Using Voronoi to determine dimensionality

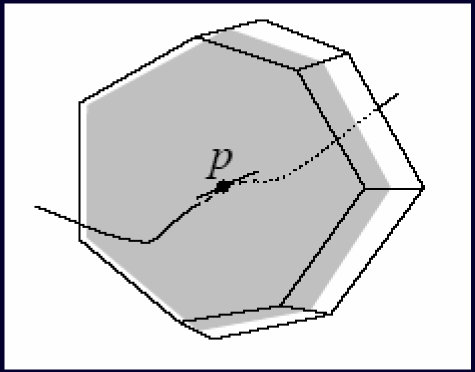
Undersampling and Oversampling in Sample Based Shape Modeling

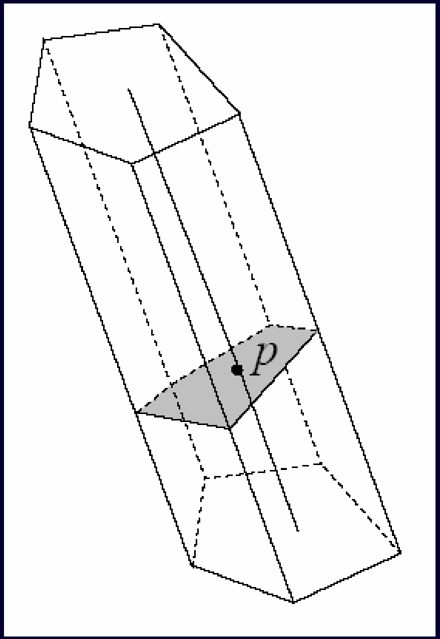
(Detecting Undersampling in Surface Reconstruction)

Shape Dimension and Approximation from Samples

Properties of Voronoi cells

- Reflects local neighborhood
- Round
  - Points in all directions
- Skinny
  - Points in a plane





Issues in surface reconstuction

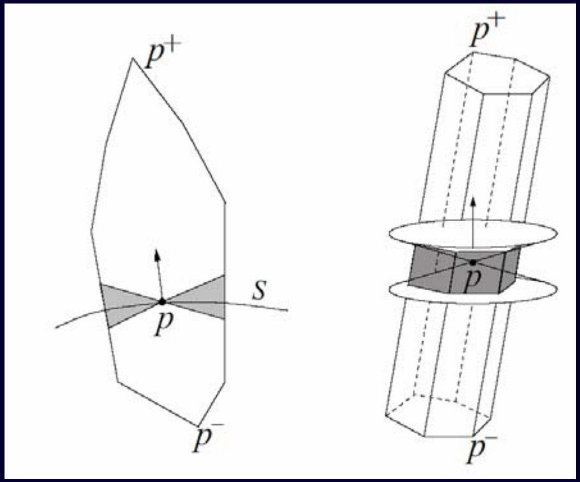
- Noise
- Partial scans
  - Boundary detection
- Undersampling
  - Real data does not respect sampling condition
- Oversampling
  - Too many points in flat areas

Definitions

- (ε, δ) sample
  - Each $x \in M$ has a sample within $\varepsilon f(x)$
  - Each sample has no point within $\delta f(x)$
- Pole
  - Farthest vertex in Voronoi cell
  - Approximates normal
- Height
  - Distance between point and negative pole

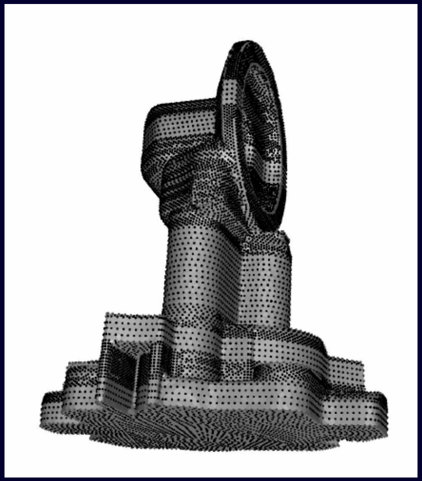
Cocone

- $C_p(\theta, v) = \{y \in V_p : ((y-p),v) \geq \pi/2-\theta\}$
- Usually v is the pole
- q is a cocone neighbor of p if the cocone of p overlaps with V_q



Cocone Algorithm

- Compute Voronoi
- Determine candidate triangles
  - Using the cocone
- Remove triangles with free edges
- Extract manifold



What is well sampled?

