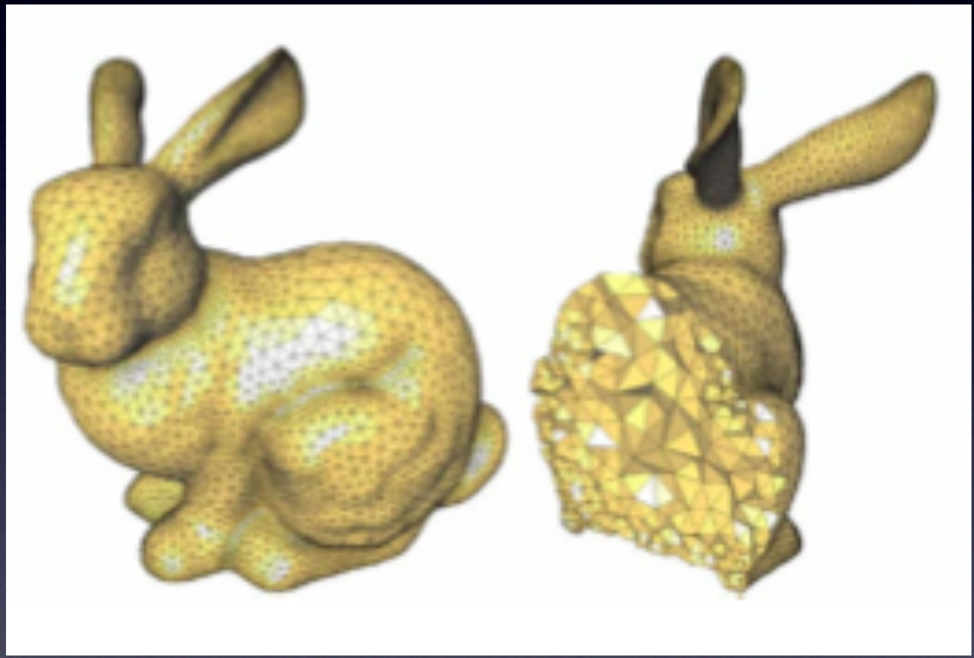
Variational Tetrahedral Meshing

Pierre Alliez

David Cohen-Steiner

Mariette Yvinec

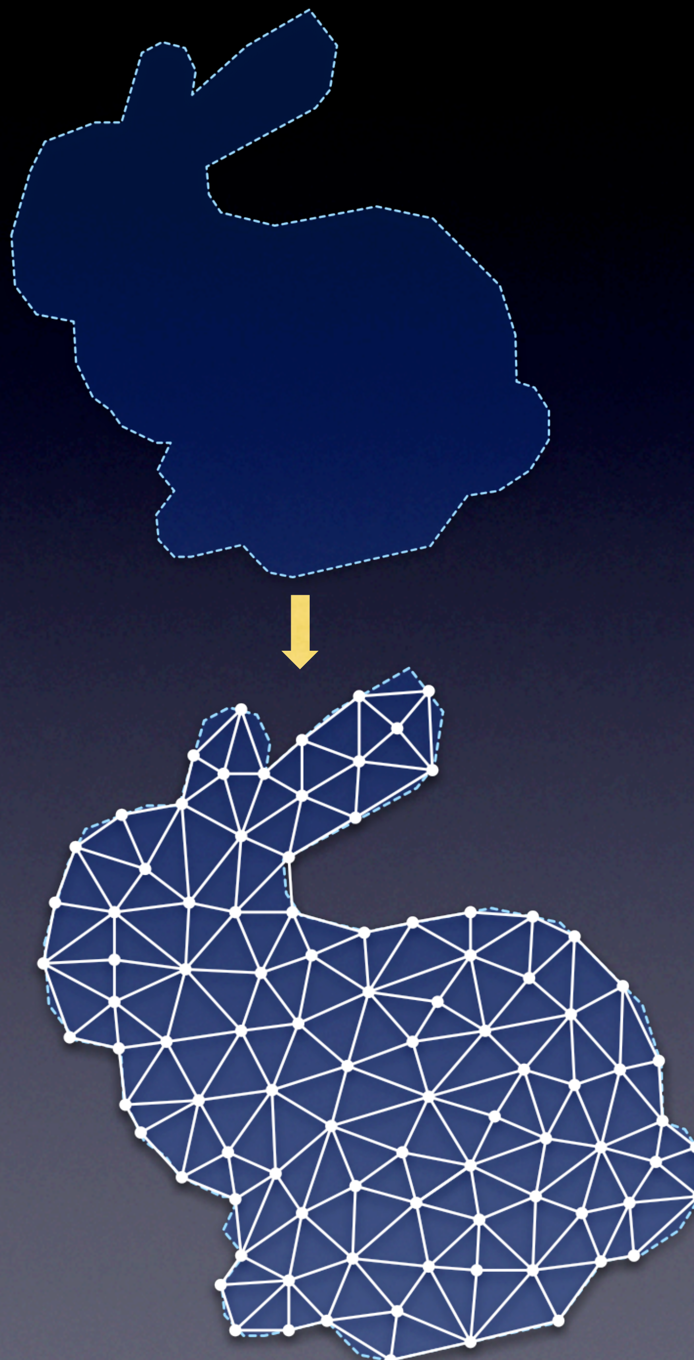
Mathieu Desbrun



Figures and slides borrowed from: BryanK's talk,
Slides the authors posted,
www.cs.uiuc.edu/class/fa05/cs598anh/slides/Bell2005_AlCoYvDe2005.pdf

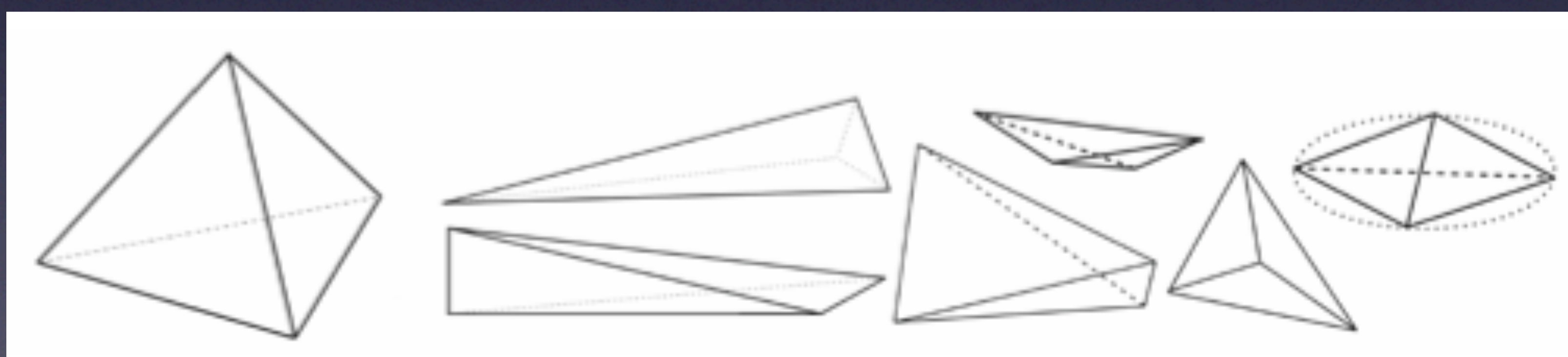
Goals

- 2D:
  - In: Non-intersecting closed curve
  - Out: Triangle Mesh
- 3D:
  - In: Given a watertight, non-intersecting manifold triangle mesh
  - Out: Tetrahedral Mesh



