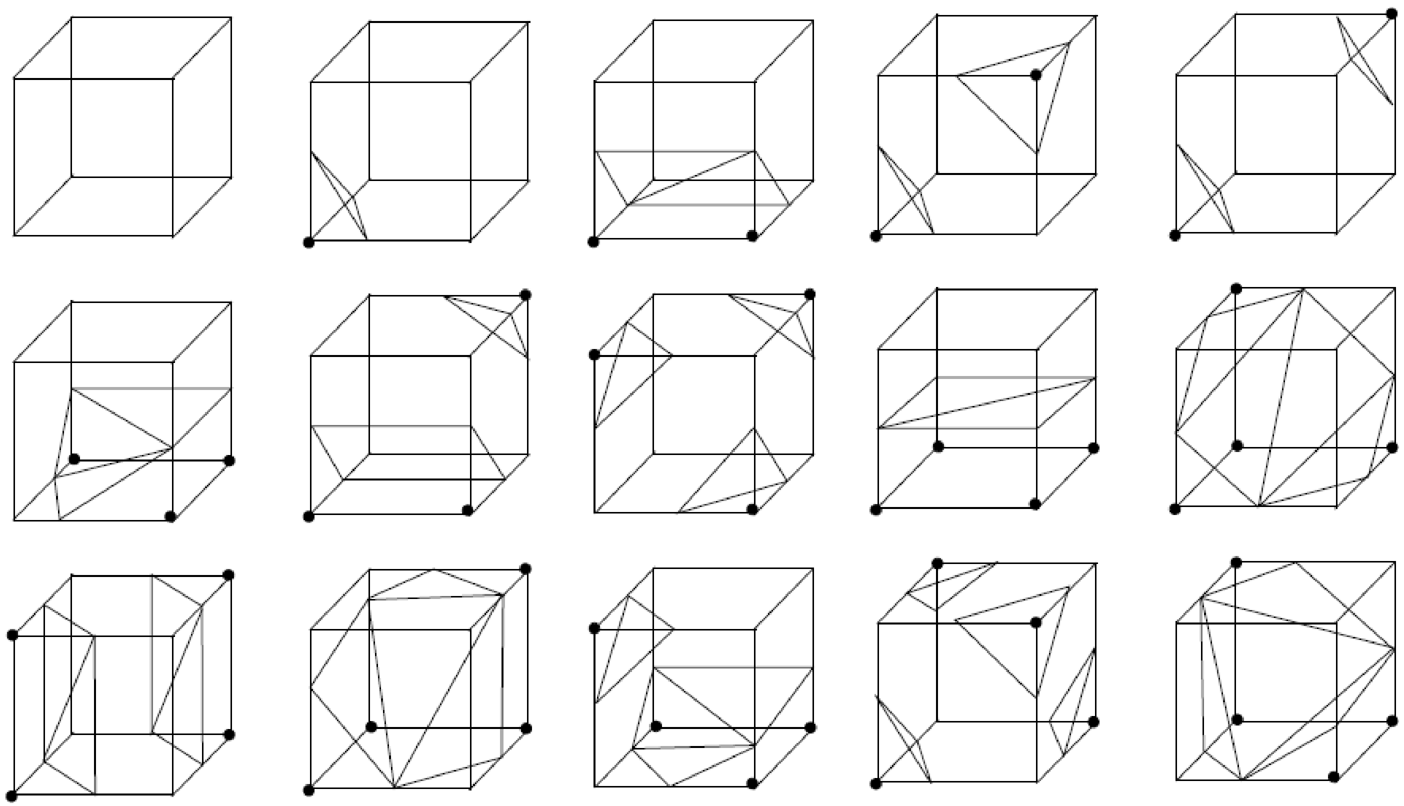






Marching Cubes

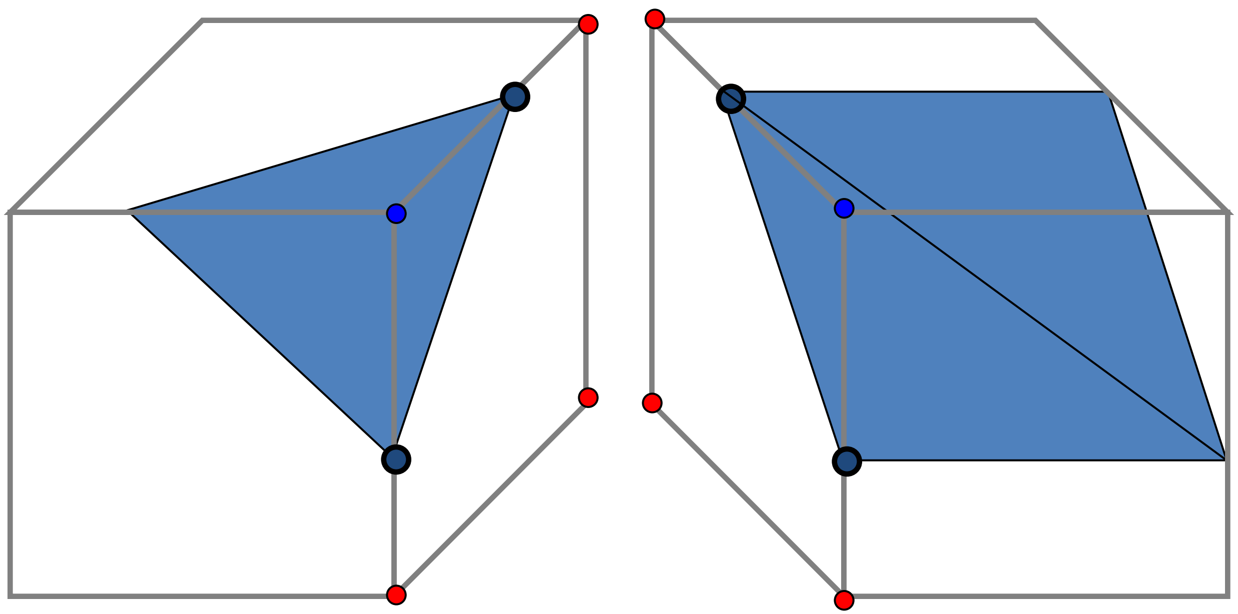
If the function is sampled on a regular voxel grid, we can independently triangulate each voxel.



Marching Cubes

Iso-vertices on an edge are only determined by the values on the corner of the edge:

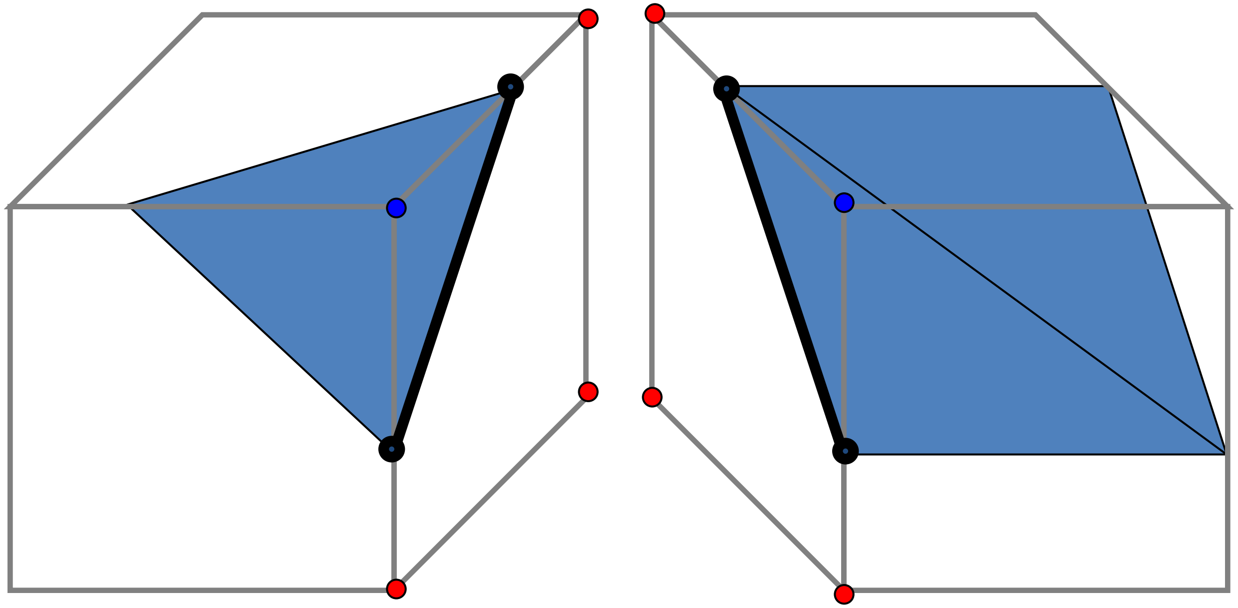
$\Rightarrow$ Iso-vertices are consistent across voxels.



Marching Cubes

Iso-edges on a face are only determined by the values on the face:

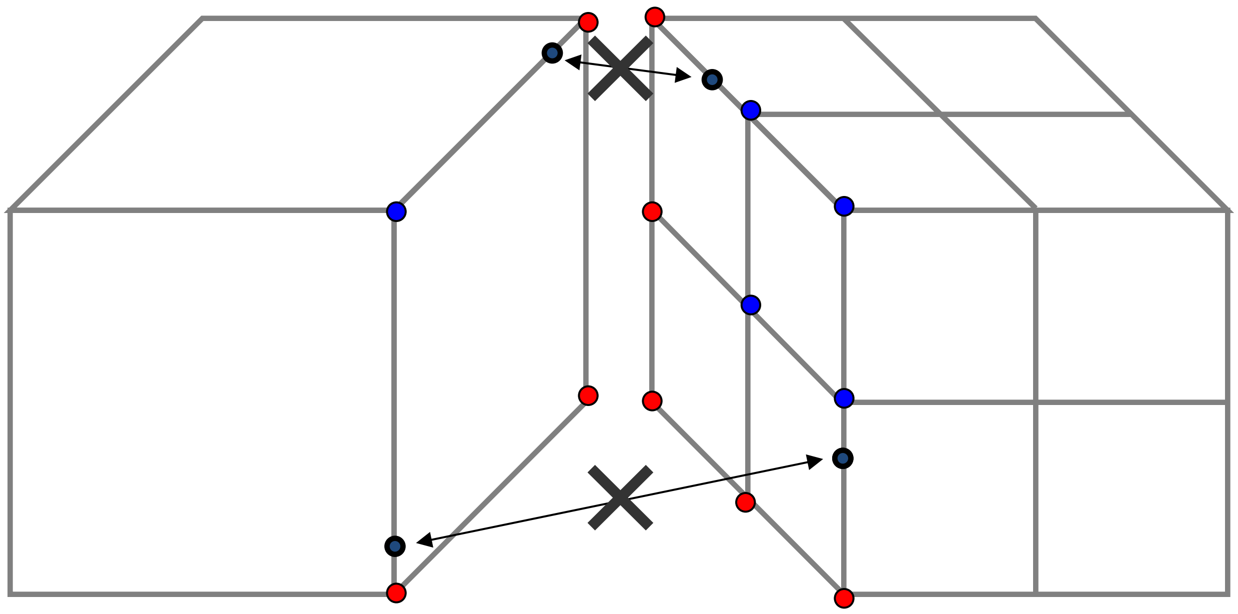
⇒ Each iso-edge is shared by two triangles so the mesh is water-tight.