- Long skinny Voronoi cells
- The poles of neighbors agree
- The paper codifies these ideas
- Example picture

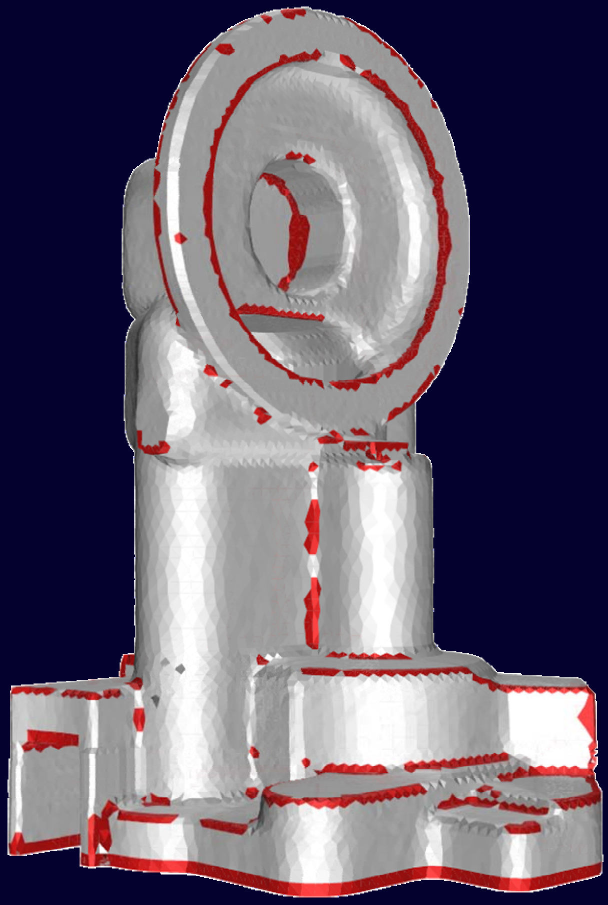
Flat Points

- Good points, define by ρ, α
  - Ratio condition
    - $r_p/h_p \leq \rho$
  - Normal condition

    $\forall$ q with $p \in N_q$, $(v_p, v_q) \leq \alpha$
- Samples which are not Cocone neighbors to a boundary sample are flat

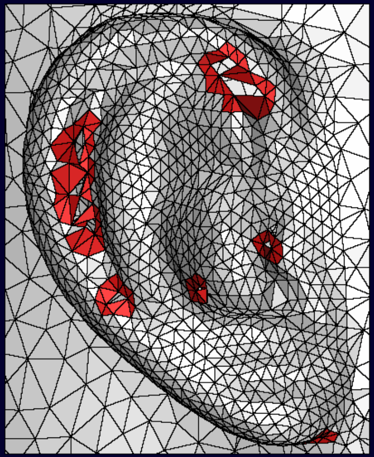
Boundary detection

- Identify interior points
- Grow from interior points
  - If normal is close to one previous point
- Return rest as boundary



Surface Reconstruction

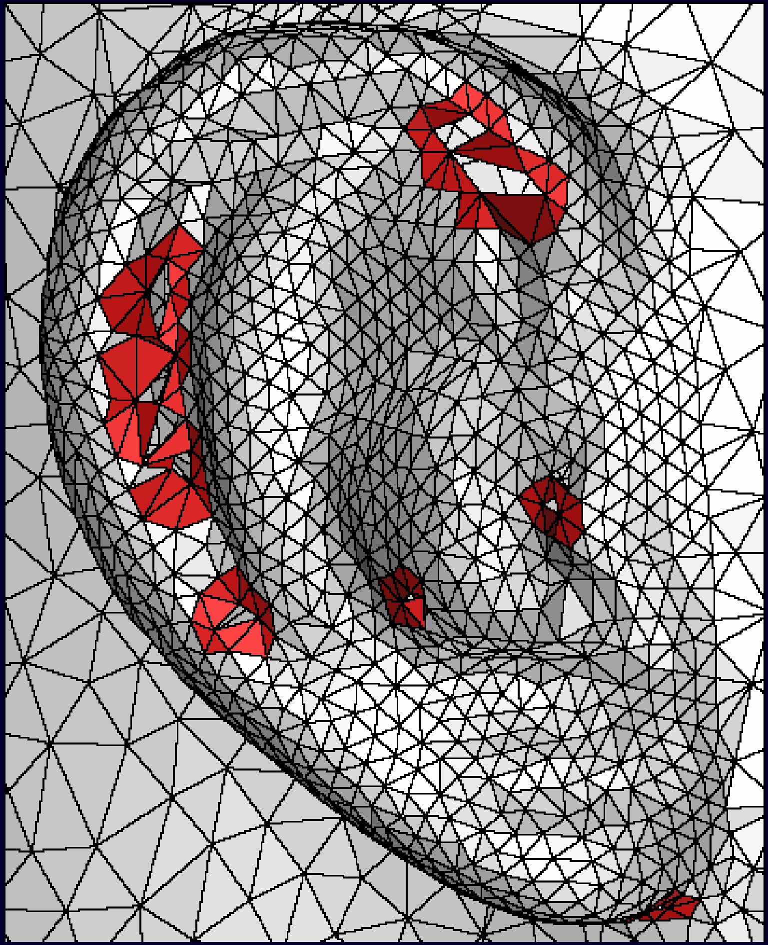
- Compute Voronoi
- Determine candidate triangles
  - Using the cocone
- Remove triangles with free edges
  - *And not on a boundary*
- Extract manifold
- Patch (small) holes