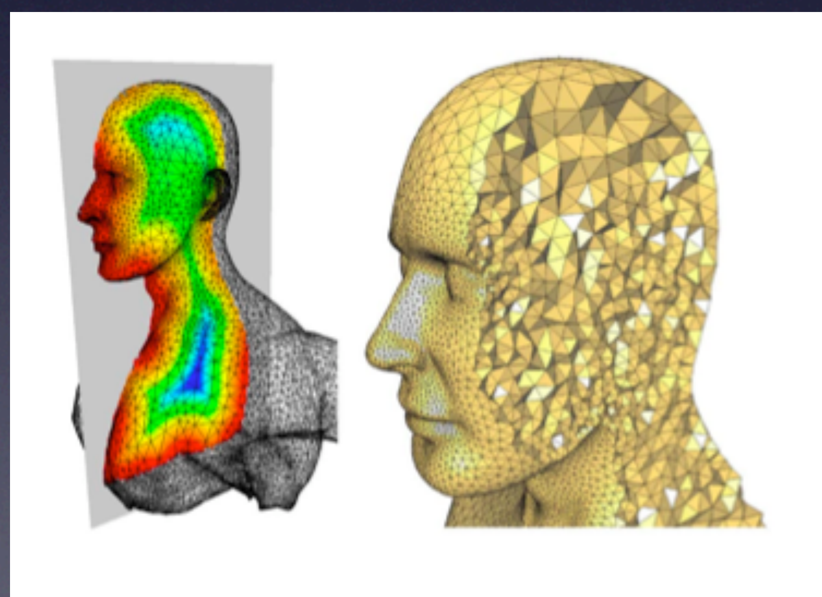
Mesh quality

- In this paper
  - radius-edge ratio = r_{in} / r_{circ}



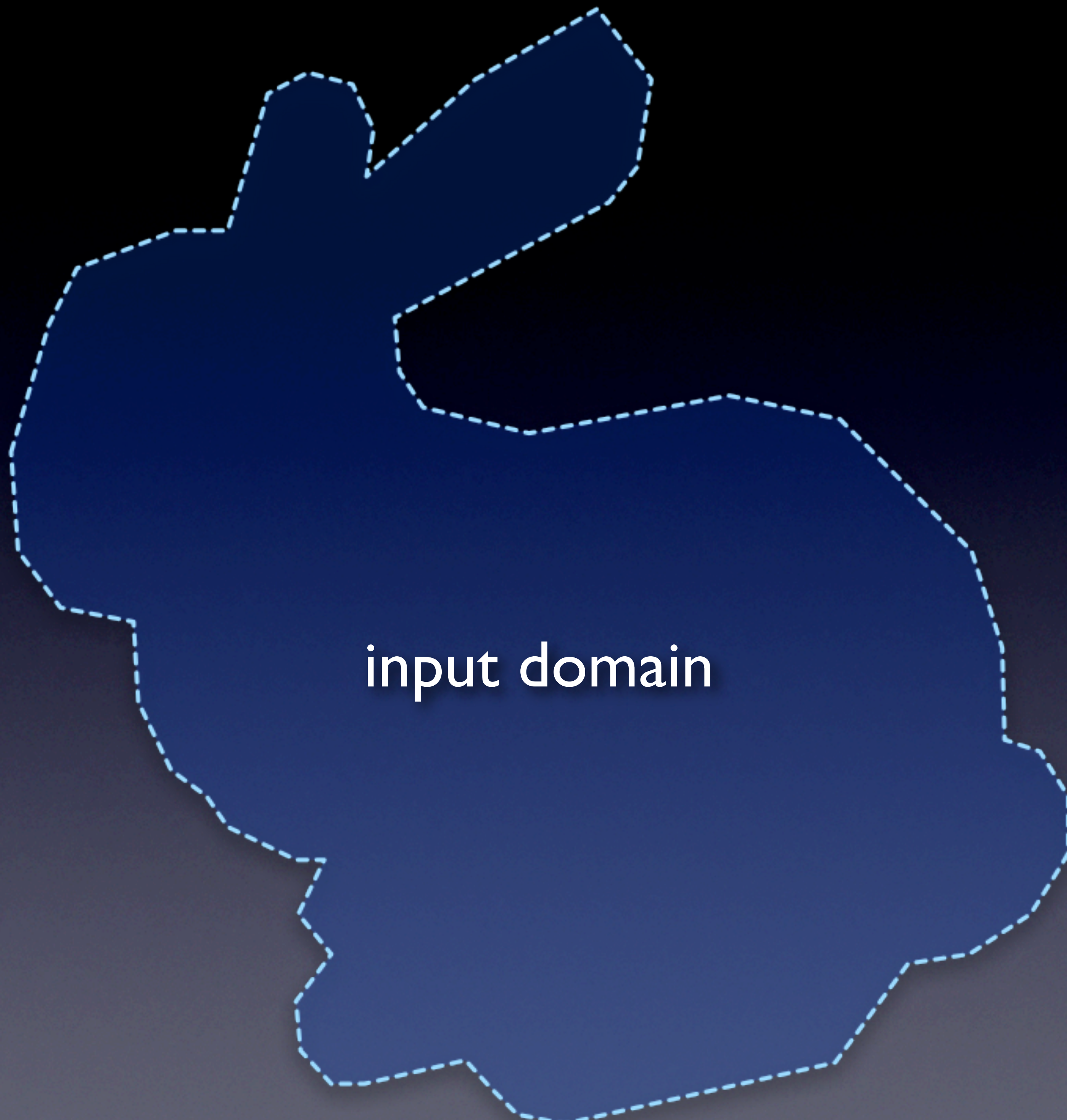
Other requirements

- Graded mesh
  - Tets size based on a sizing field
  - Sizing field $\mu(x) : \Re^3 \rightarrow \Re$
    - Indicate desired tet's edge length near x