Challenges

Extracting a surface by independently triangulating the leaf octants, depth-disparities can cause:

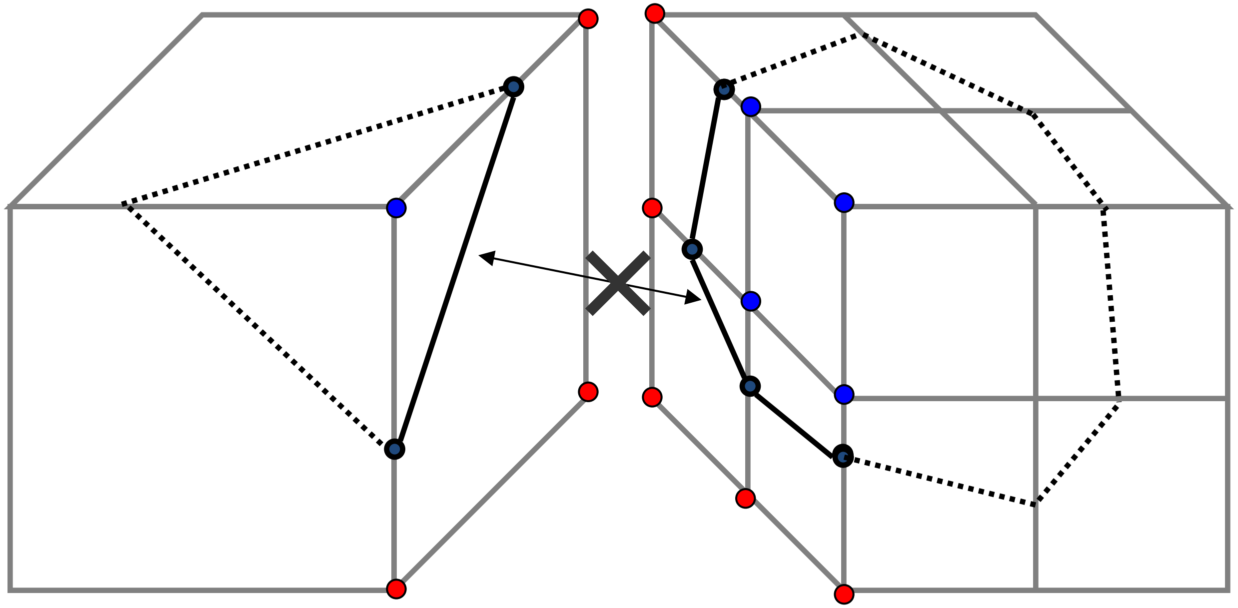
- Inconsistent extrapolation to <u>edges</u>
  - ⇒ Inconsistent iso-vertex positions



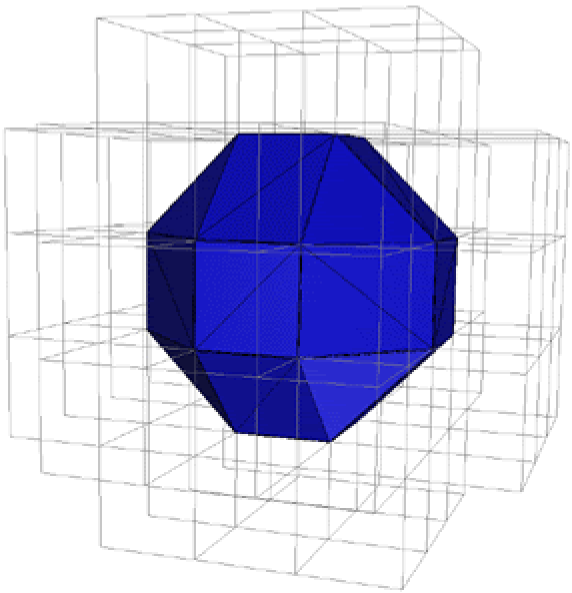
Challenges

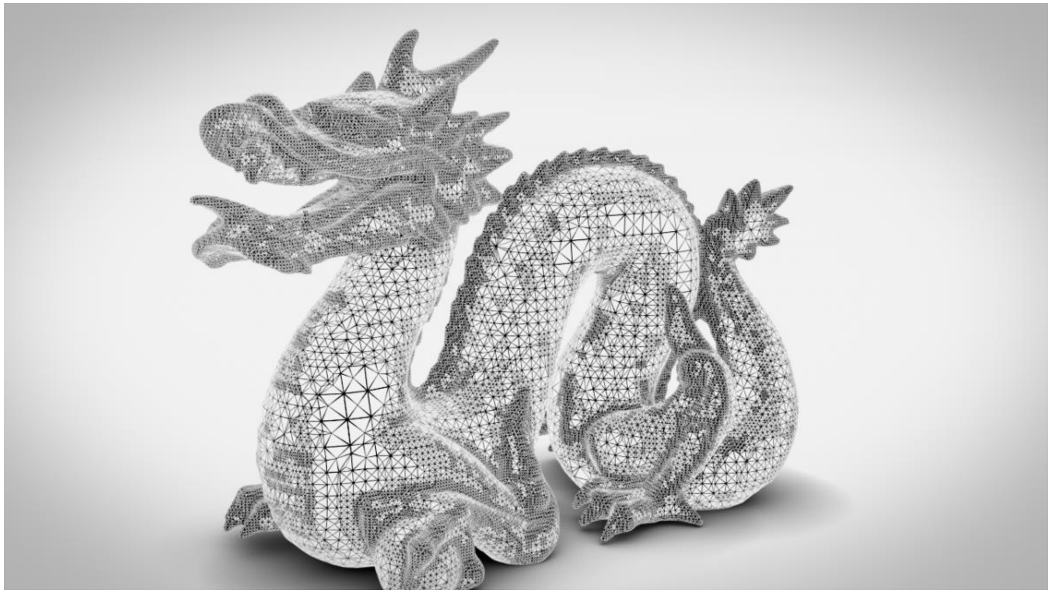
Extracting a surface by independently triangulating the leaf octants, depth-disparities can cause:

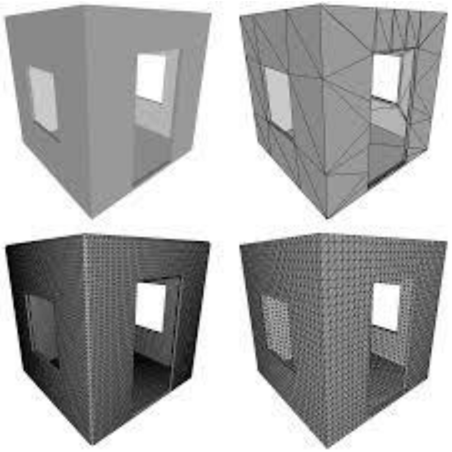
- Inconsistent extrapolation to faces
  - ⇒ Inconsistent iso-edges



Marching Cubes

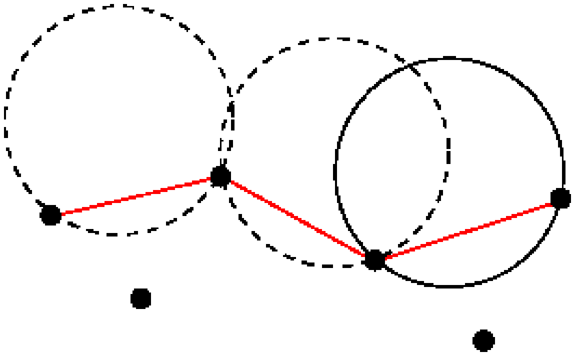






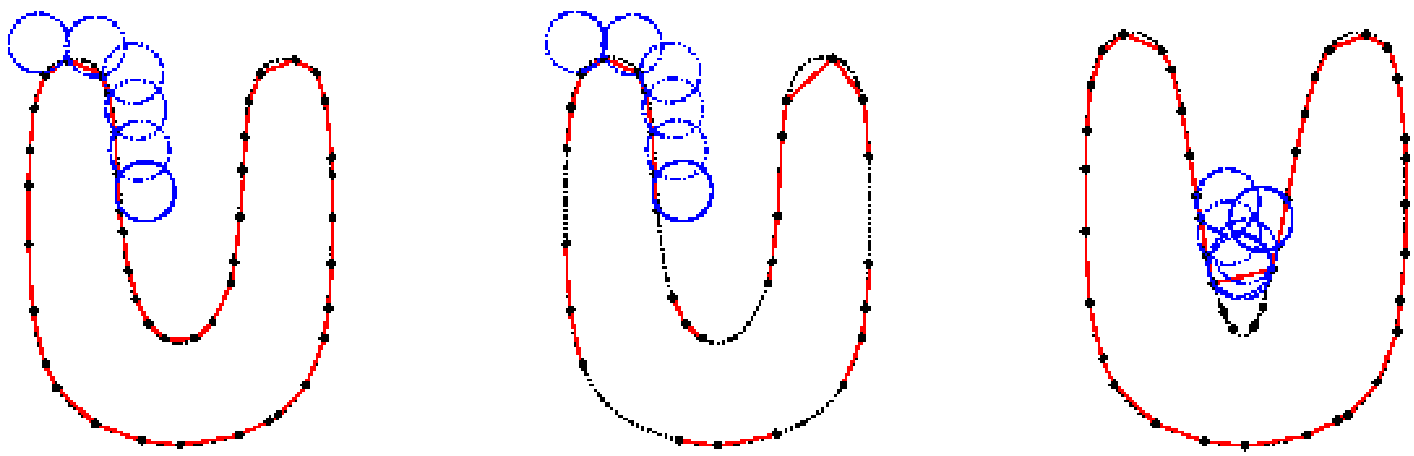
Ball-pivoting

Bernardini et al., IBM



Fixed-radius ball “rolling” over points selects subset of alpha-shape.

Pivoting in 2D



(a) (b) (c)

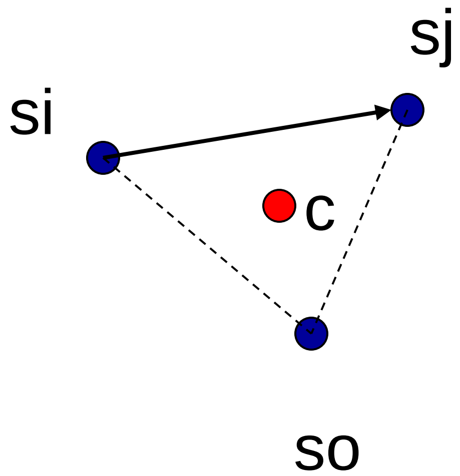
(a) *Circle of radius ρ pivots from point to point, connecting them with edges.*

(b) *When sampling density is low, some of the edges will not be created, leaving holes.*

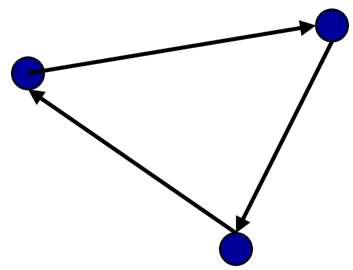
- *(c) When the curvature of the manifold is larger than 1/ρ, some of the points will not be reached by the pivoting ball, and features will be missed.*

The algorithm [Edge representation]

- Edge (si, sj)
  - Opposite point so, center of empty ball c
  - Edge: “Active”, “Boundary”, or “Frozen”



Pivoting example



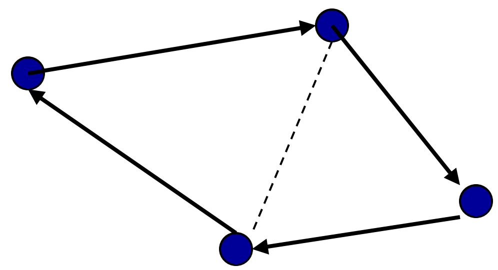
Initial seed triangle:

Empty ball of radius ρ passes through the three points

Active edge

Point on front

Pivoting example

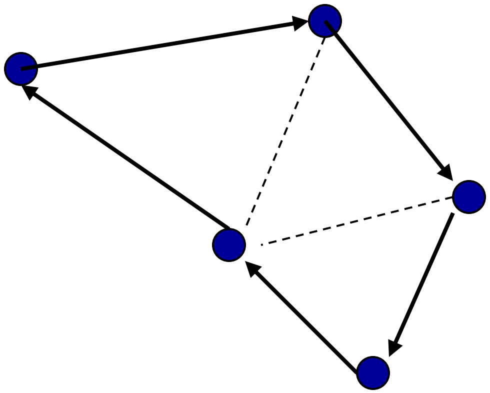


Ball pivoting around active edge

Active edge

Point on front

Pivoting example

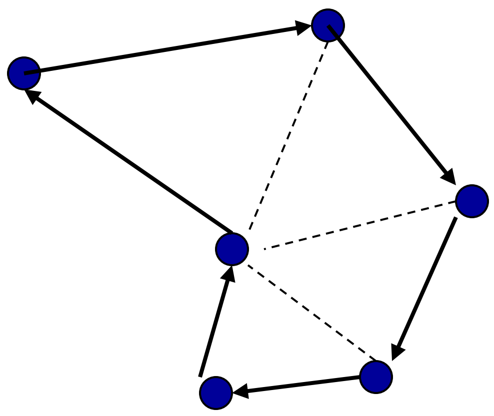


Ball pivoting around active edge

Active edge

Point on front

Pivoting example

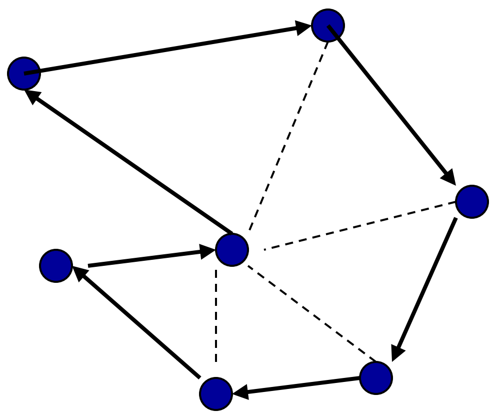


Ball pivoting around active edge

Active edge

Point on front

Pivoting example

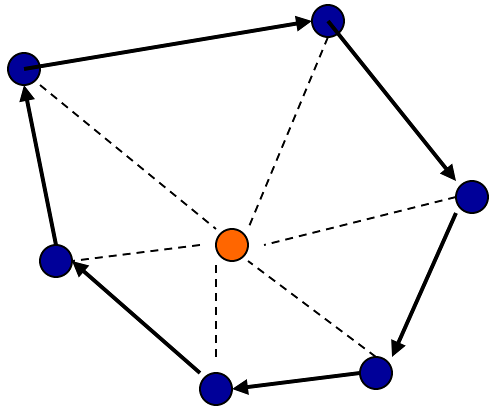


Ball pivoting around active edge

Active edge

● Point on front

Pivoting example



Ball pivoting around active edge

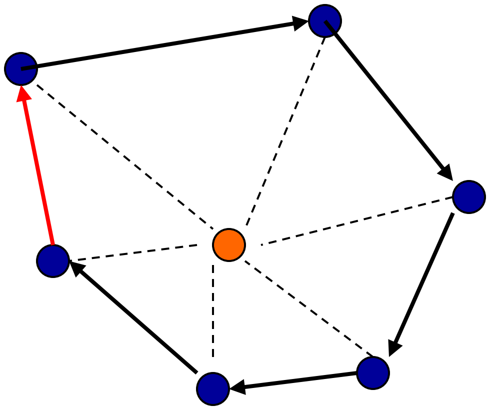
Active edge

Point on front

Internal point

Pivoting example

Boundary edge



Ball pivoting around active edge

No pivot found

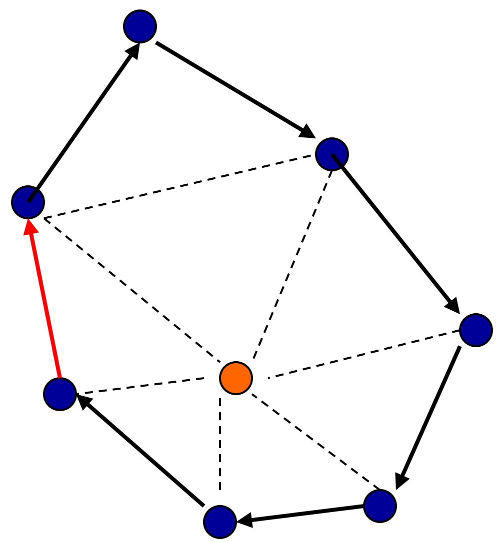
Active edge

Point on front

Internal point

Pivoting example

Boundary edge



Active edge

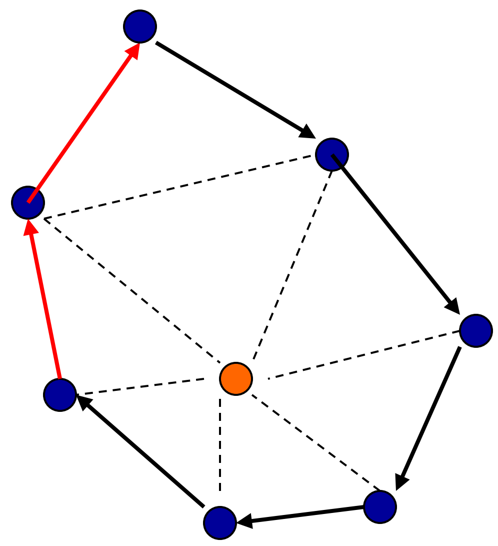
Ball pivoting around active edge

Point on front

Internal point

Pivoting example

Boundary edge