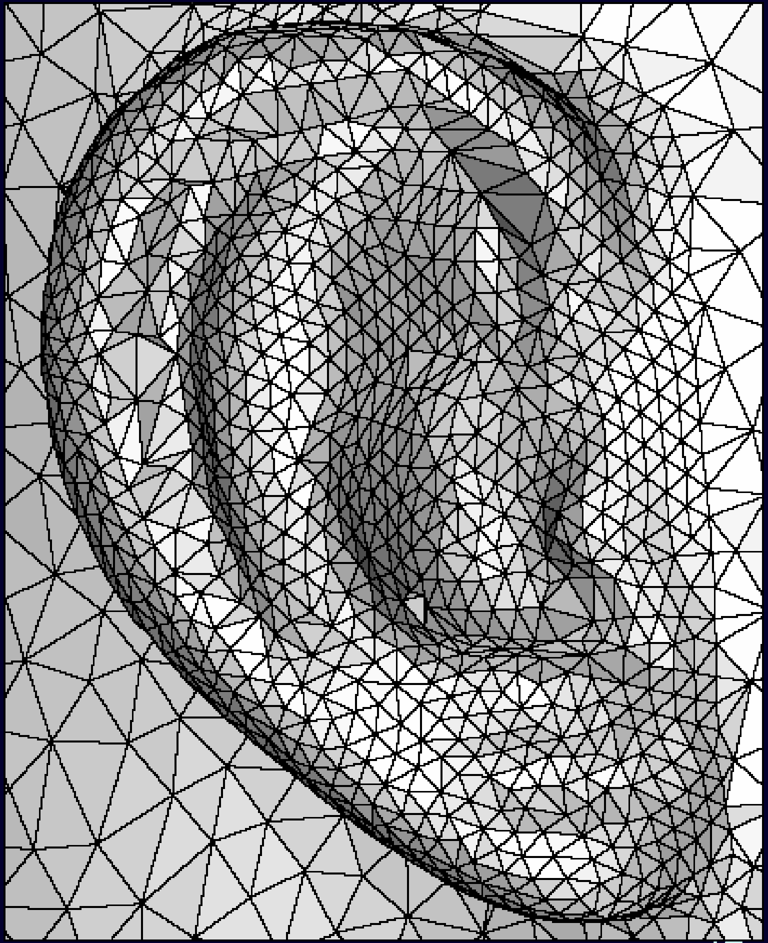


Experimental

- Good settings are
  - $\theta = \pi/8$
  - $\rho=.99$
  - $\alpha=\pi/6$
- Theory says
  - $\theta = \pi/8$
  - $\rho \leq 1.3\varepsilon$
  - $\alpha \leq .14$ radians