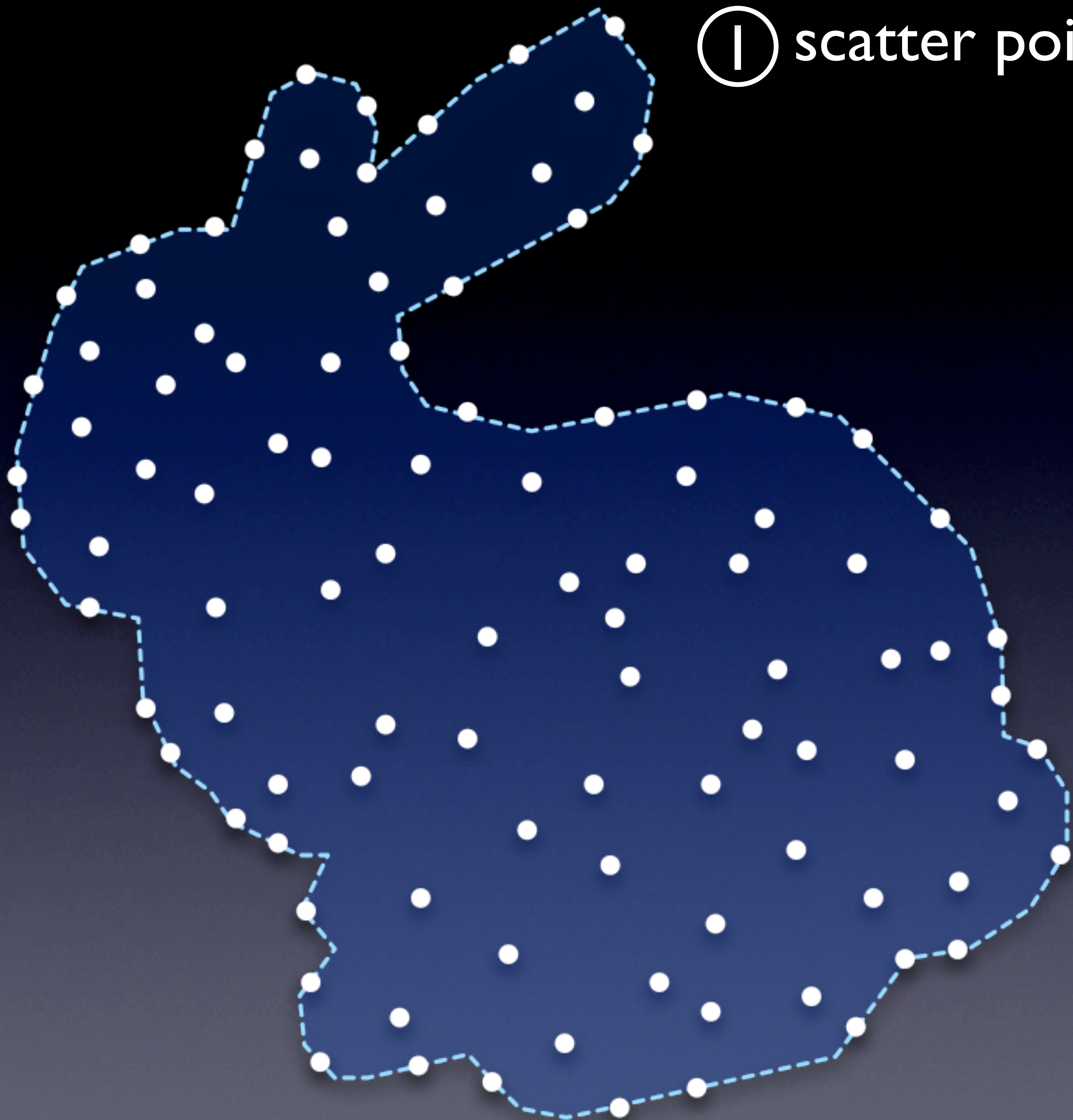


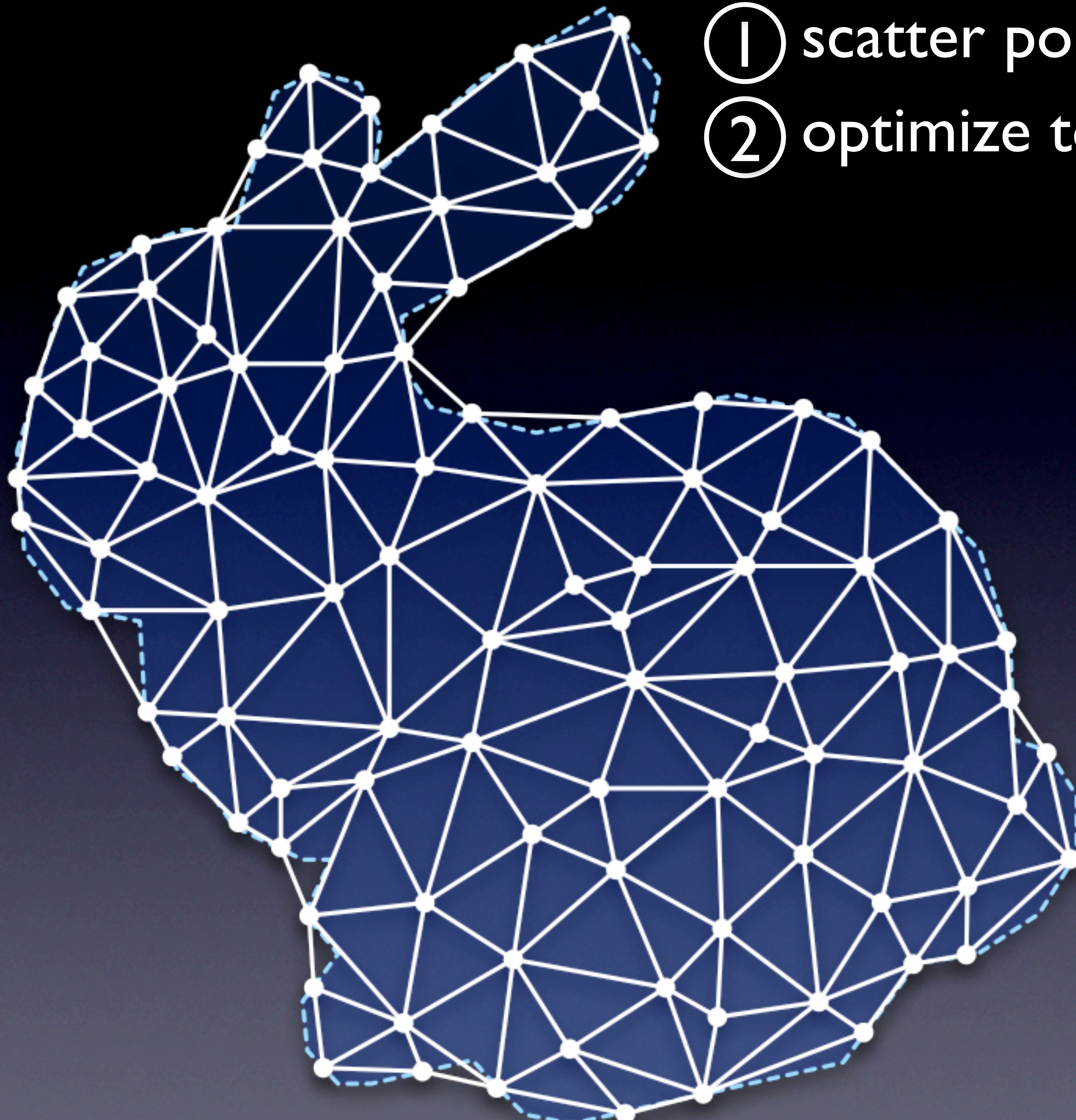
Algorithm

```
Initialize vertices based on sizing field
While (! Good enough quality) {
  Delaunay Triangulation/Tetrahedralization
  Optimize vertices position
}
```