Active edge

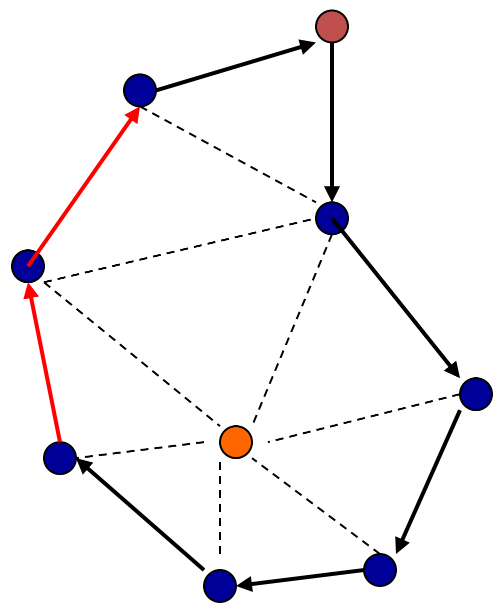
- Point on front
- Internal point

Ball pivoting around active edge

No pivot found

Pivoting example

Boundary edge

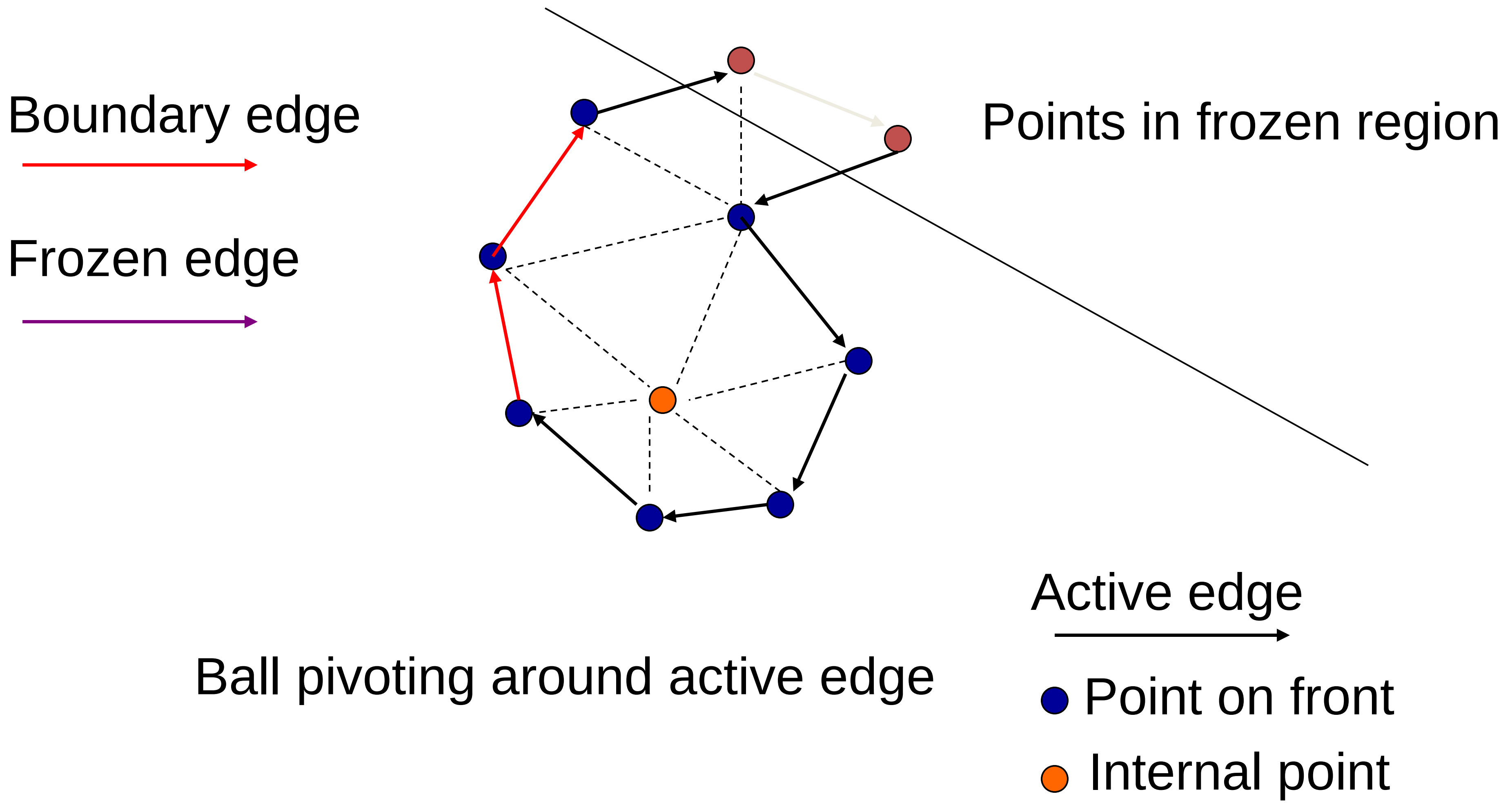


Active edge

Ball pivoting around active edge

- Point on front
- Internal point

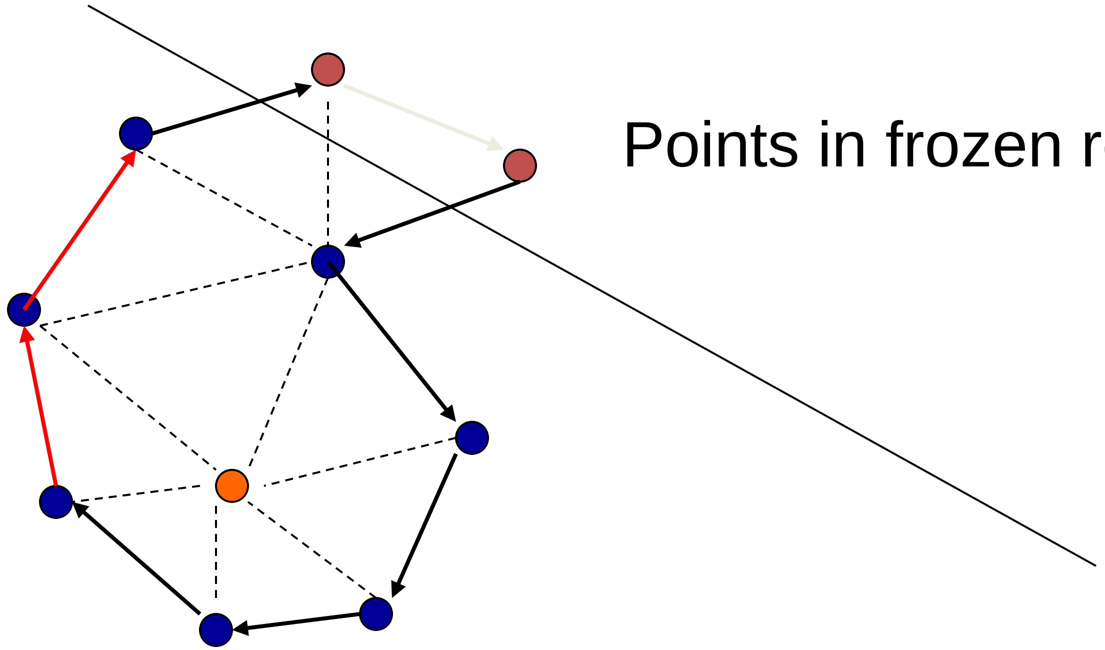
Pivoting example



Pivoting example

Boundary edge

Frozen edge



Points in frozen region

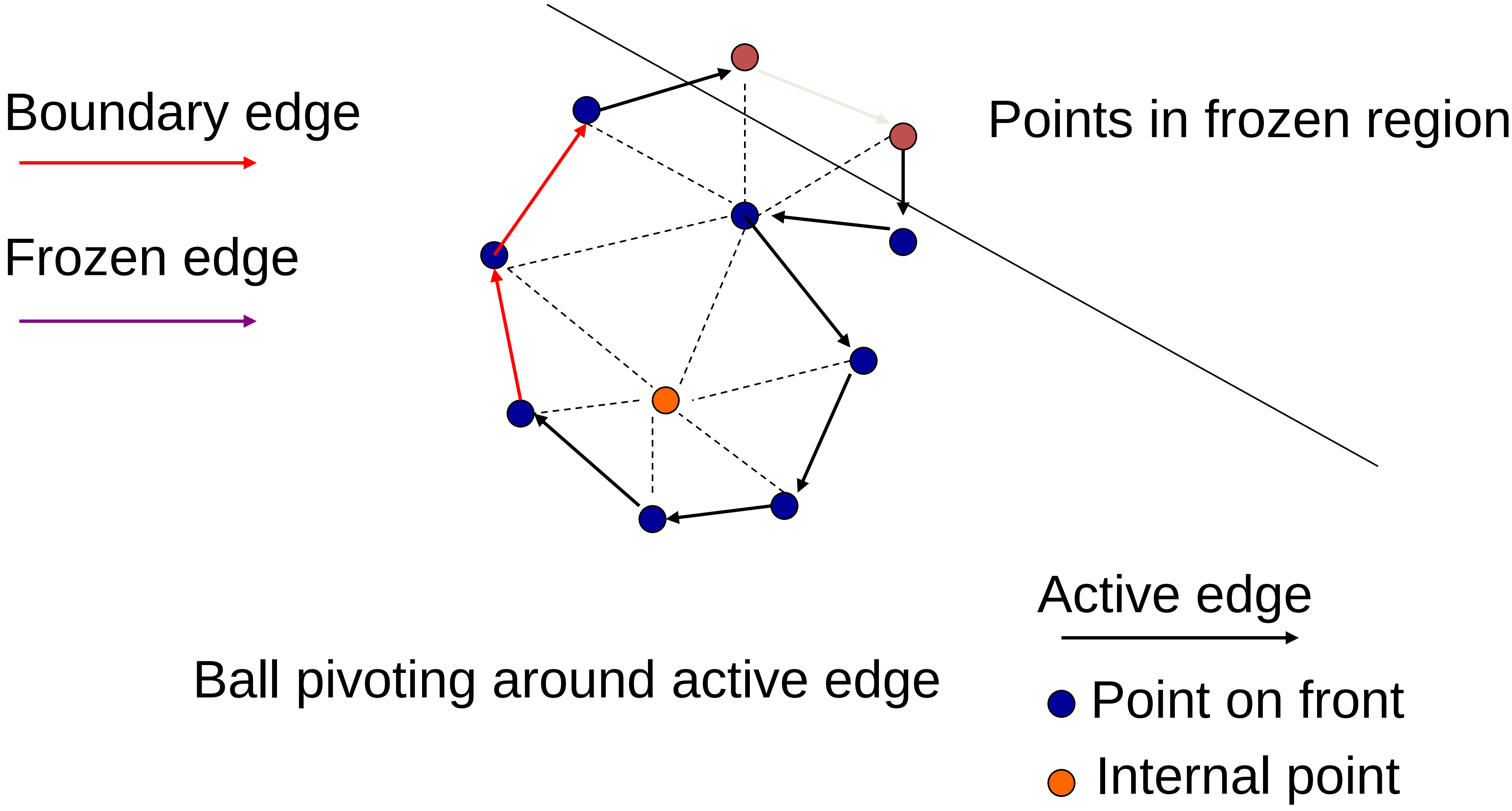
Ball pivoting around active edge

Active edge

Point on front

Internal point

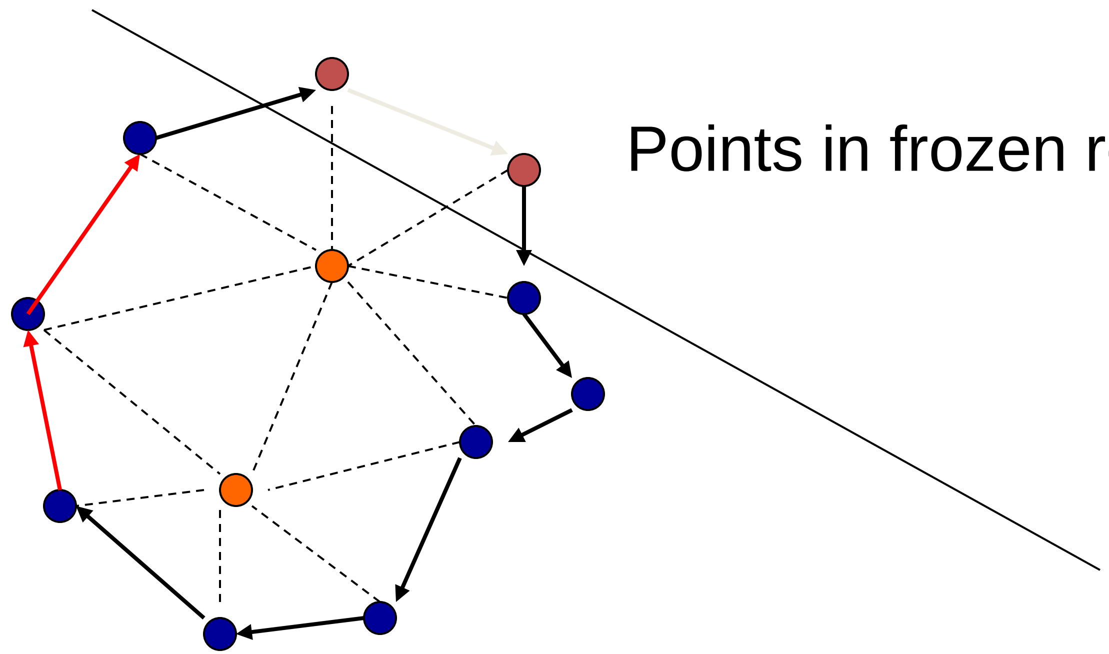
Pivoting example



Pivoting example

Boundary edge

Frozen edge



Points in frozen region

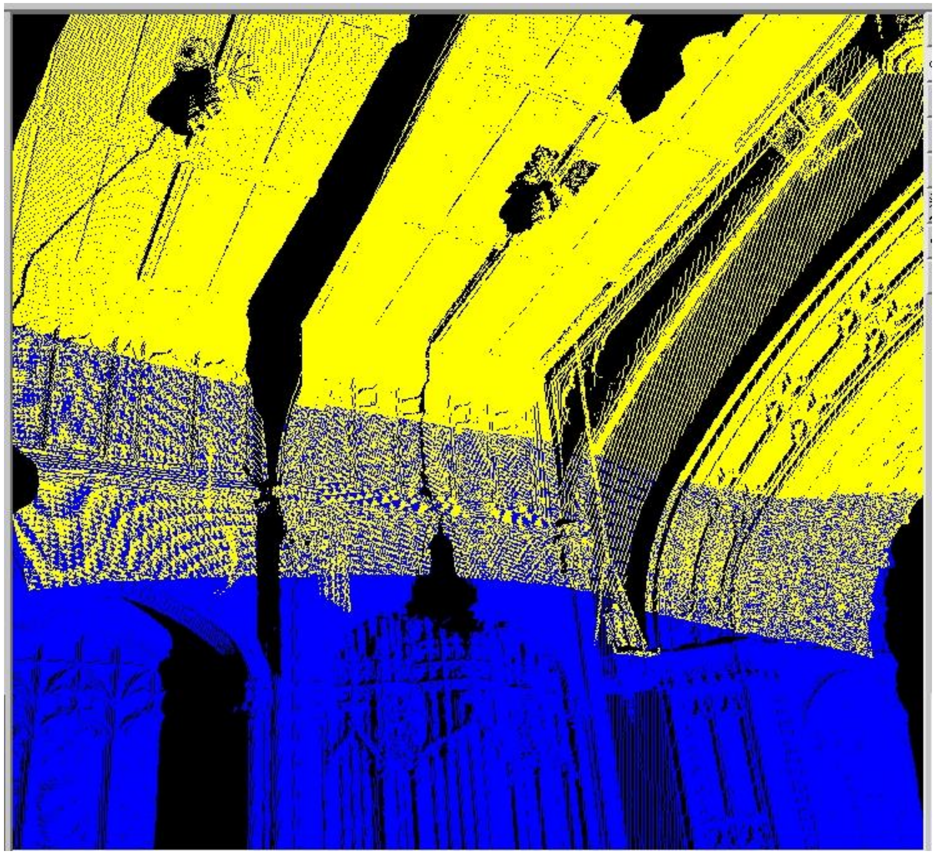
Active edge

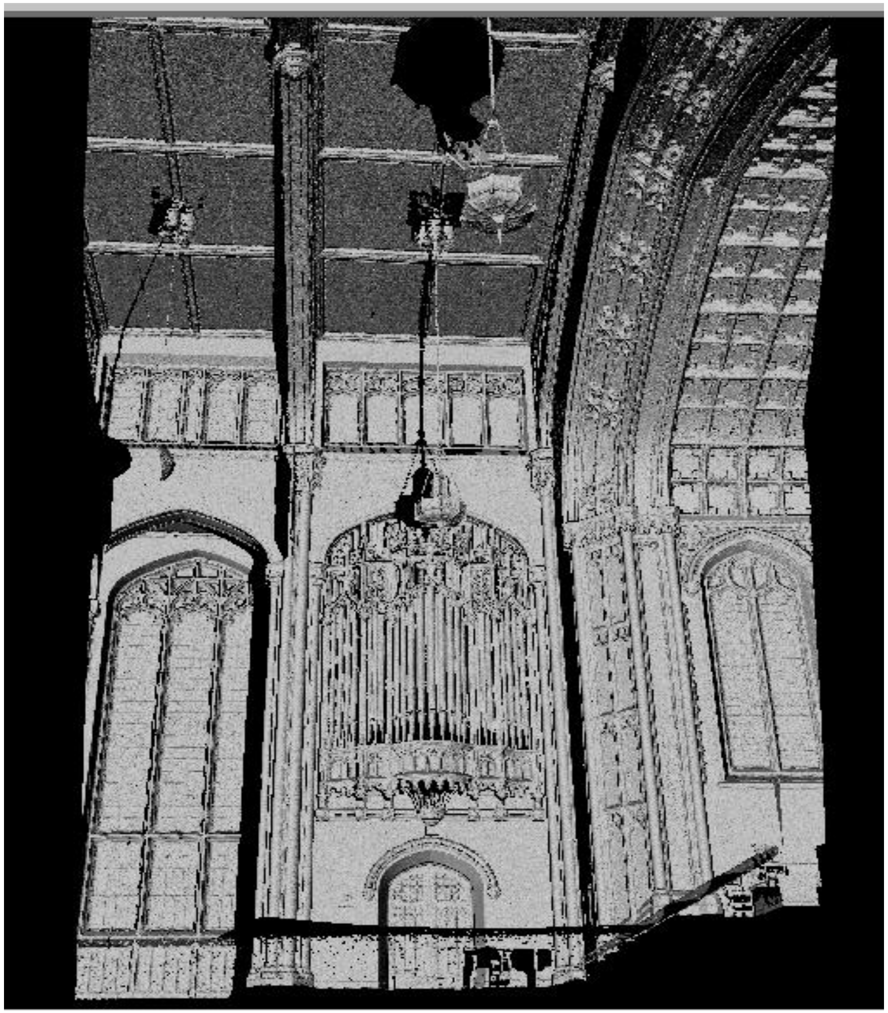
Ball pivoting around active edge

Point on front

Internal point

Ball Pivoting Algorithm



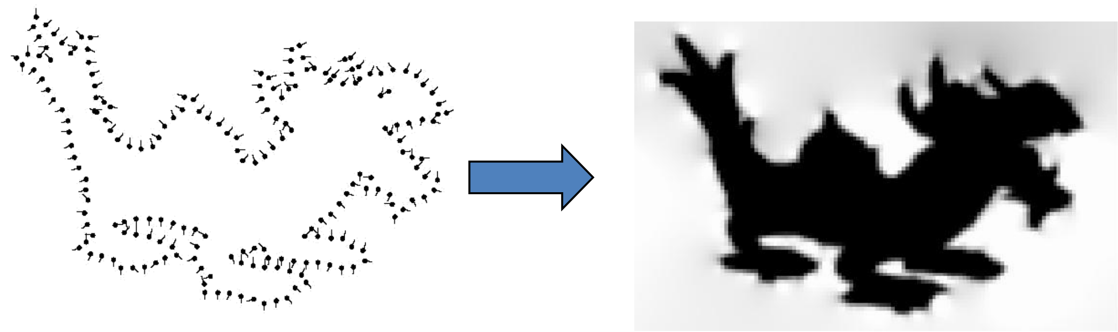


Ball Pivoting Algorithm

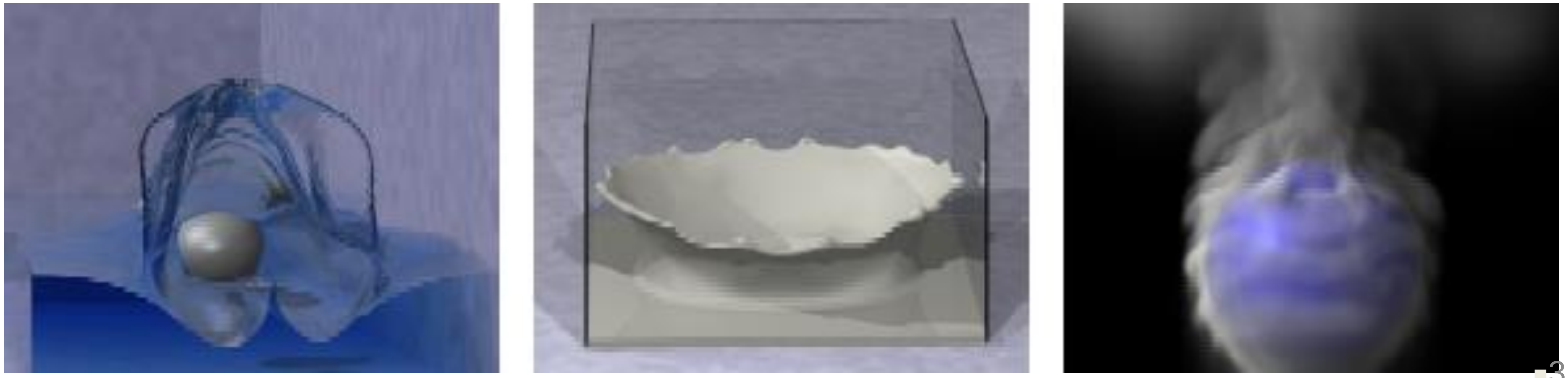
Implicit Representation

Another option is representing a 3D model by an implicit function for:

- Reconstruction
- Fluid Dynamics
- 3D Texturing



Kazhdan 2005



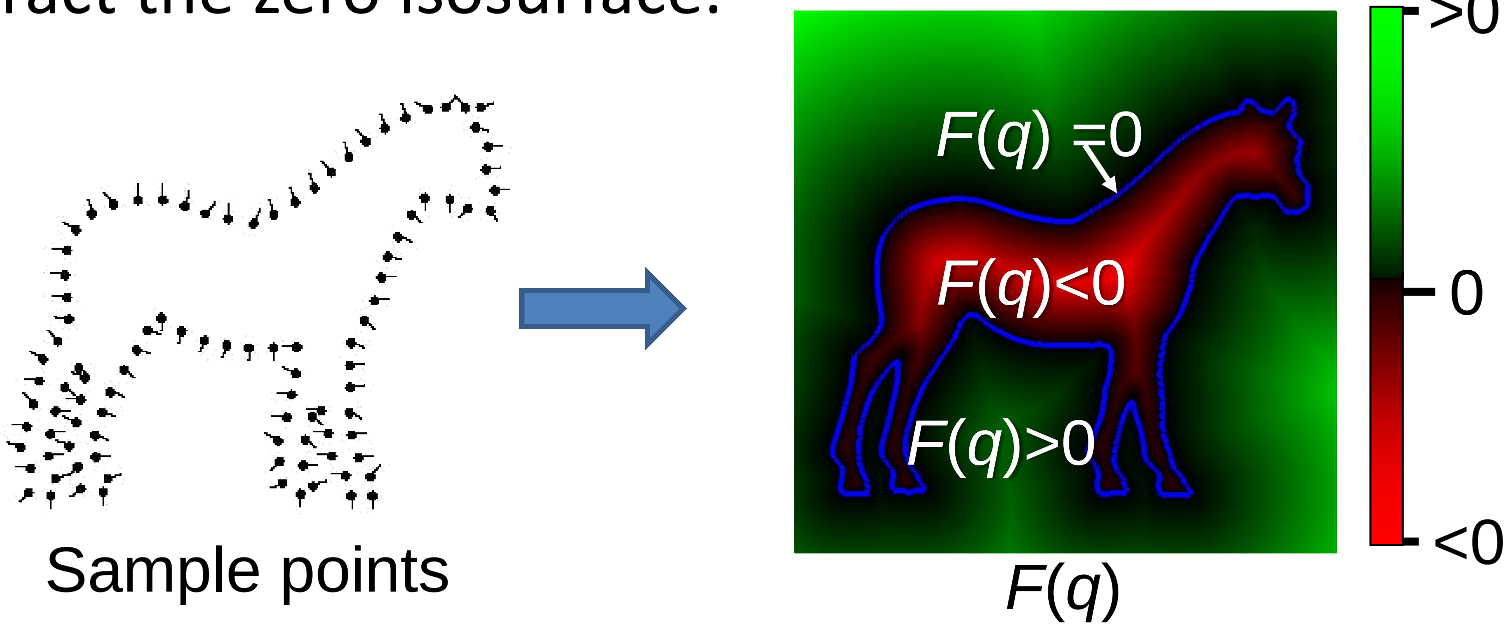
Losasso *et al.* 2004

Implicit Function Fitting

Given point samples:

– Define a function with value zero at the points.

– Extract the zero isosurface.



Triangulation Complexity (in general)

- Theorem: (Gary et. al. 1978) A simple n-vertex polygon can be triangulated in O(nlogn) time and O(n) storage
- The problem has been studied extensively between 1978 and 1991, when in 1991 Chazelle presented an O(n) time complexity algorithm.

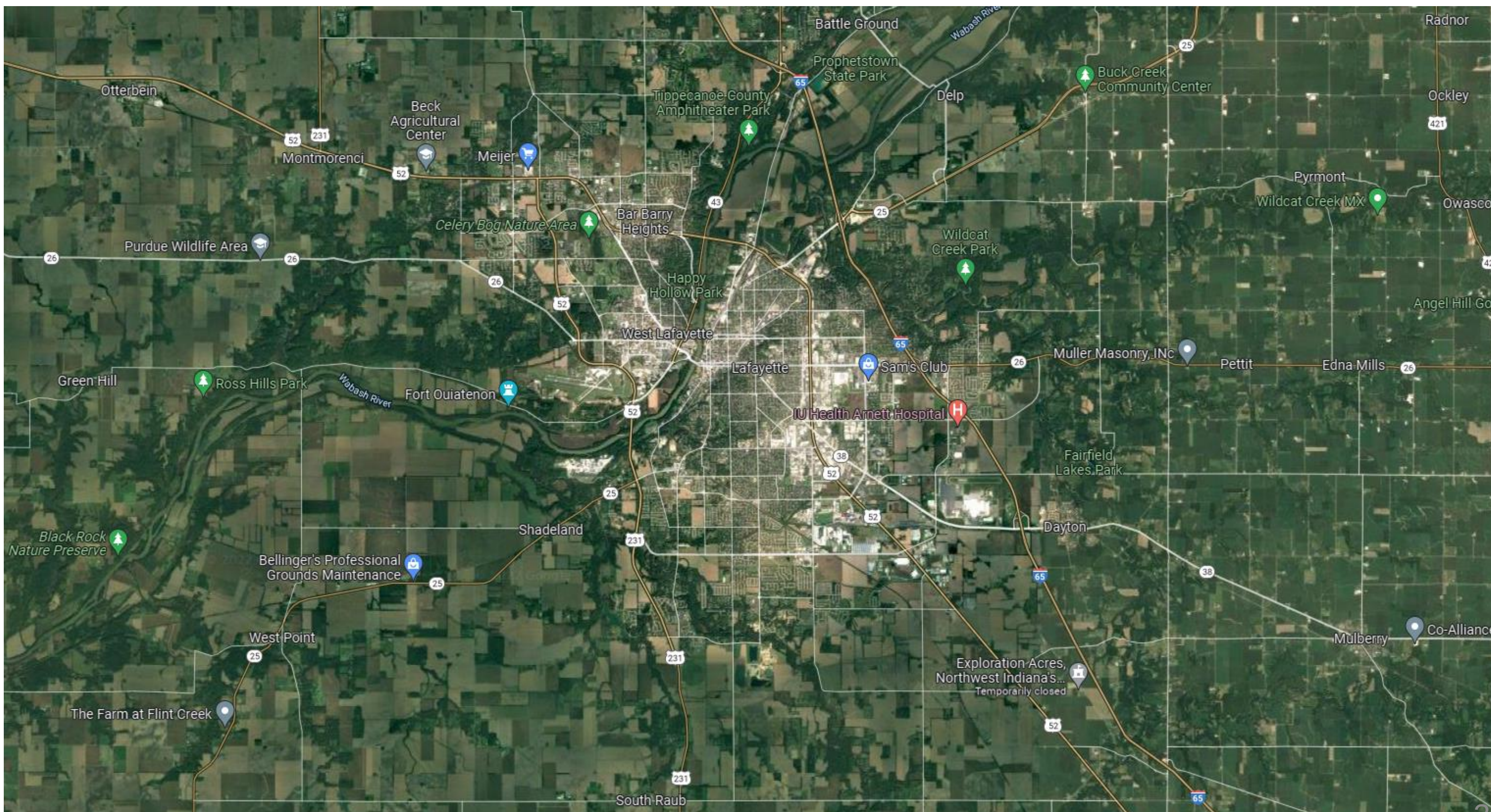
Delaunay Triangulation

- Another very popular algorithm…
- But first, Voronoi Diagrams…

- Relevant Conversation:
  - Captain Kirk: "Spock! Which tricorder tower (i.e., cell phone) should I be using?"
  - Commander Spock: "Logically, the closest one, Jim."

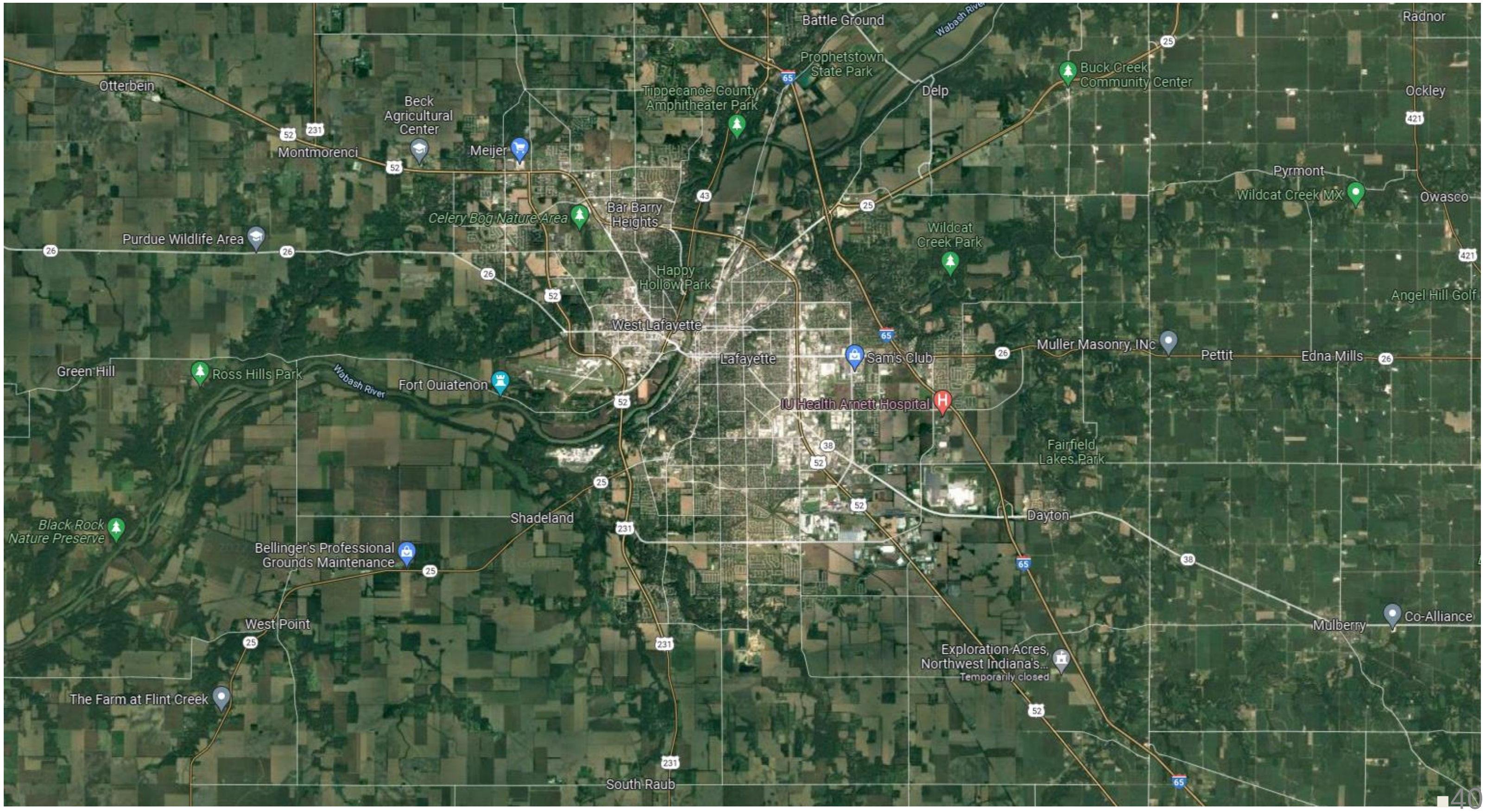
- How do you do that?