Results

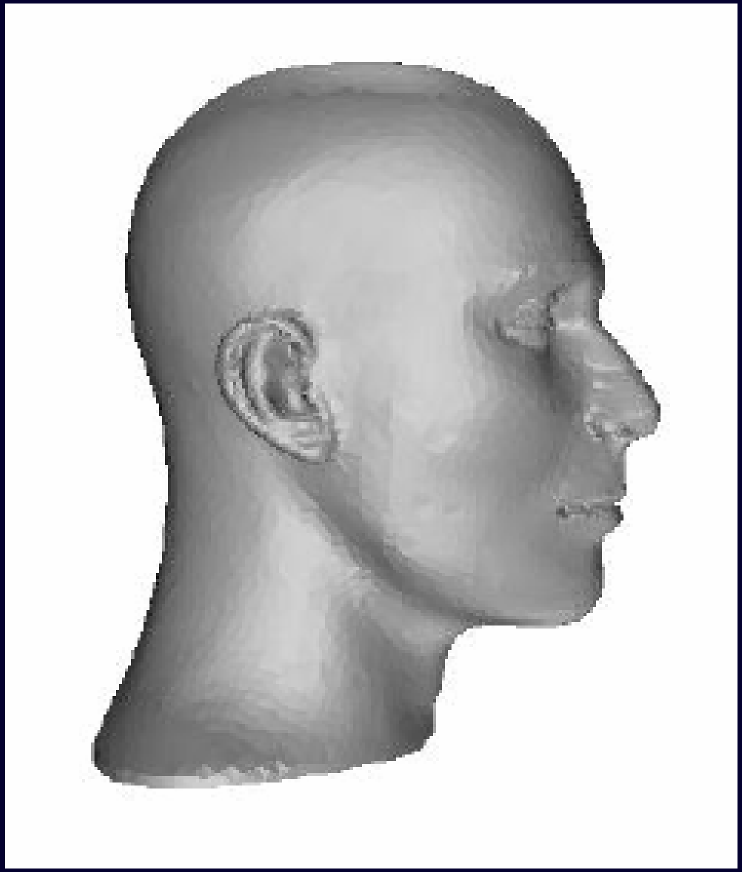


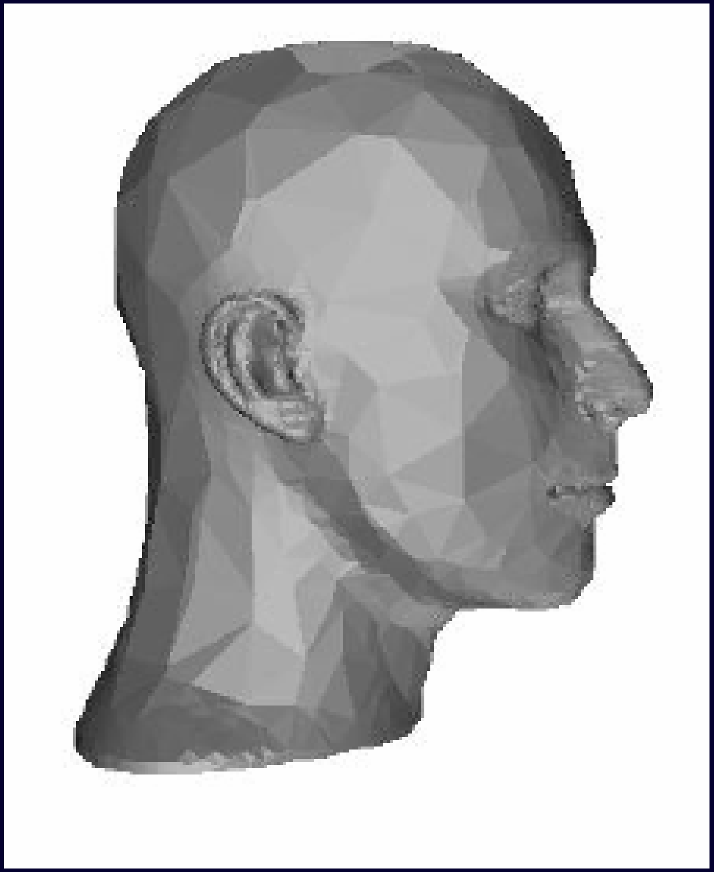


Oversampling

- Voronoi cells too skinny
  - So remove some of the points
- Can preserve poles
  - Since the original estimates are the best
- Don’t have to recompute Voronoi

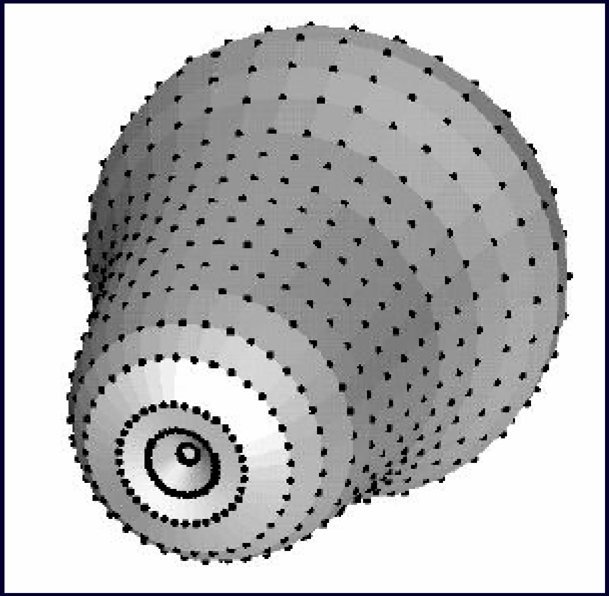
Results





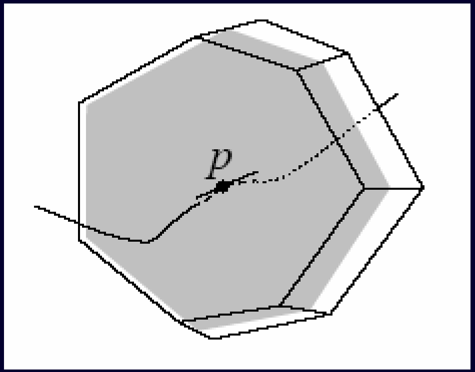
Dimension estimation

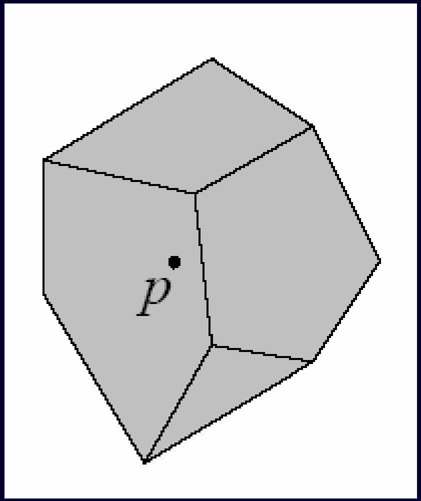
- k-manifold
  - Voronoi cells with d-k large dimensions and k small ones
- They codify this
- Need the stronger sampling condition