input domain

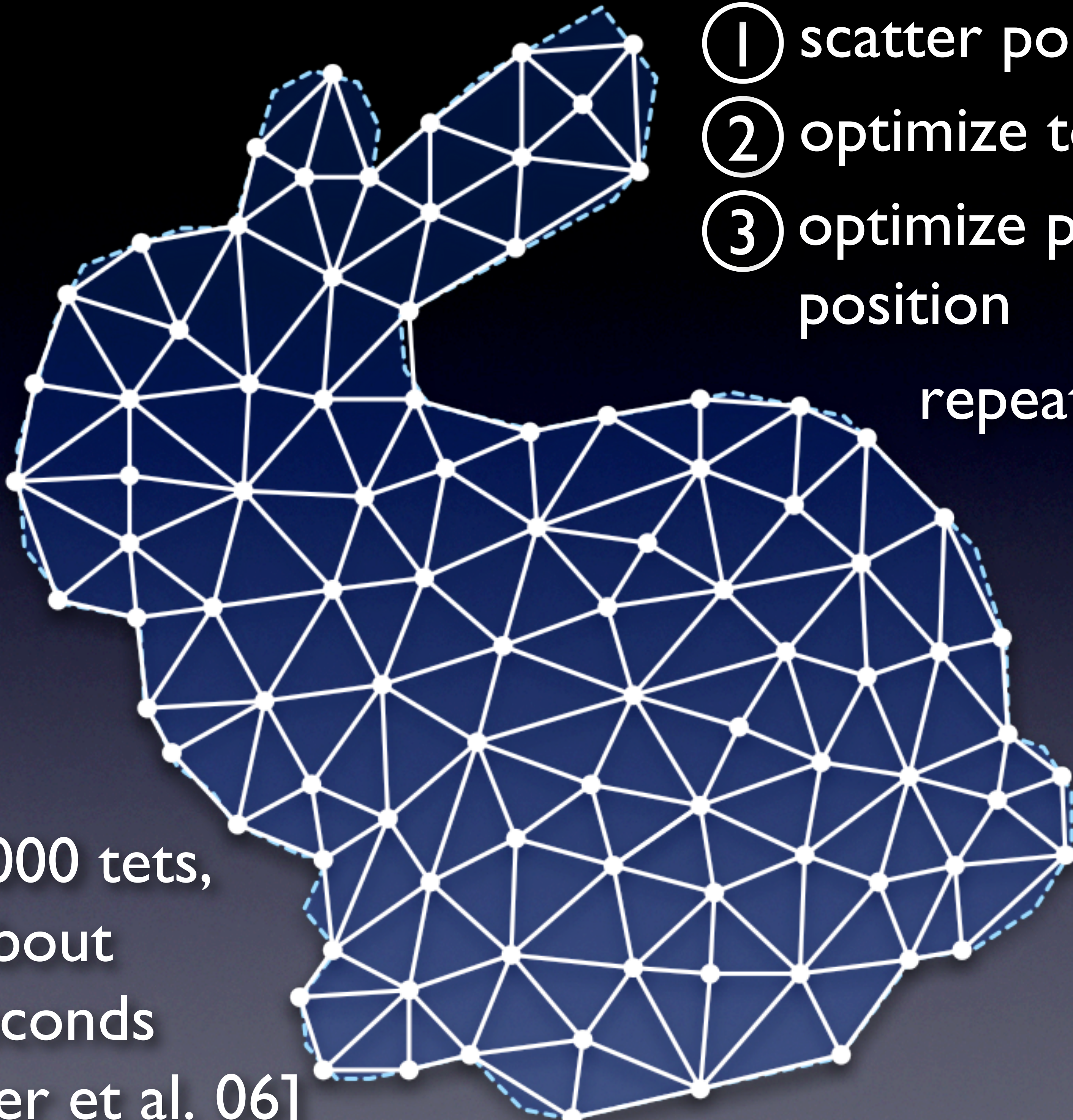
(1) scatter points





① scatter points

② optimize topology



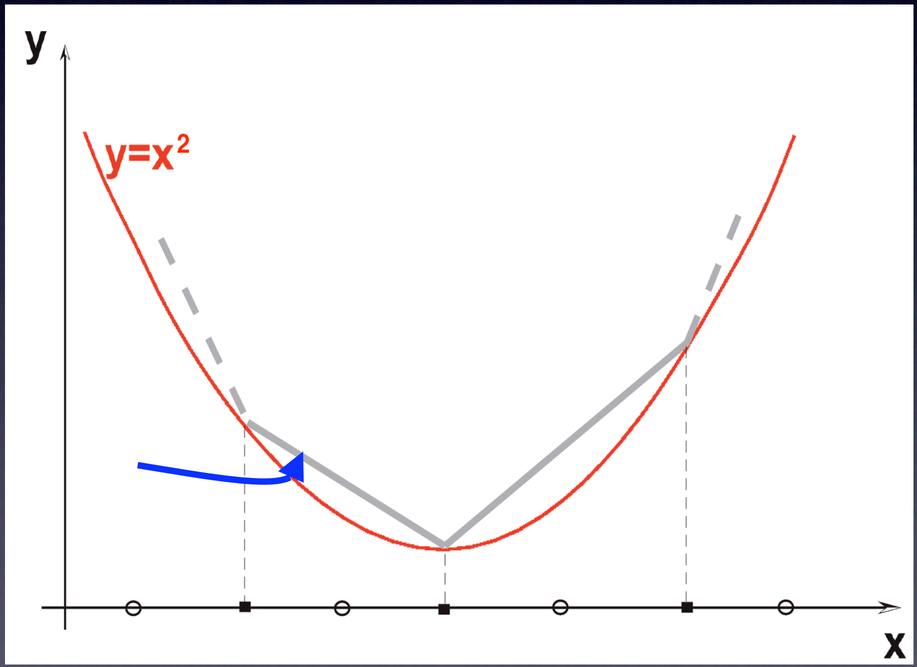
① scatter points
② optimize topology
③ optimize point position

repeat ②,③

for 50,000 tets,
takes about
1-10 seconds
[Klingner et al. 06]

Optimization

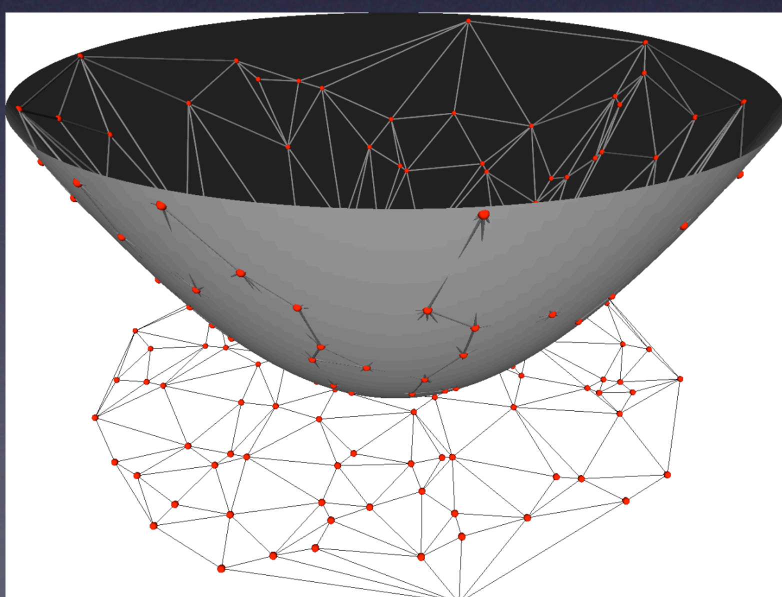
$$E_{ODT} = \left\| f - f_{PWL}^{overlaid} \right\|_{L_1}$$