Where to place cell phone towers? <u>or</u> Which cell phone tower should I use?



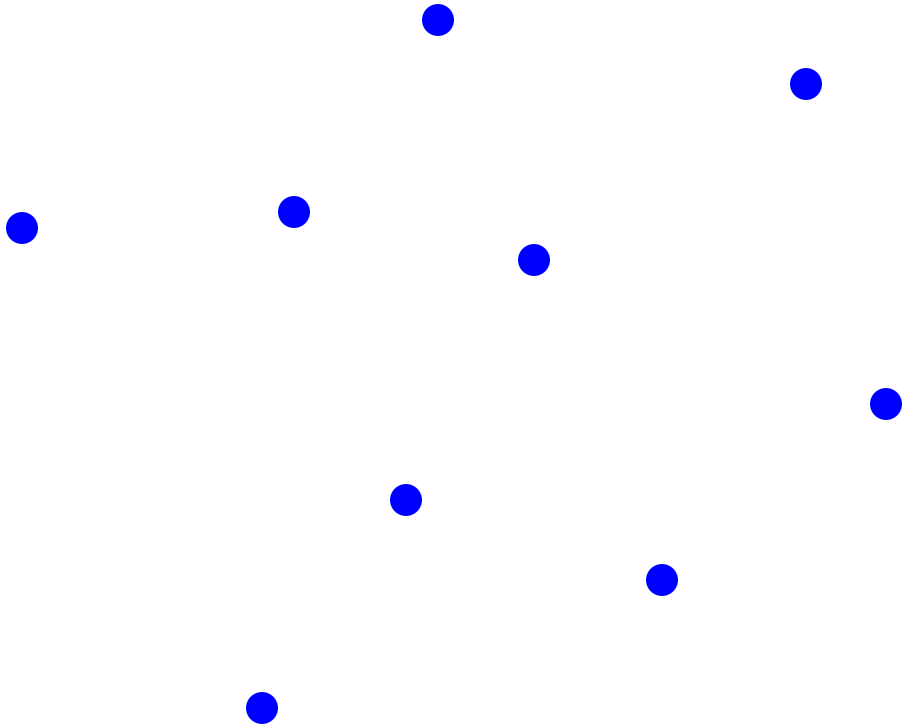
Cell phone towers

P = { p_1, p_2, … , p_n} a set of n points in the plane.

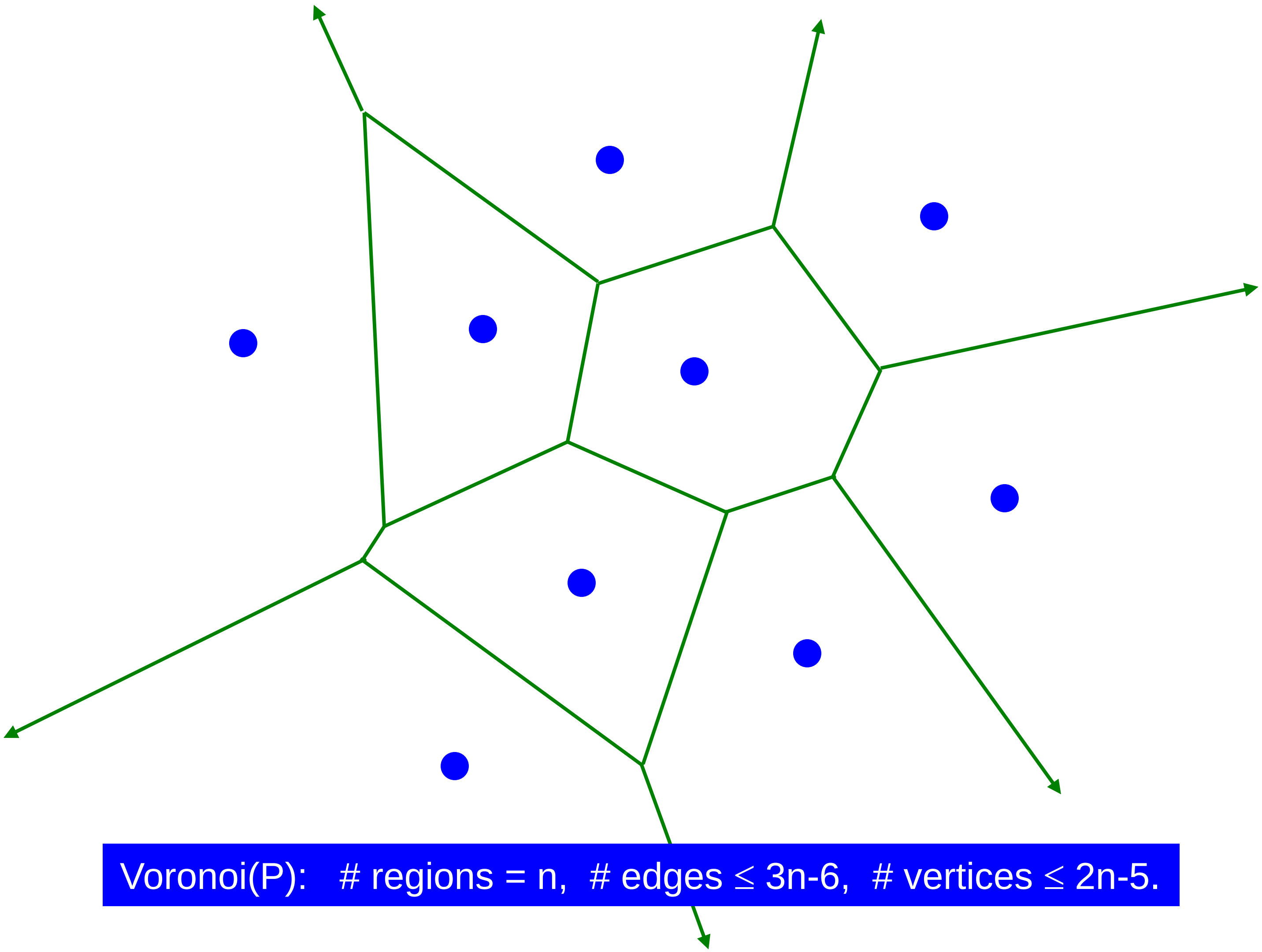


Voronoi Diagram

P = { p_1, p_2, … , p_n} a set of n points in the plane.

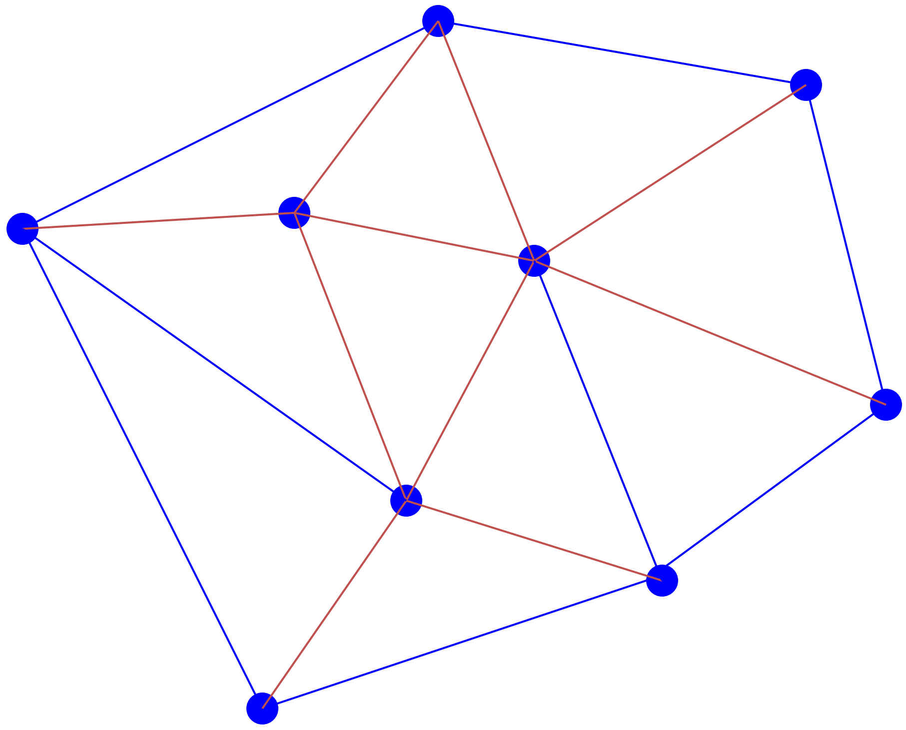


Voronoi Diagram:



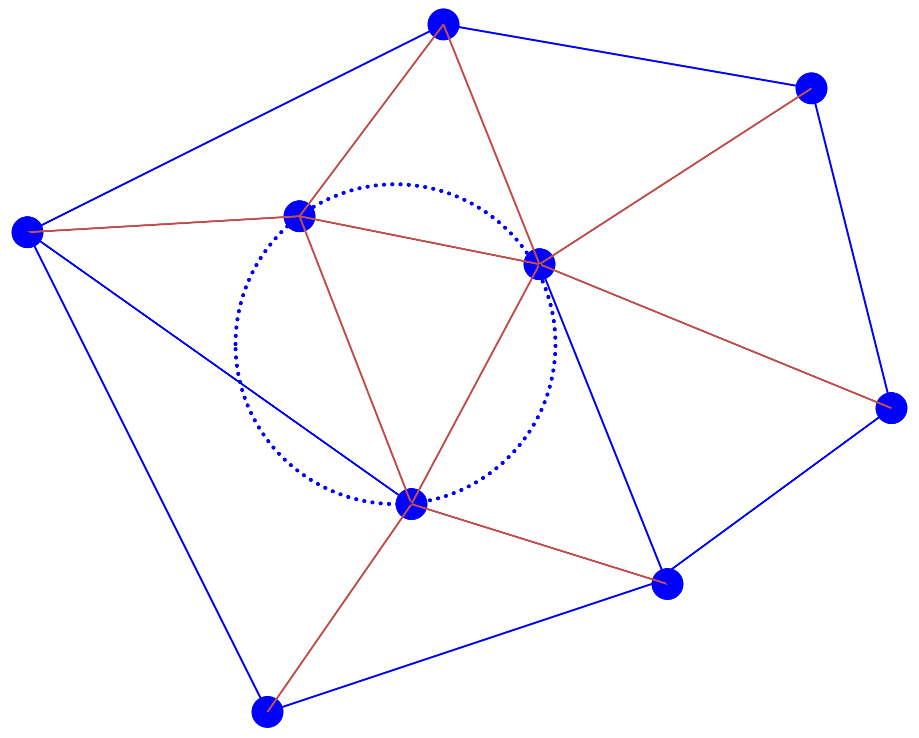
Voronoi(P): # regions = n, # edges ≤ 3n-6, # vertices ≤ 2n-5.