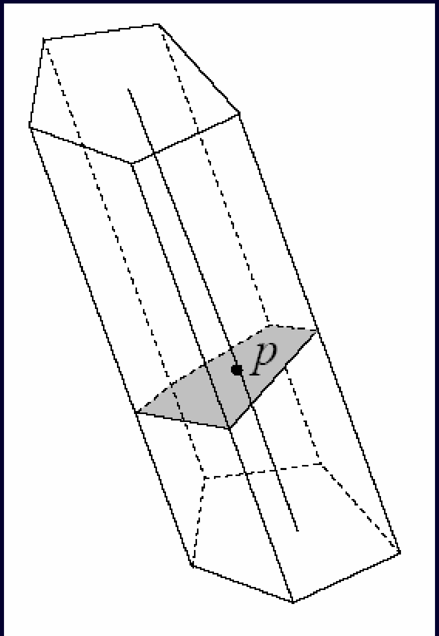


Definitions

- Tangent space
  - Subspace spanned by tangents
  - Dimension of manifold, k
- Normal space
  - dim d-k

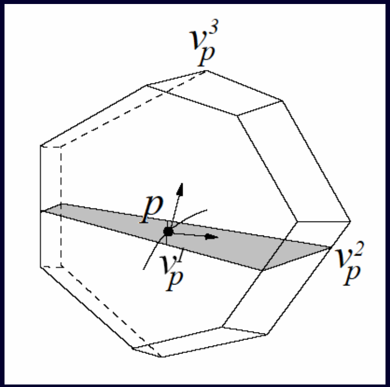


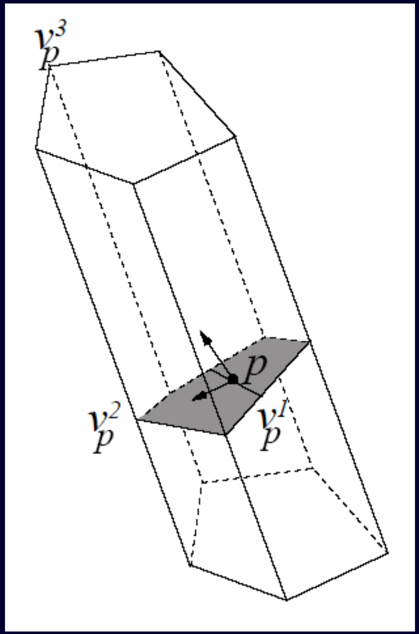




Height

- H_p^i is
  - The size of the ith largest dimension of the voronoi cell
- Compute pole, take orthogonal, compute farthest…





Dimension Estimation

- $p \in P$ is from a manifold of dim k if
  - $H_p^i \geq f(p)$ for $k < i \leq d$
  - $\delta/2\ f(p) \leq H_p^i \leq \varepsilon/(1-\varepsilon)\ \sec(\alpha/2(1+4\sqrt{(d-k)})\ f(p)$ for $1 \leq i \leq k$
    - Basically $\delta/2\ f(p) \leq H_p^i \leq \varepsilon f(p)$
- Find the dimension where H first is small