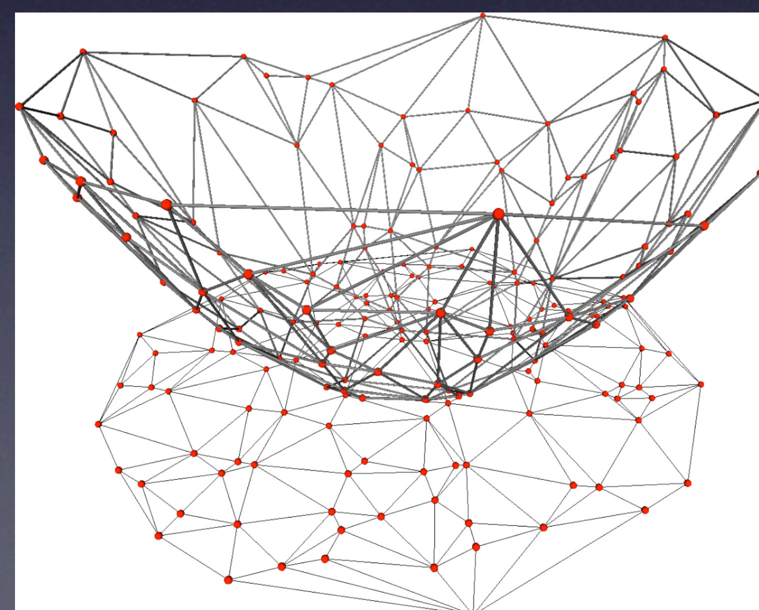


- Minimize area between PWL and paraboloid

Optimization

- For **fixed vertex locations**
  - Delaunay triangulation is *the* optimal connectivity
    - Exists for any points set
    - Has several nice properties





Optimization

- for **fixed connectivity**
  - min of quadratic energy leads to *the* optimal vertex locations

$$E_{CVT} = \frac{1}{N+1} \sum_i \int_{\Omega_i} ||x - x_i||^2 dx$$

- x_i is vertex i position
- $|\Omega_i|$ is volume of tets in 1-ring neighbor of vertex i

Optimal vertex position

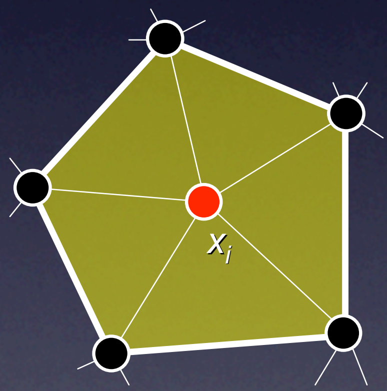
- For uniform sizing field, turns out to be

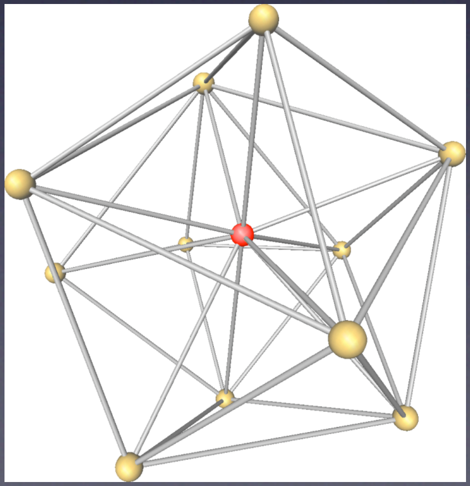
$$\mathbf{x}_i^\star = \frac{1}{|\Omega_i|} \sum_{T_j \in \Omega_i} |T_j| \mathbf{c}_j$$

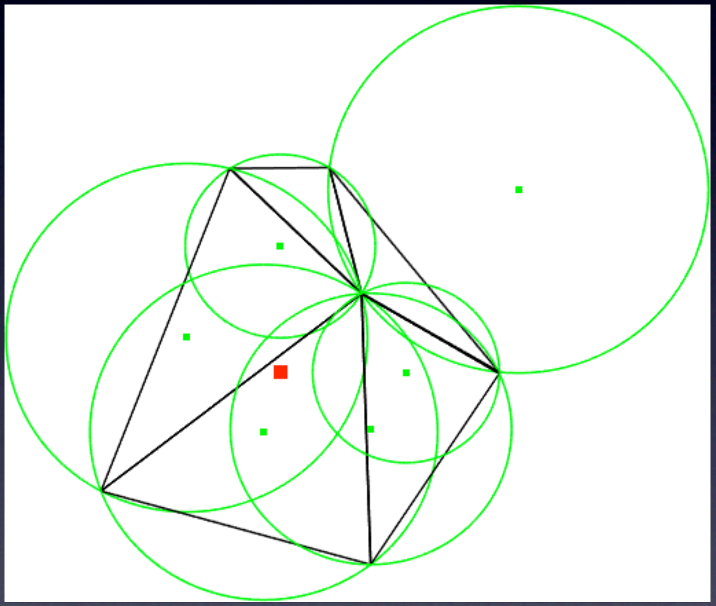
- $|T_i|$ is volume of tet i
- c_i is circumcenter of tet i

Optimal vertex position

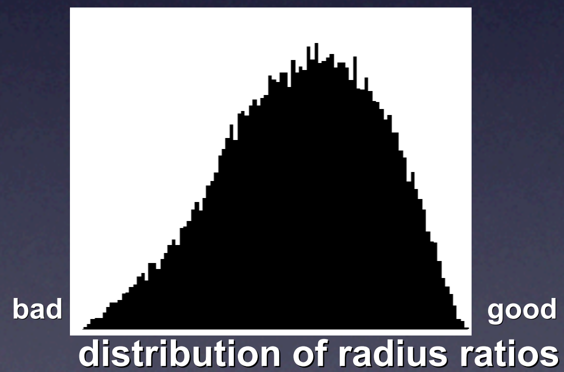
$$x_i^* = \frac{1}{|\Omega_i|} \sum_{T_j \in \Omega_i} |T_j| c_j$$

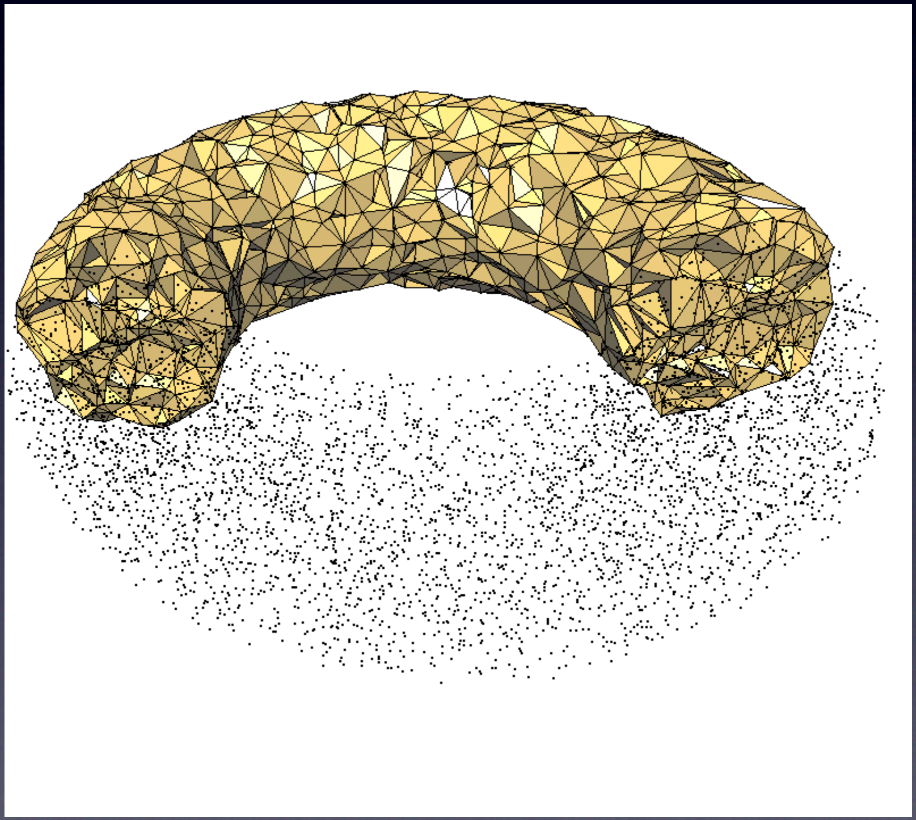




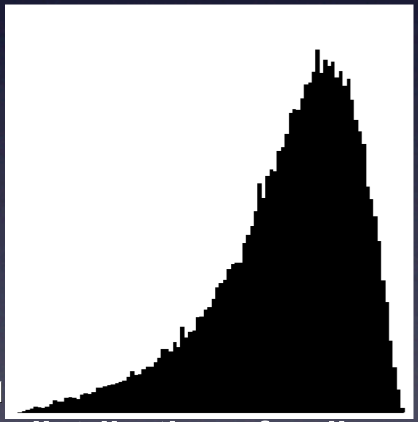


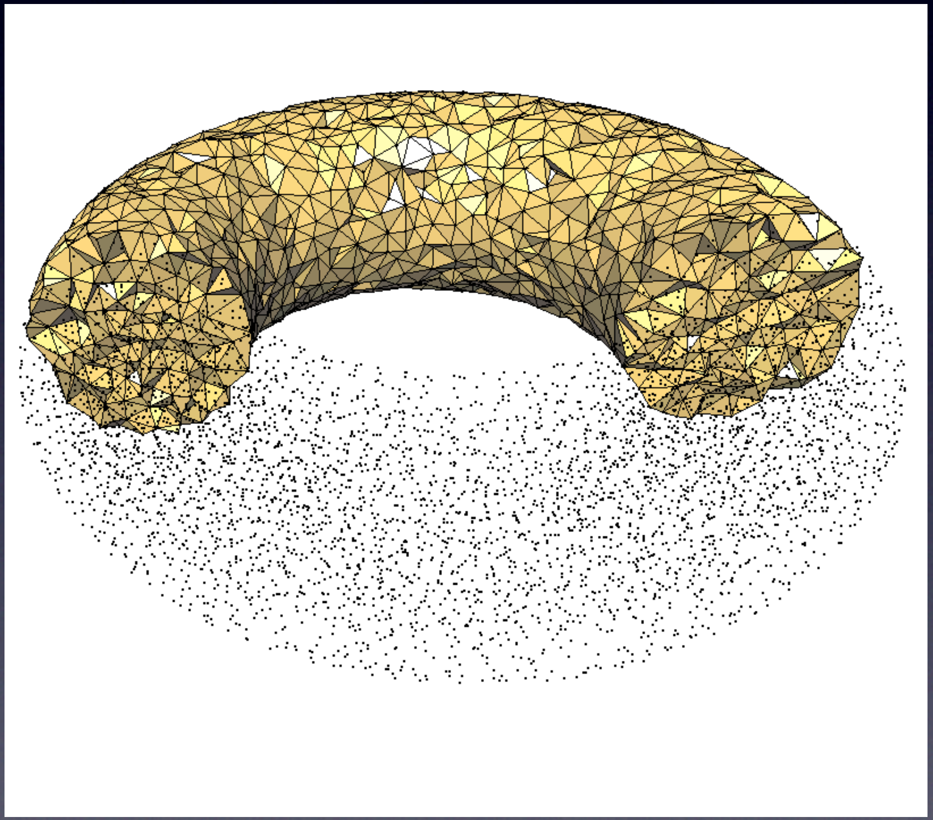
Optimization: Init



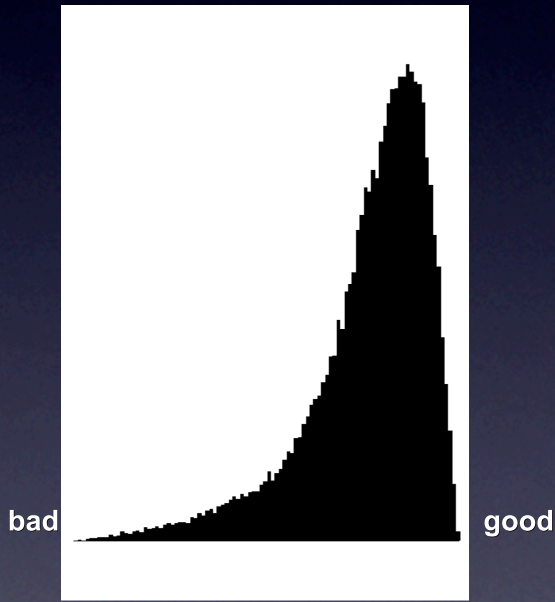


Optimization: Step 1

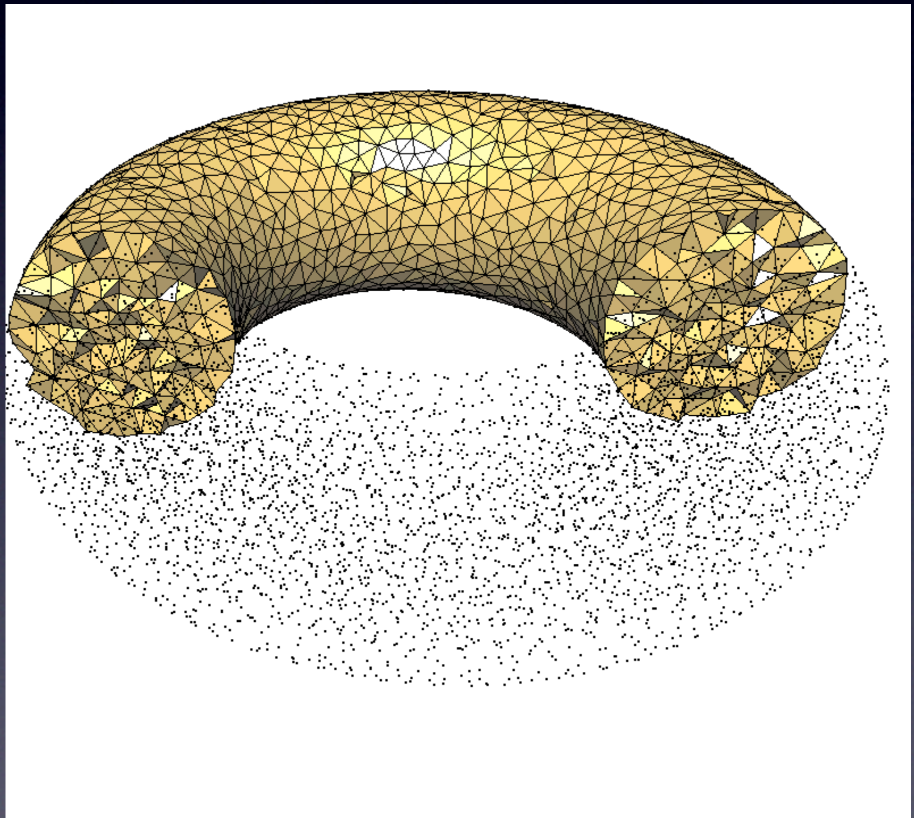




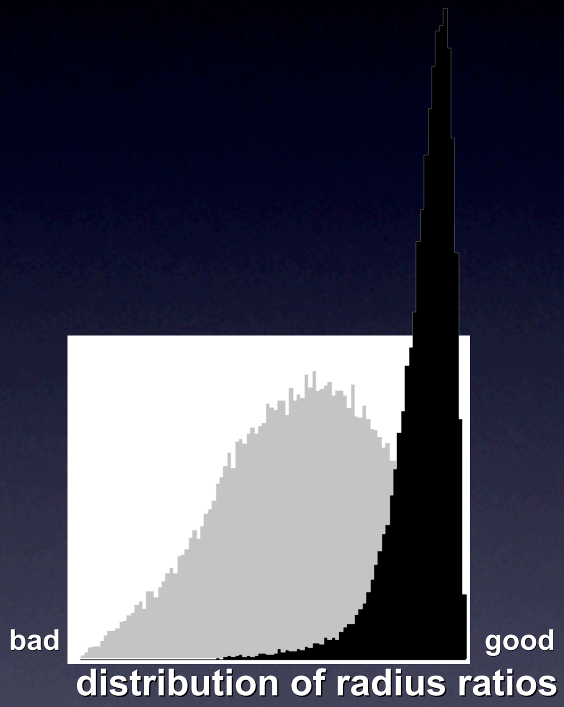
Optimization: Step 2

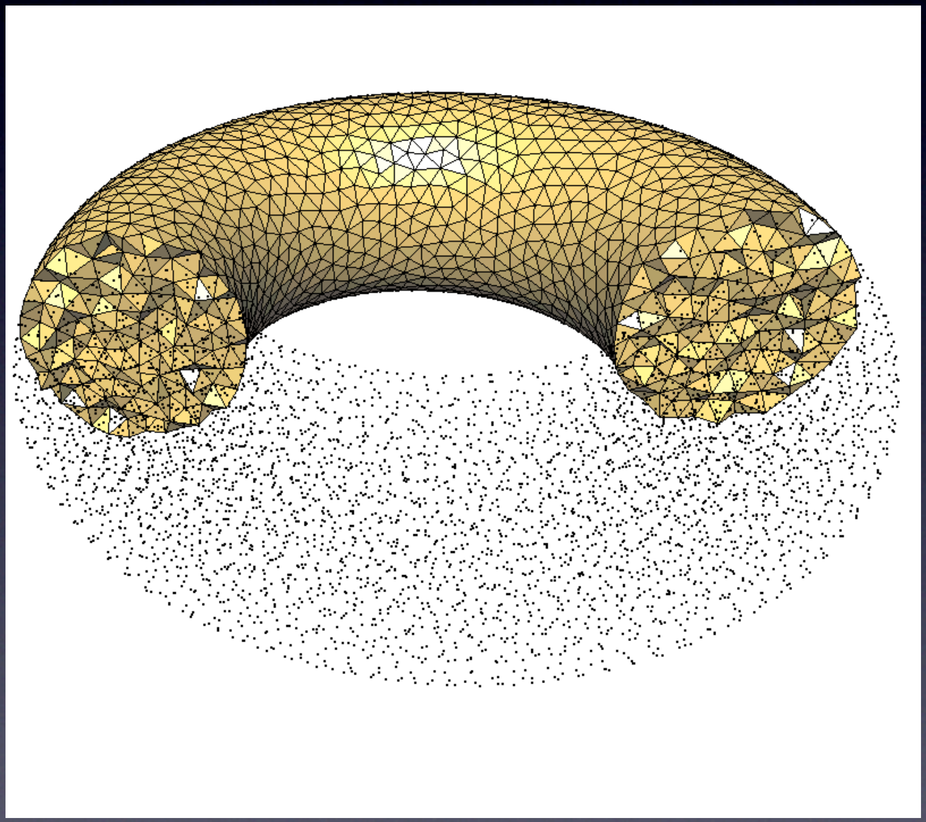


distribution of radius ratios



Optimization: Step 50





Graded mesh

- So far, uniform, we also want:
  - To minimize number of elements
  - To better approximate the boundary
    - While preserving good shape of elements

Sizing Field!

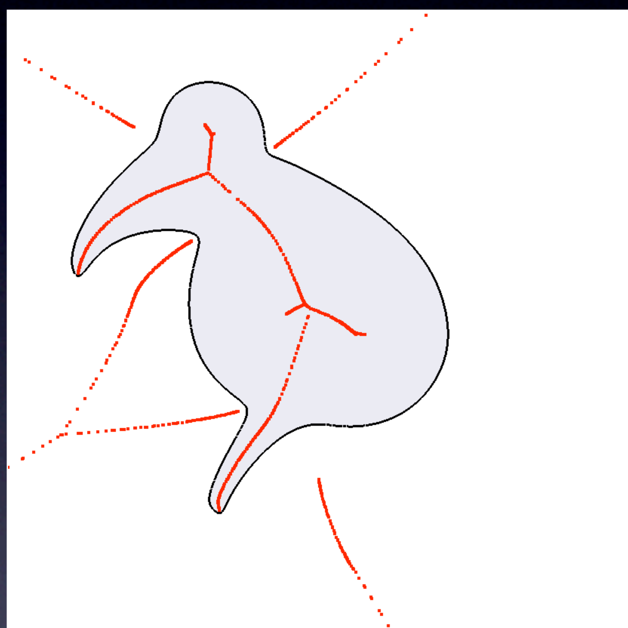
Sizing Field

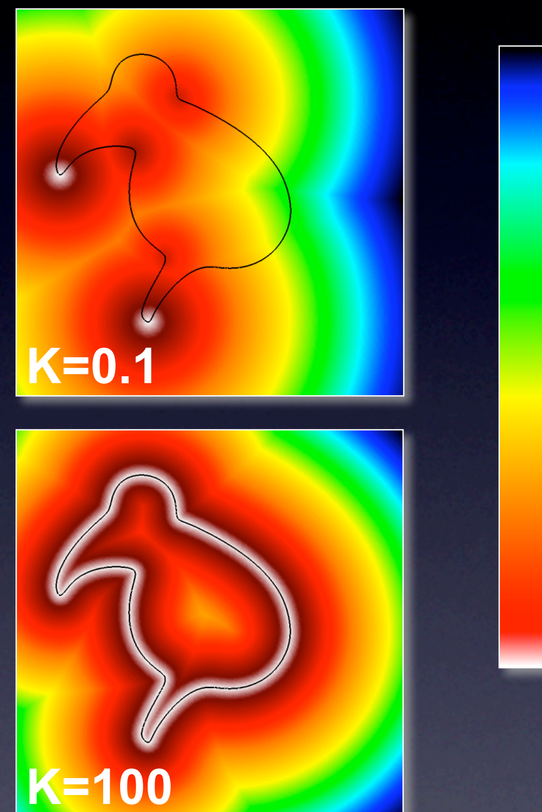
Properties:

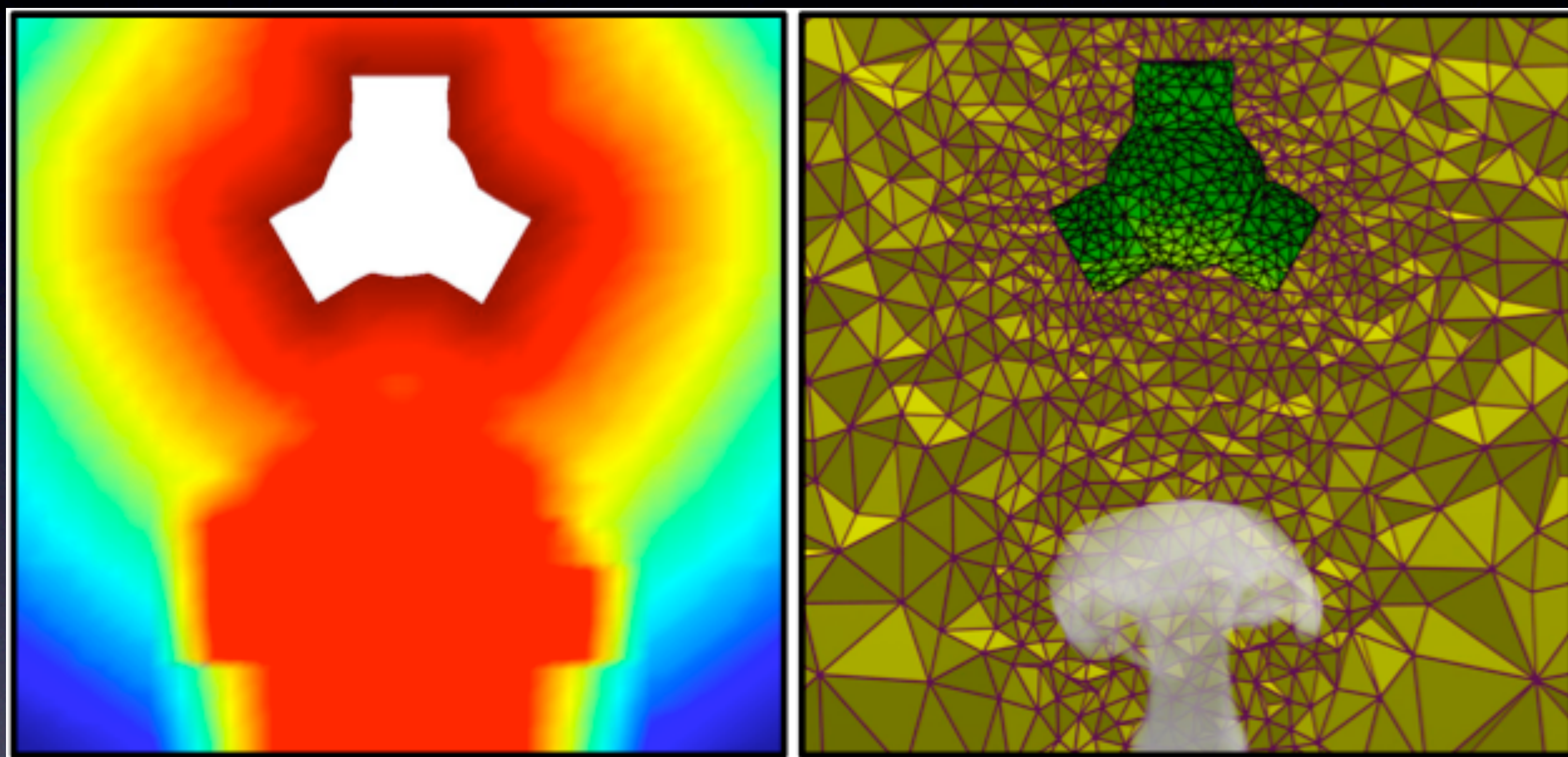
- size ≤ *lfs* (local feature scuize) on boundary
  - lfs = Distance to medial axis
- sizing field is K-Lipschitz

$$\mu(x) = \inf_{y \in \partial\Omega} \left[K \| x - y \| + lfs(y) \right]$$

parameter




K=0.1
K=100



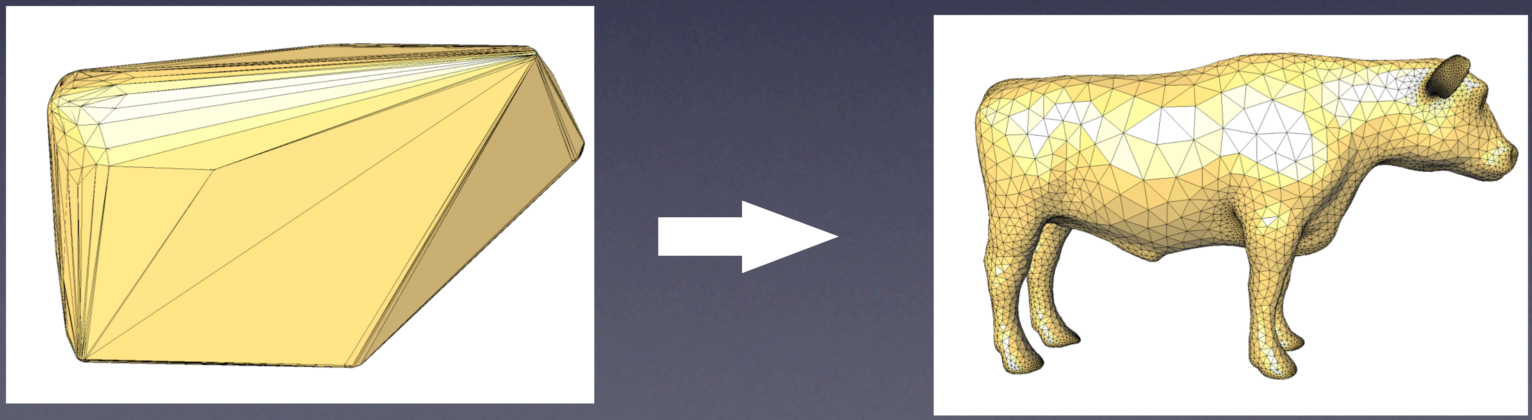
Need to modify vertex optimization

$$\mathbf{x}_i^\star = \frac{1}{\sum_{T_k \in \Omega_i} \frac{|T_k|}{\mu^3(\mathbf{g}_k)}} \sum_{T_j \in \Omega_i} \frac{|T_j|}{\mu^3(\mathbf{g}_j)} \mathbf{c}_j$$

Intuition:
Tet whose sizing field at circumcenter is small has big weight

Other details

- Need to handle vertices near boundary specially
  - The vertex optimization does not respect boundary
- Need to get rid of tets outside the mesh
  - Because DT include tets that cover convex hull



Boundary Handling

- Create densely sampled set of points on the surface, quadrature points
  - Associate weight with each quadrature point
    - Corner - Infinite weight
    - Crease - $dl / \mu(x)^3$
    - Surface - $ds / \mu(x)^4$

Boundary Handling

- Loop through all quadrature points, q
  - Let v be the closest vertex to q
  - S(v) = S(v) U {q}
- For all vertex v,
  - If S(v) != Ø,
    - Position(v) = weighted average of position of q's in S(v)
  - Else
    - Position(v) will be determined by the optimization

Outside tet strippping

- The method in the paper does not seem to work.
- What we did:
  - Loop through all tets:
    - A tet is outside if 4 vertices of a tet are boundary vertices and
      - Its quality is bad OR
      - Its barycenter is outside
  - Then loop through all tets:
    - If >= 2 of its neighboring tets are outside (as determined from the previous step) , this tet is outside as well

Observations

- Worst tets usually found near boundary
- Worst tets quality improve when we replace circumcenter with barycenter in the vertex optimization
  - No theoretical support
  - Average quality decrease