Delaunay Triangulation = Dual of the Voronoi Diagram



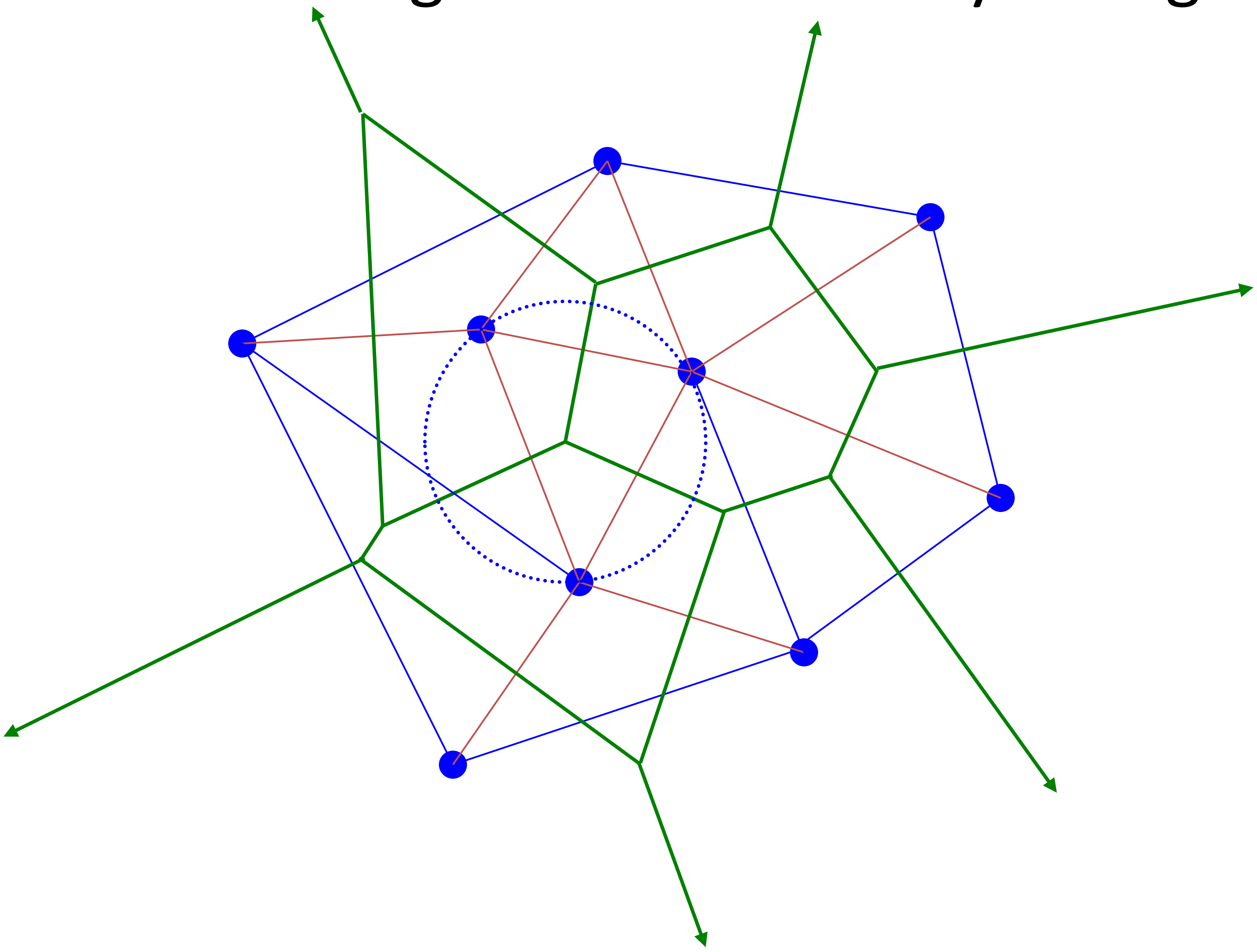
DT(P): # vertices = n, # edges ≤ 3n-6, # triangles ≤ 2n-5.

Delaunay Triangulation



Delaunay triangles have the "empty circle" property.

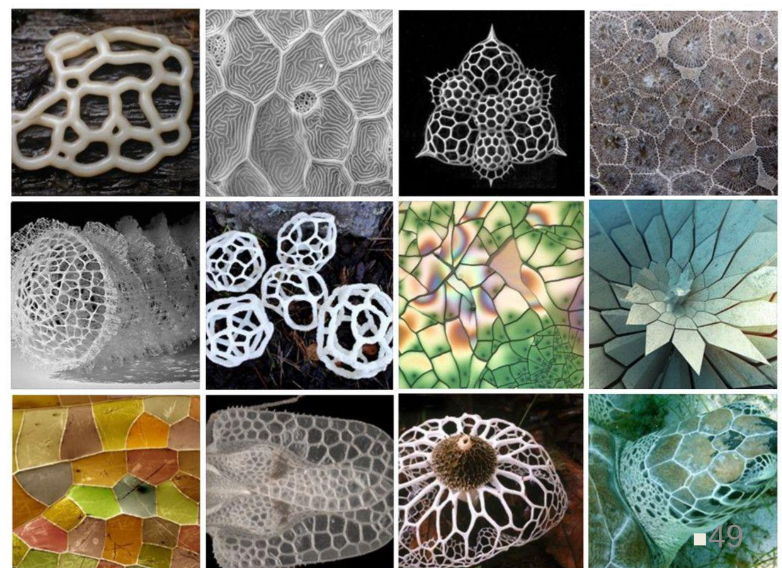
Voronoi Diagram and Delaunay Triangulation



VD Properties

- Each Voronoi region $V(p_i)$ is a convex polygon (possibly unbounded).
- $V(p_i)$ is unbounded $\Leftrightarrow$ p_i is on the boundary of CH(P).
- Consider a Voronoi vertex $v = V(p_i) \cap V(p_j) \cap V(p_k)$.
  Let C(v) = the circle centered at v passing through p_i, p_j, p_k.
- C(v) is circumcircle of Delaunay Triangle (p_i, p_j, p_k).
- C(v) is an empty circle, i.e., its interior contains no other sites of P.

Voronoi Regions in Nature

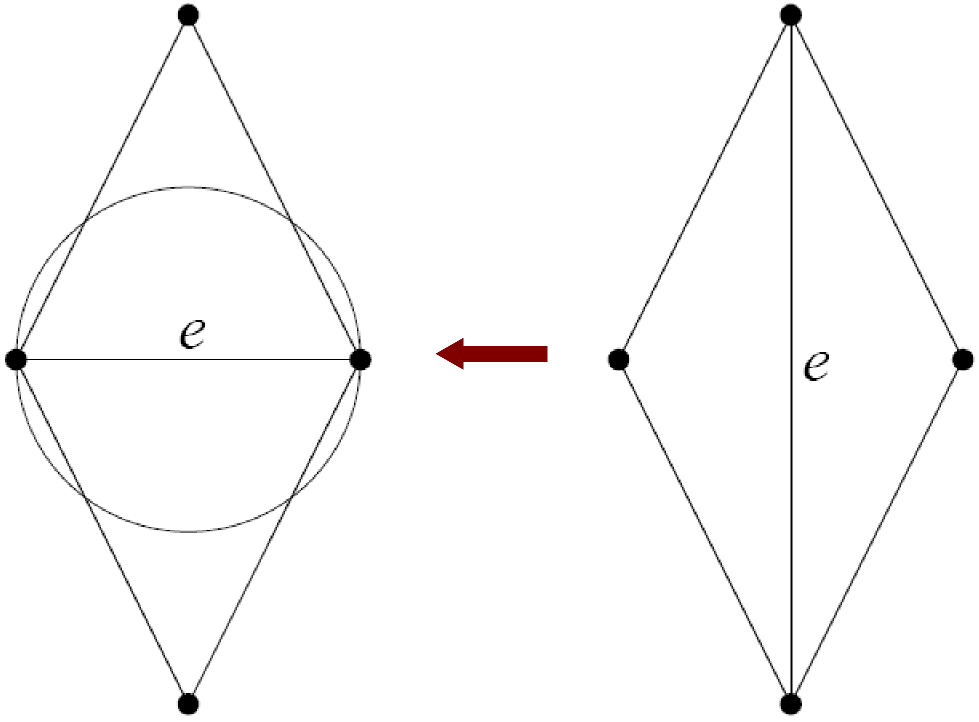


Computing Delaunay Triangulation

- Many algorithms: O(nlogn)

- Lets use flipping:
  - Recall: A *Delaunay Triangulation* is a set of triangles T in which each edge of T possesses at least one empty circumcircle.
  - Empty: A circumcircle is said to be empty if it contains no nodes of the set V

What is a flip?



A non-Delaunay edge flipped

Flip Algorithm

- ??

Flip Algorithm

1. Let V be the set of input vertices.
2. T = Any Triangulation of V.
3. Repeat until all edges of T are Delaunay edges.
   a. Find a non-delaunay edge that is flippable
   b. Flip

Naïve Complexity: $O(n^2)$

Locally Delaunay → Globally Delaunay

- If T is a triangulation with all its edges locally Delaunay , then T is the Delaunay triangulation.
- Proof by contradiction:
  - Let all edges of T be locally Delaunay but an edge of T is not Delaunay, so flip it…

Flipping

- Other flipping ideas?

Randomized Incremental Flipping

- Complexity can be O(nlogn)

Fortune’s Algorithm

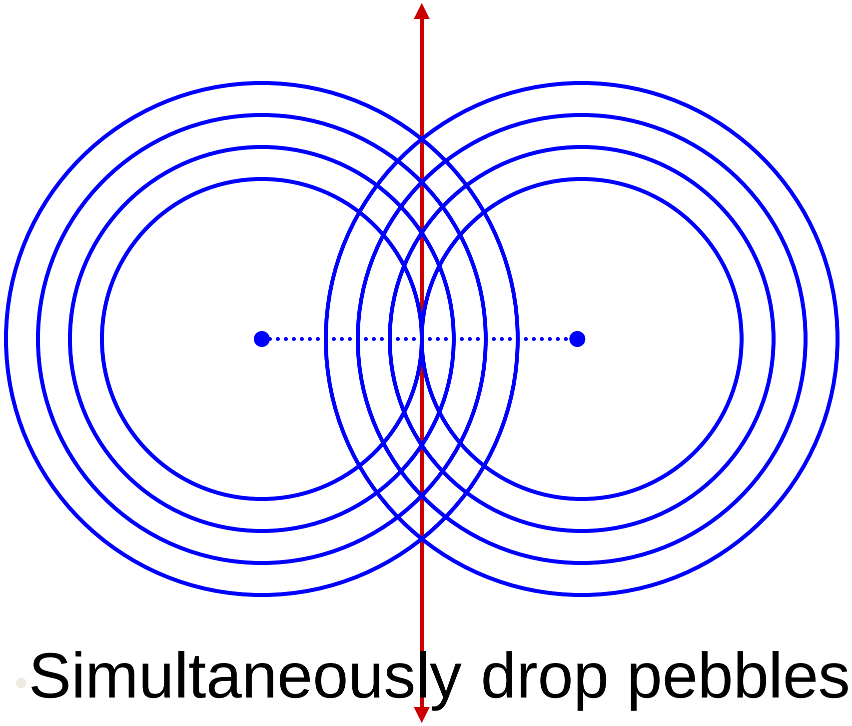
- “A sweepline algorithm for Voronoi “Algorithms”, 1987, O(nlogn)

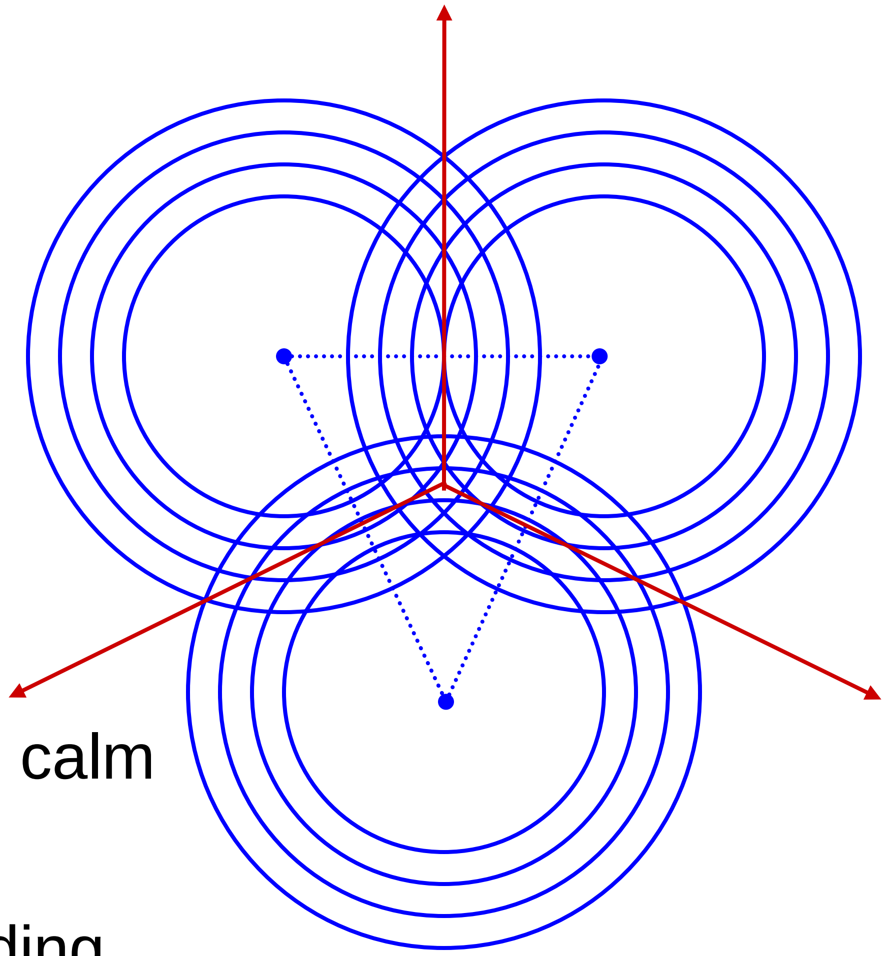
  https://www.youtube.com/watch?v=k2P9yWSMaXE