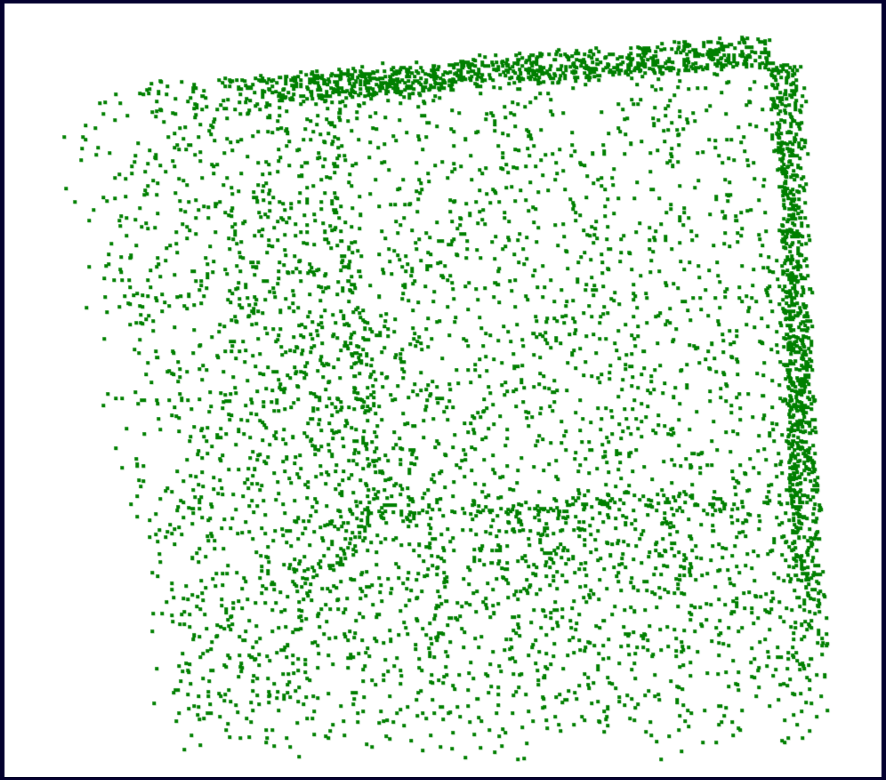
Manifold Computation

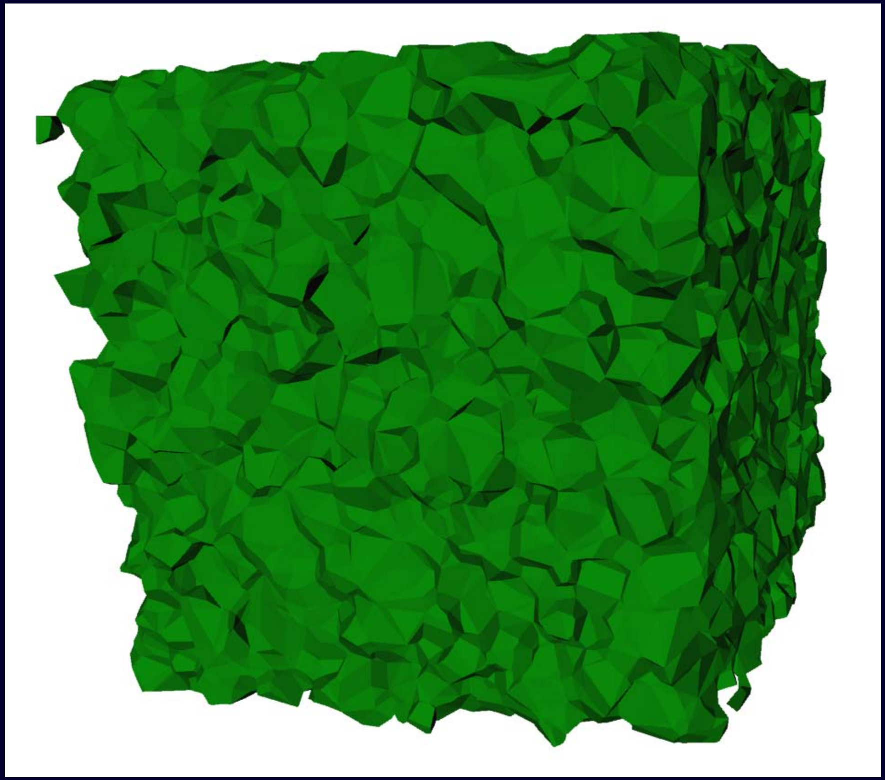
- Compute the dimension of a point
- Extract the duals of the d-k voronoi features which intersect the Cocone

Computing Delaunay

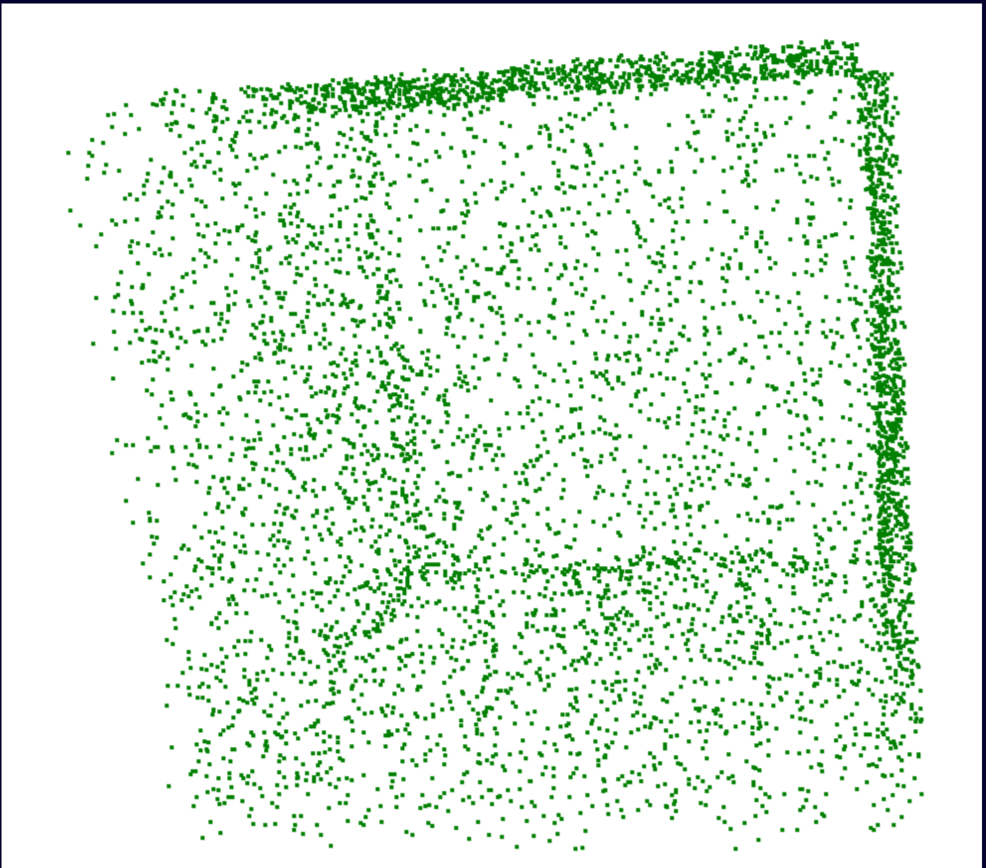
- It is slow compared to other approaches
- Some good news in that direction
  - BRIO
  - CMU result
  - Oct-tree based
- 10^7 or so points in half an hour

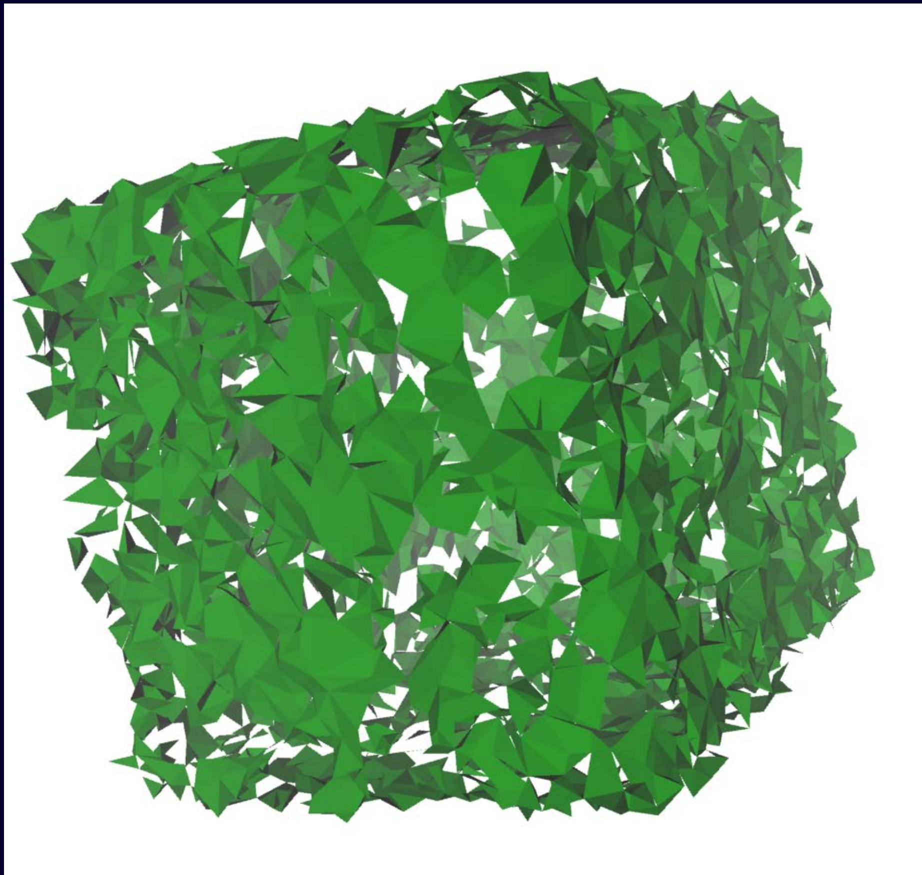
Powercrust .05



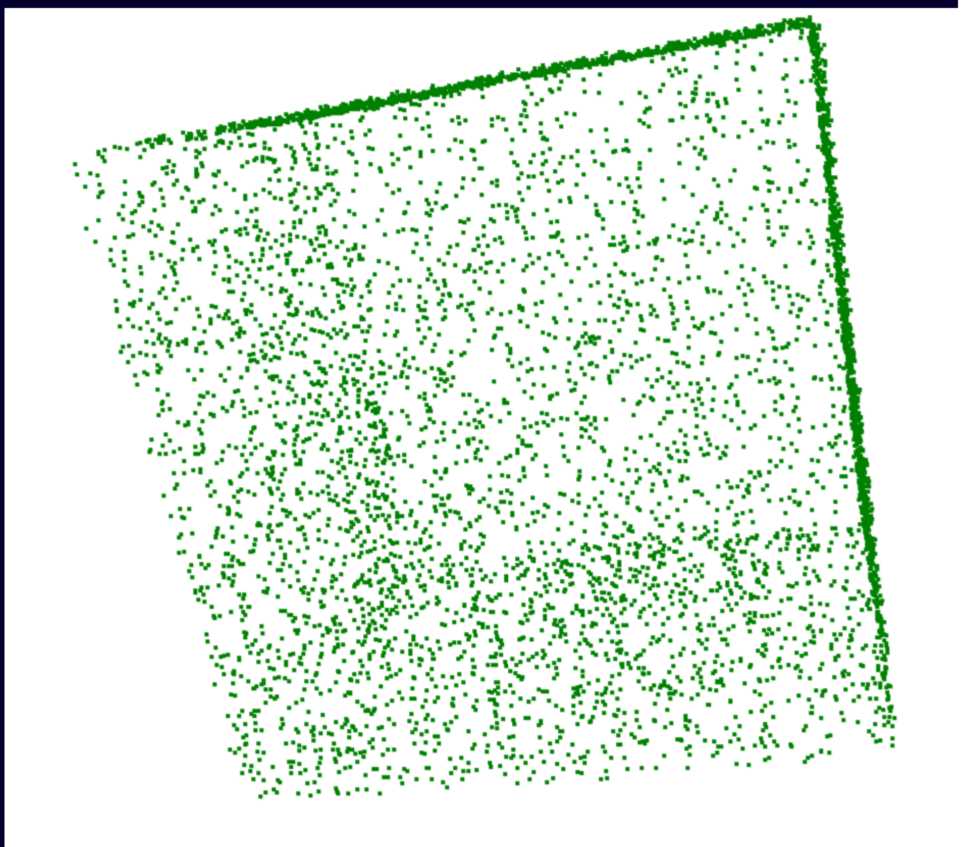


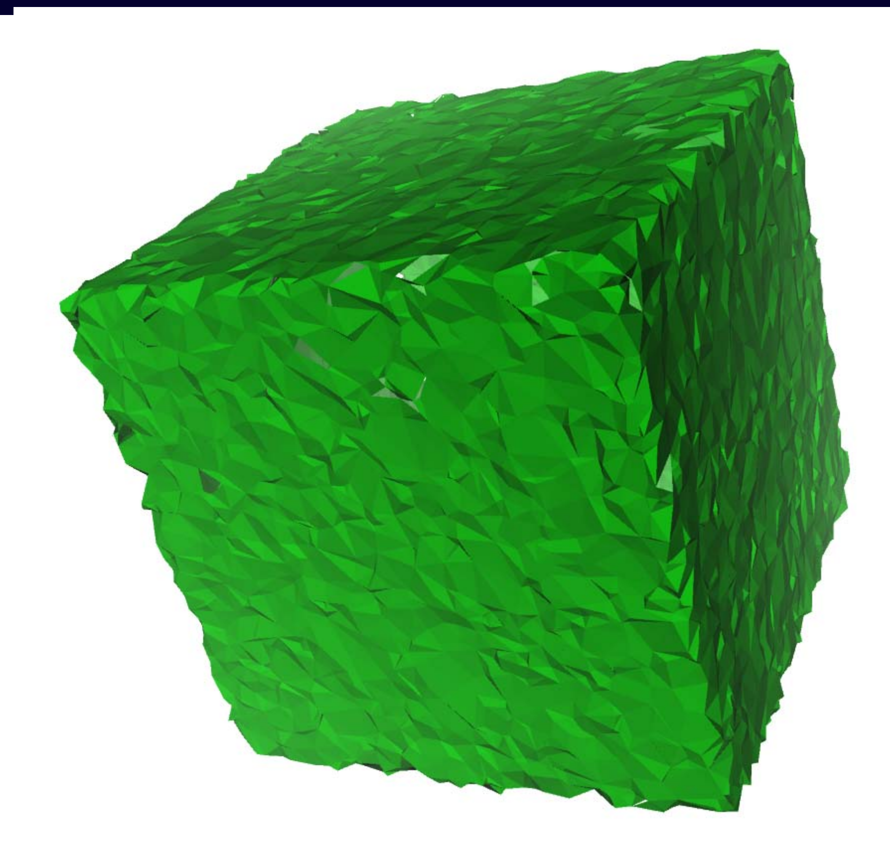
Cocone .05





Cocone .01





0 error

- Cocone, Powercrust