  Pseudocode:

```
add a site event in the event queue for each site
while the event queue is not empty
        pop the top event
        if the event is a site event
                insert a new arc in the beachline
                check for new circle events
        else
                create a vertex in the diagram
                remove the shrunk arc from the beachline
                delete invalidated events
                check for new circle events
```

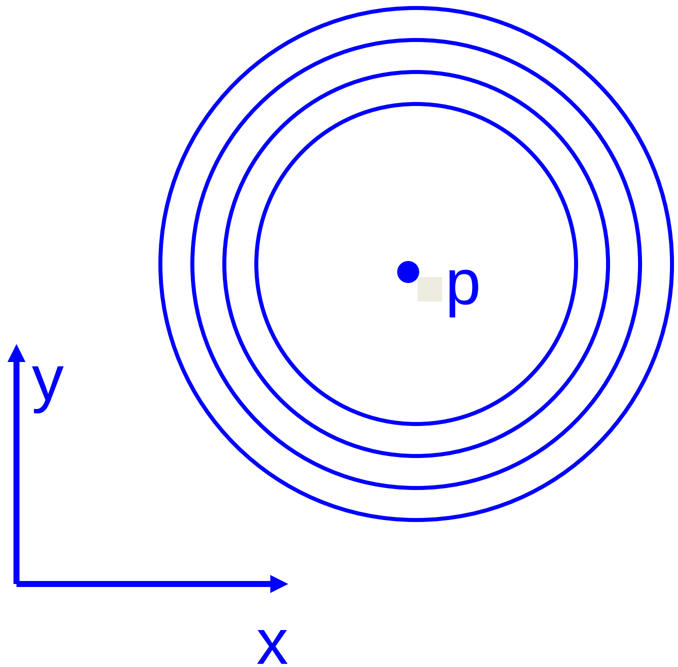
Wave Propagation View

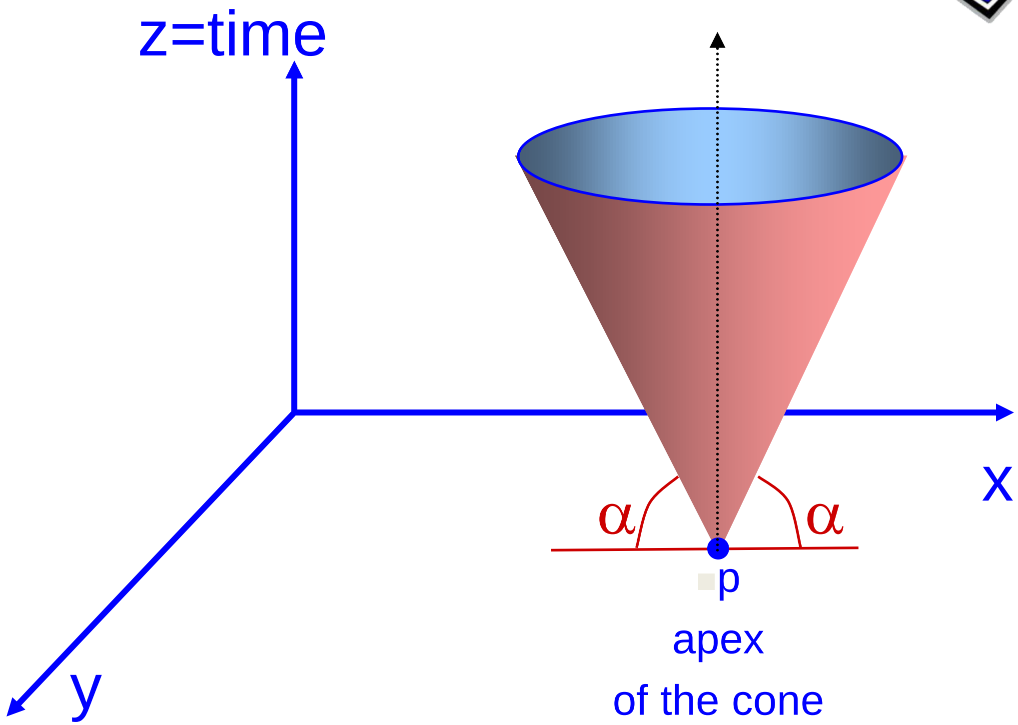




- Simultaneously drop pebbles on calm lake at n sites.
- Watch the intersection of expanding waves.

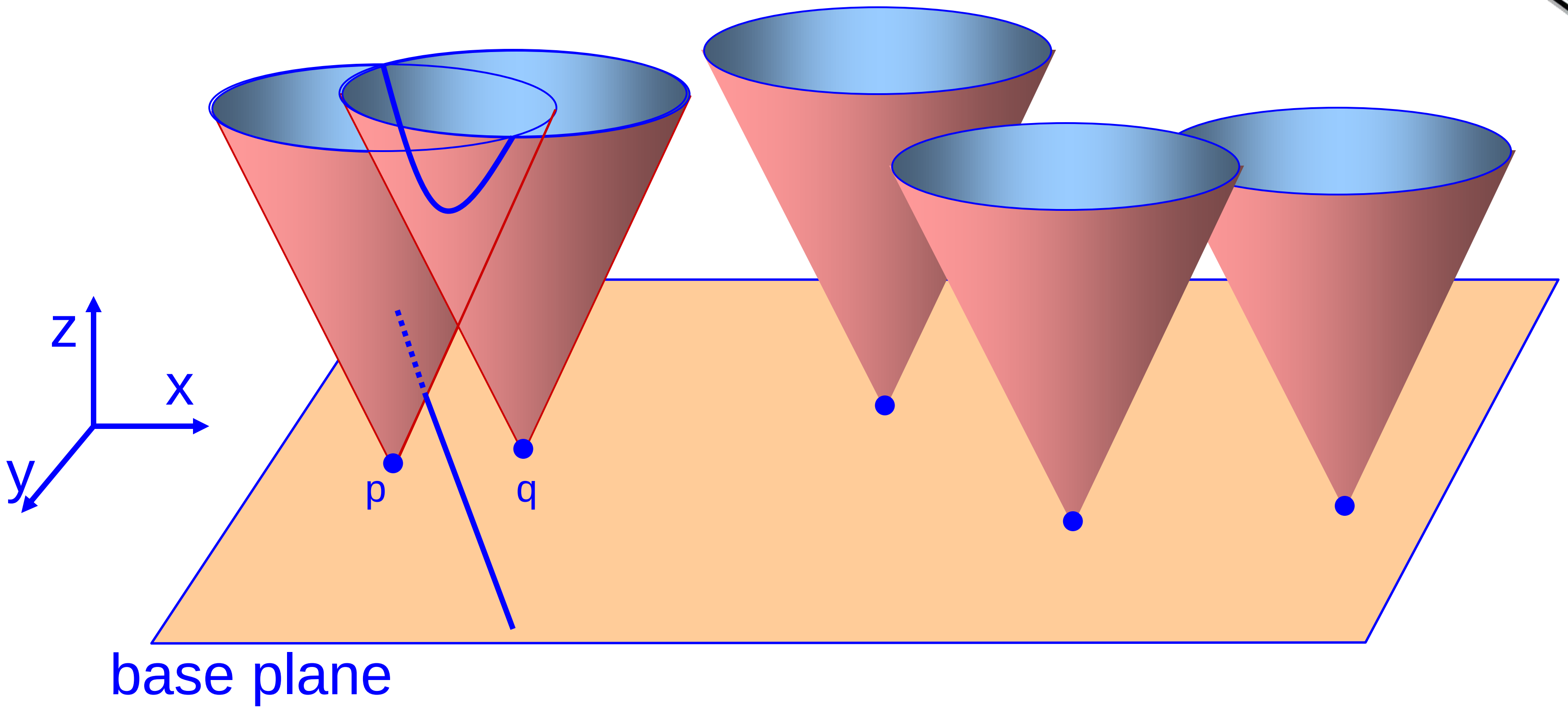
Let Time be the 3rd Dimension





All sites have identical opaque cones.

Let Time be the 3rd Dimension



All sites have identical opaque cones.

cone(p) ∩ cone(q) = vertical hyperbola h(p,q).

vertical projection of h(p,q) on the xy base plane is PB(p,q).

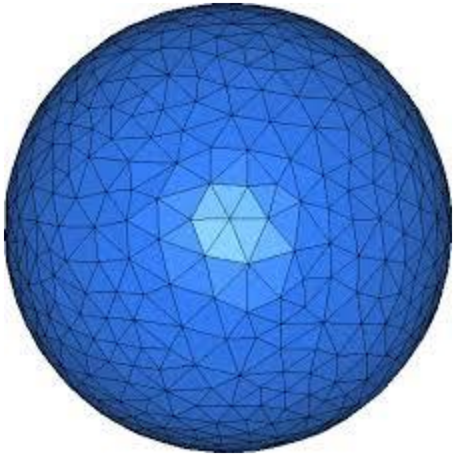
Voronoi Diagrams

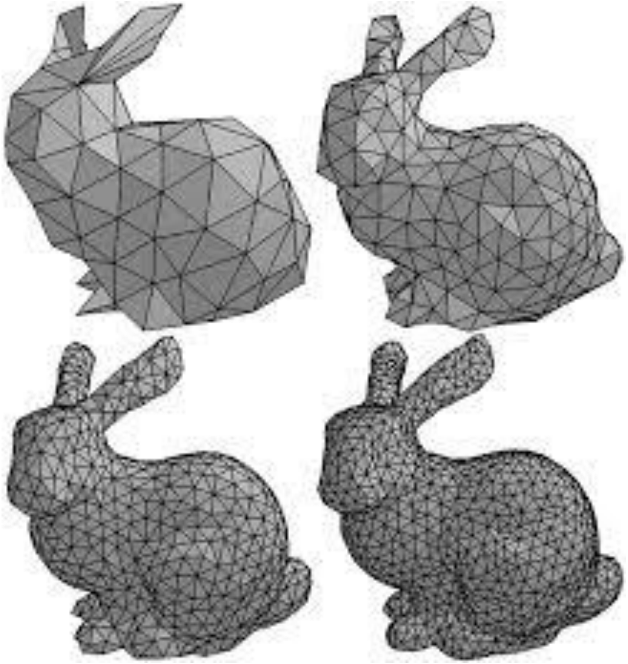
- http://alexbeutel.com/webgl/voronoi.html

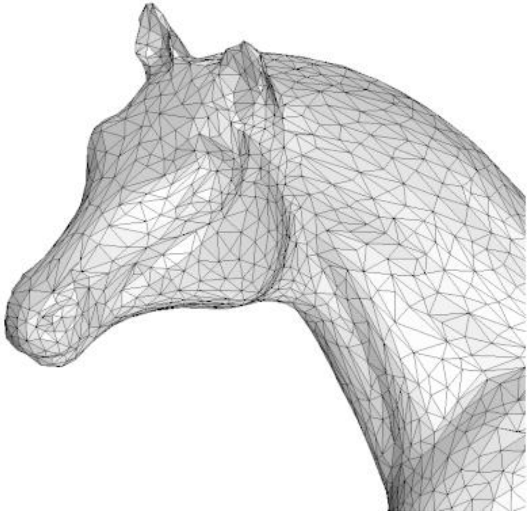
Voronoi Diagram

- http://www.raymondhill.net/voronoi/rhill-voronoi-demo5.html

Examples Triangulations







And Beyond…

- Not "relaxation" but more general:
  - Reaction Diffusion…
    - https://pmneila.github.io/jsexp/grayscott/
    - Textures: