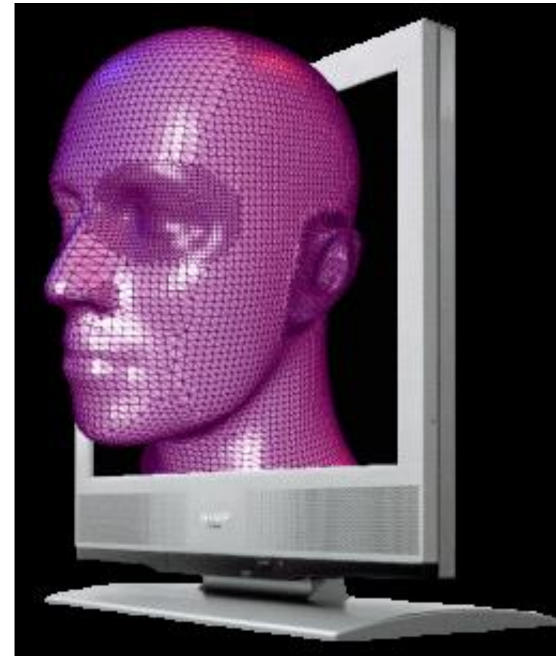
L4 – Direct Surface Reconstruction

- Techniques to generate B-rep of surfaces
  - Direct triangulation
  - Voronoi methods
  - Segmentation based method
  - Adaptive Spherical Cover (ASC) based method

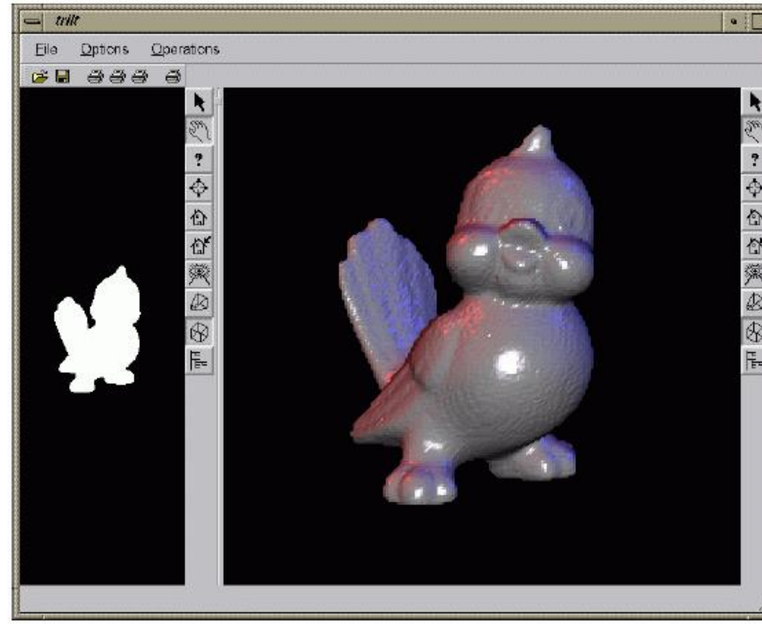
Direct Triangulation

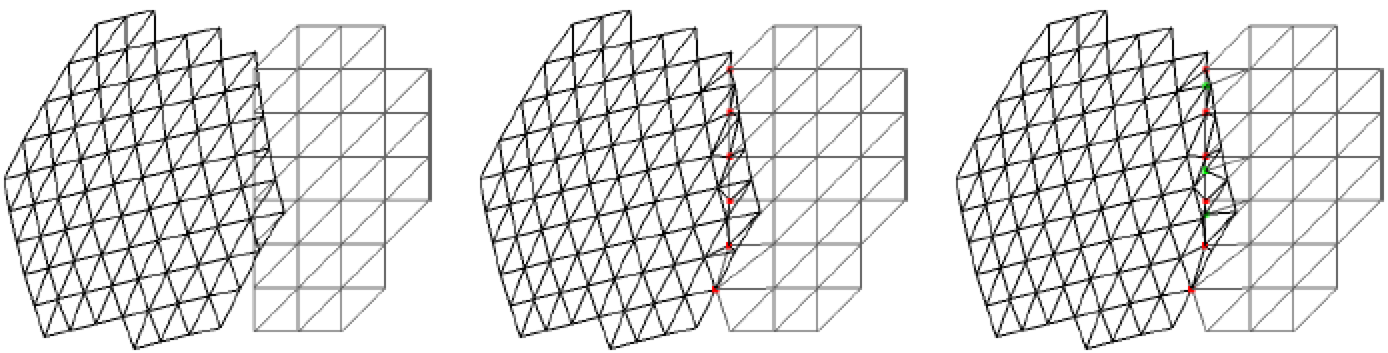


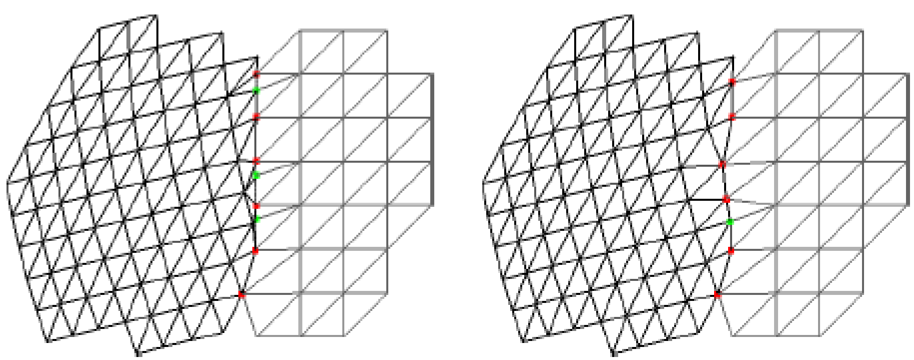
- Using a virtual scanner
- Allow users to rotate the given geometry on the screen in to adjust the optimal viewing direction
- The fact that real 3D scanners usually yield a rather dense cloud of data points which appears as a continuous surface when rendered on the screen
- Rendering sample points into the z-buffer
  - Down-sampled pixels
  - Topology from the neighborhood relation (patch-by-patch)

Direct Triangulation

- By information obtained from z-Buffer (i.e., the height field)

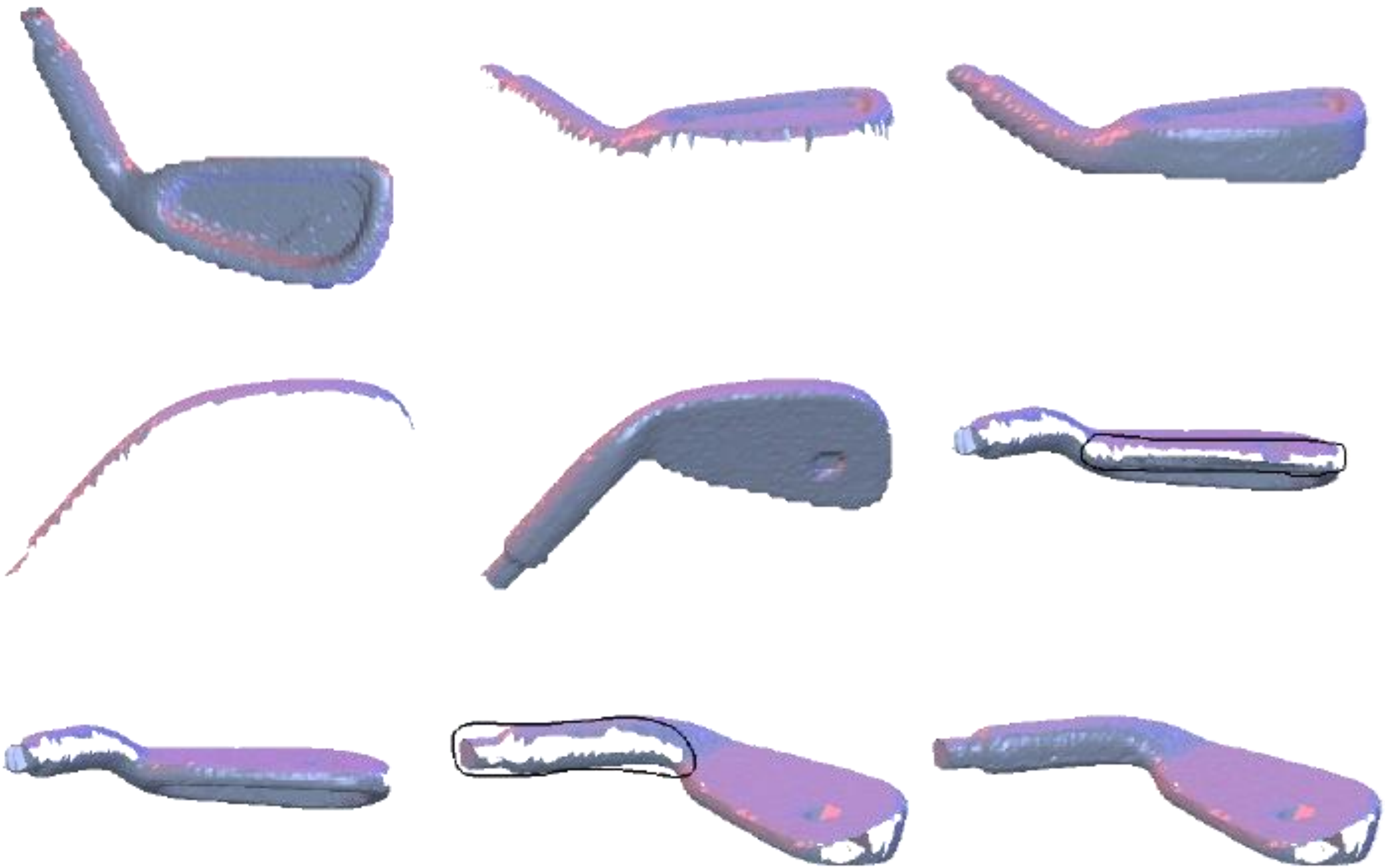






Stitching triangulated surface patches

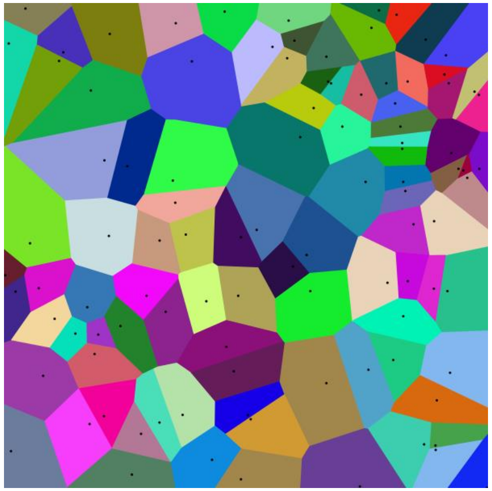
Progressive Surface Reconstruction



Voronoi Method

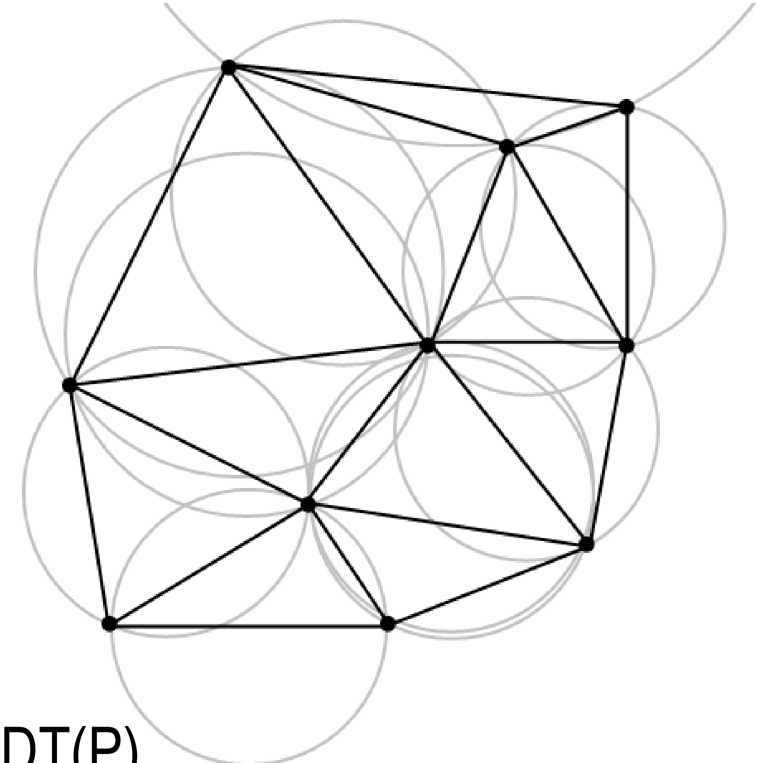
- Using Voronoi partitioning in 3D space
- The motivation is to find the correct topology of the sampled surface even if samples are scattered sparsely
- The proposed schemes typically come with some bound on the minimum sampling density depending on the local surface curvature
- ***Common Feature***: they are theoretically sound by guaranteeing correct reconstruction if the bounds on the sampling density are met

Voronoi Diagram

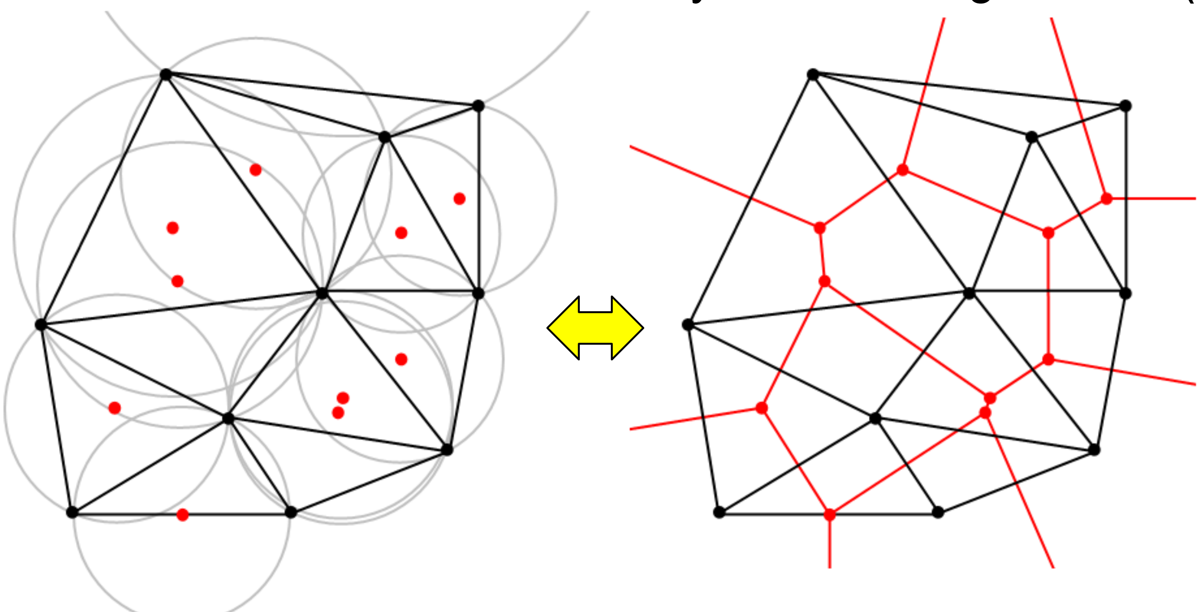


- Definition
  - Let *S* be a set of points in Euclidean space
  - In general, the set of all points ***closer*** to a point **c** of *S* than to any other point of S is the interior of a (sometime unbounded) convex polytope called the ***Dirichlet domain*** or ***Voronoi cell*** for **c**
  - The set of such polytopes tessellates the whole space, and is the ***Voronoi Tessellation*** (and also called ***Voronoi Diagrams***)
- Motivation
  - For 3D surface reconstruction from a set of scattered sample points, Voronoi Diagram gives reference for the topology of surface represented by the points
  - Why? Dual-graph of Delaunay Triangulation

Delaunay Triangulation



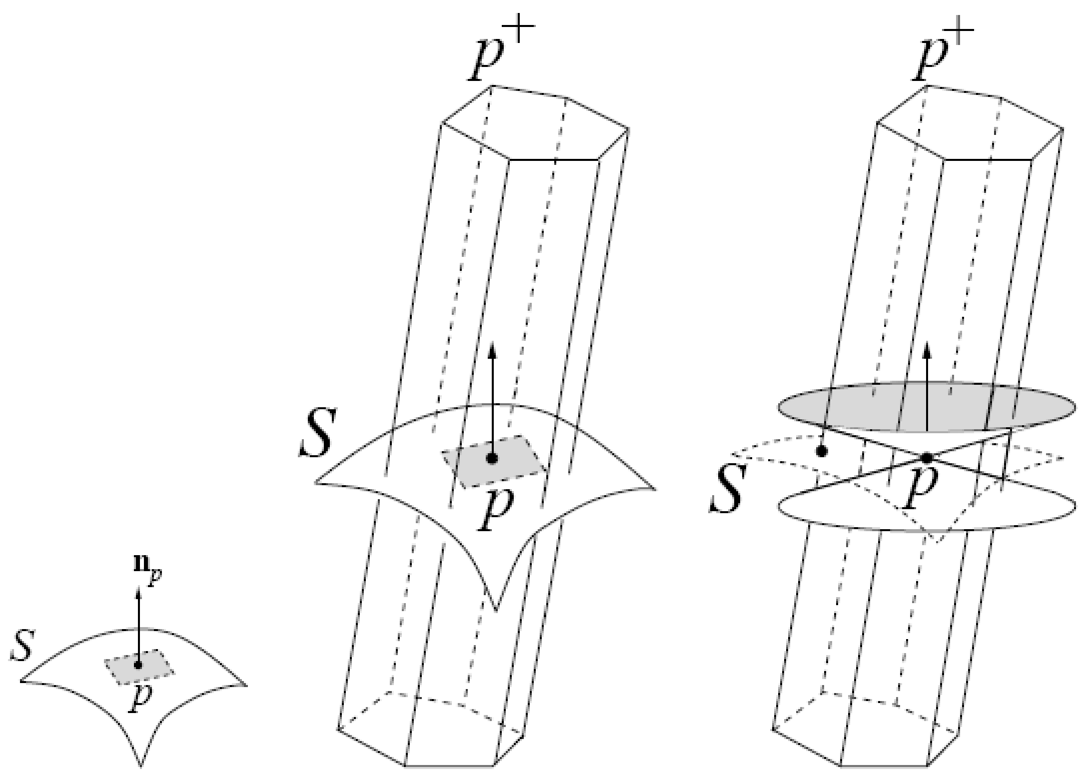
- Definition:
  - A [Delaunay triangulation](#) for a set *P* of points in the plane is a triangulation DT(*P*) such that no point in *P* is inside the circumcircle of any other triangle in DT(P).



This can be further extended into 3D

VD Based Surface Reconstruction

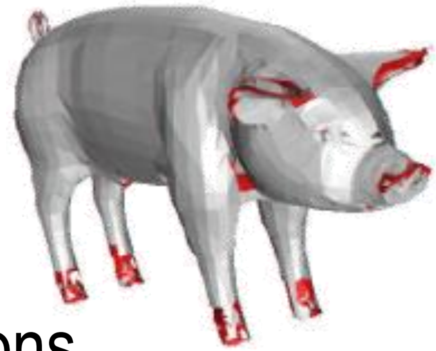
- Voronoi cells are long and thin along the direction of the normals at each sample point if the sample is sufficiently dense
- CoCone of a sample **p**: $C_p = \{y \in V_p \ : \ \angle((y-p), \mathbf{v}_p) \geq \frac{3\pi}{8}\}$



CoCone Algorithm

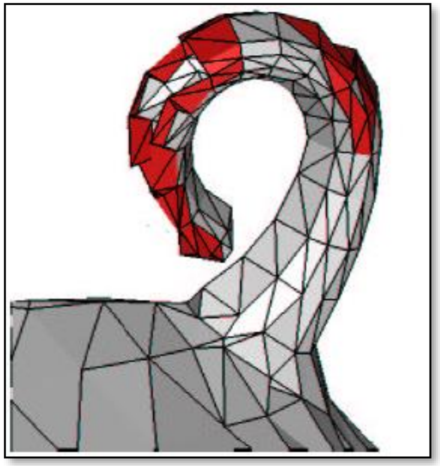
- Each sample chooses a set of triangles from the Delaunay triangulation of the sample **p** whose dual Voronoi edges are intersected by the CoCones defined at the sample
- All such chosen triangles over all samples are called the *candidate triangles*.
- If the sampling density is sufficiently high, these candidate triangles lie close to the original surface *S*
- A subsequent manifold extraction step extracts a manifold surface out of this set of candidate triangles
- This manifold is homeomorphic and geometrically close to *S*

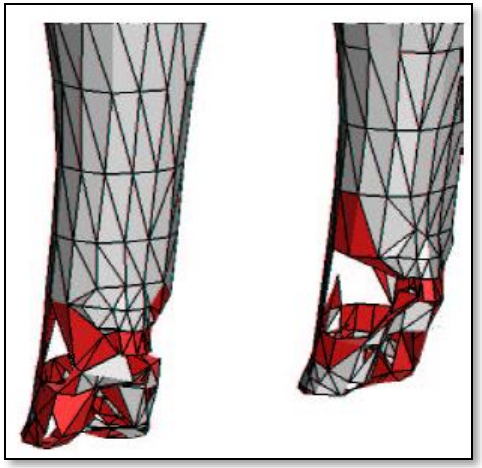
Problems of CoCone Algorithm



- Undesirable triangles near *undersampled* regions
  - The undersampling may be caused by non-smoothness, inadequate sampling or noise
  - The Voronoi cells of these undersampled points are not long and thin along the normals to the surface, can be detected by
    - A ratio condition tests the 'skinniness' of the Voronoi cells
    - A normal condition tests if its elongation matches with those of its CoCone neighbors
- Triangles relating to under-sampled cells are excluded

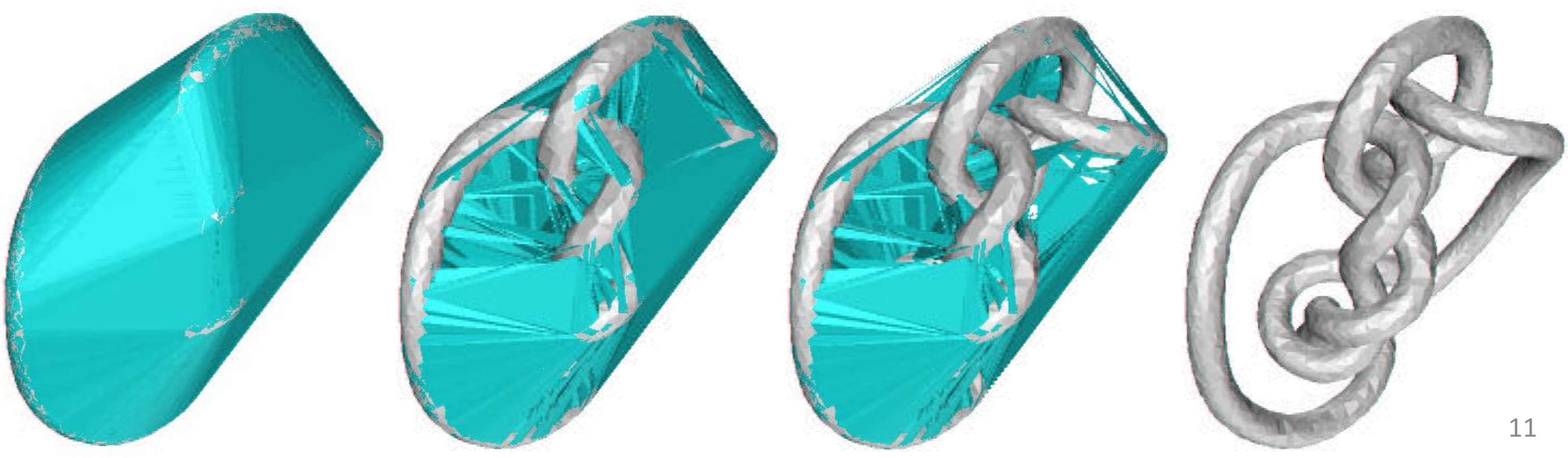
(This modified CoCone algorithm however generate holes)





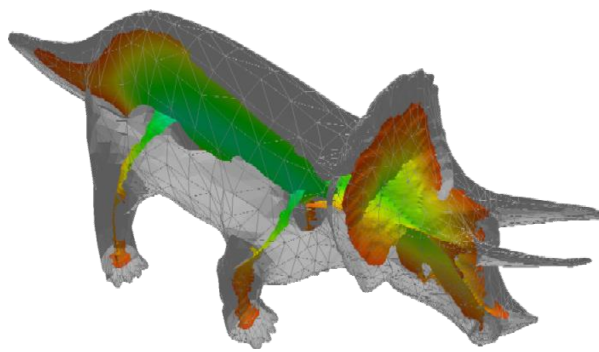
Water-Tight Reconstruction

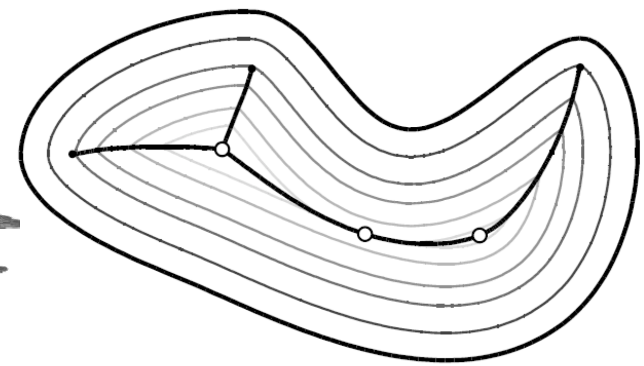
- Overall idea of *Tight-CoCone* is
  - Labeling the Delaunay tetrahedra computed from the input sample as ***in*** or ***out*** according to an initial approximation
  - Peeling off all ***out*** tetrahedra
  - This leaves the ***in*** tetrahedra, the boundary of whose union is output as the water-tight surface



Analysis of VD Based Reconstruction

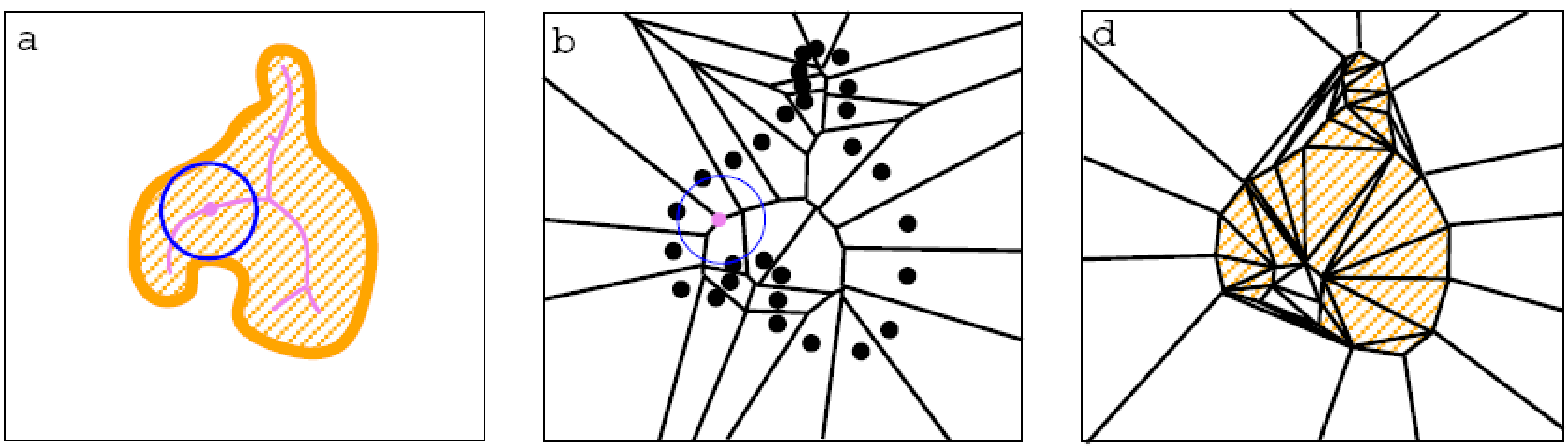
- The medial axis of a surface *S* in 3D is the closure of the set of points which have more than one closest point on *S*
  - Note that Voronoi diagram can be regarded as a discrete form of the medial axis





- The local feature size, $f(\mathbf{p})$, at point **p** on *S* is the least distance of **p** to the medial axis
- A point set *P* is called **an ε-sample of a surface *S*** if every point **p** on *S* has a sample within distance $\varepsilon f(\mathbf{p})$

Analysis of VD Reconstruction (Cont.)



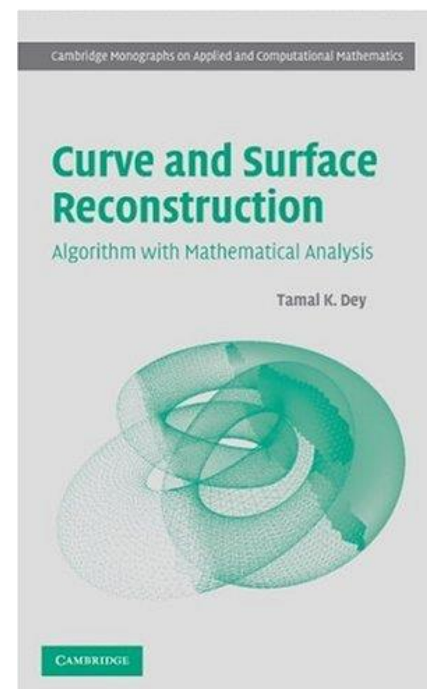
Analysis of VD Reconstruction (Cont.)

- **Theorem:** Let *P* be an ε-sample of a smooth surface *S*, with $\varepsilon \le 0.06$, the *CoCone* algorithm computes a piecewise-linear 2-manifold *N* homeomorphic to *S*, such that any point on *N* is at most $(1.15\,\varepsilon / (1-\varepsilon))\, f(\mathbf{x})$ from some point **x** on *S*.
- Note that
  - $\varepsilon \le 0.06$ is the condition for homeomorphic
  - The geometric error-bound is: $(1.15\,\varepsilon / (1-\varepsilon))\, f(\mathbf{x})$
- Reference Book

  Curve and Surface Reconstruction

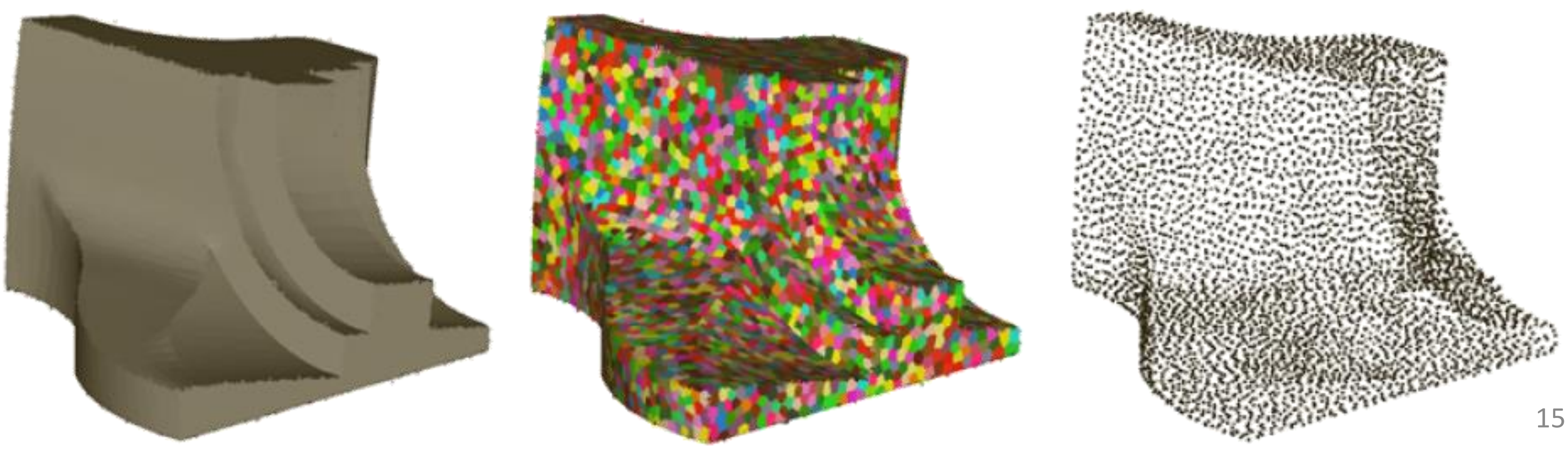
  – Algorithm with Mathematical Analysis

  By Tamal K. Dey



Point Segmentation for Reconstruction

- VD based method fails when point num becomes >100k
- Method for simplifying the points is needed
- Clustering is performed by applying the *Centroidal Voronoi Diagram* (CVD) on the surface
  - CVD is a special Voronoi Diagram where the site point (seed) of each Voronoi cell is located at the centroid position



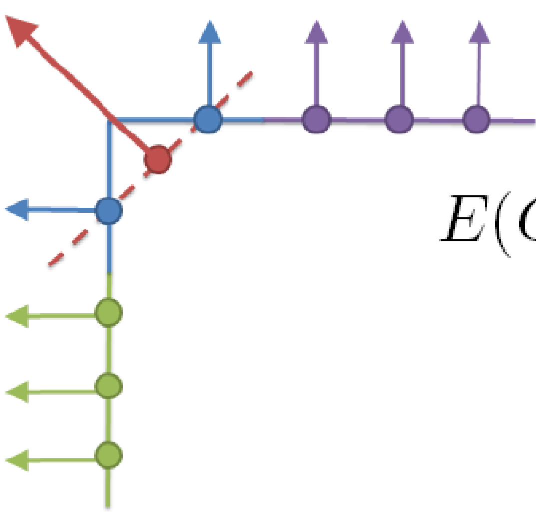
Clustering for Segmentation

- Clustering is driven by minimizing the discrete energy terms
  - Clusters should maintain a disk-like shape

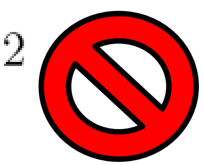
$$E_{dist}(x) = \|x - p_i\|^2$$

  - The distribution of clusters should enable their proxies to best approximate the shape of the given model

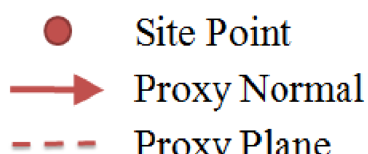
$$E_{shape}(x) = \|(x - p_i) \cdot n_x\|^2$$



$$E_P^S(x) = \|(x - p_i) \cdot n_{p_i}\|^2$$



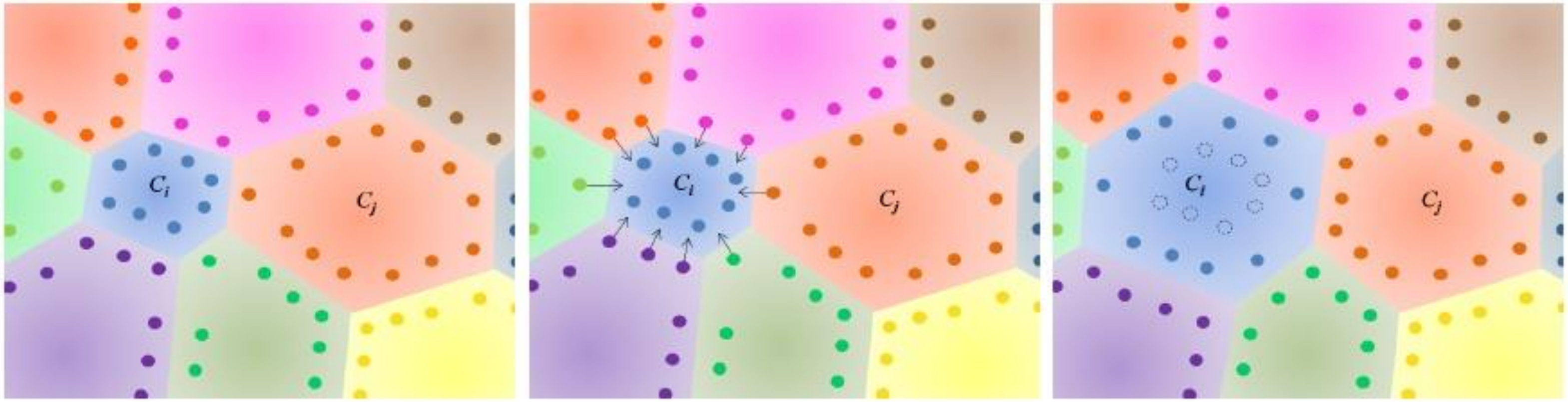
$$E(C_i) = w_1 \sum_{x \in C_i} E_{dist}(x) + w_2 \sum_{x \in C_i} E_{shape}(x)$$



$$E_{global} = \sum_i E(C_i)$$

Optimization for Clustering

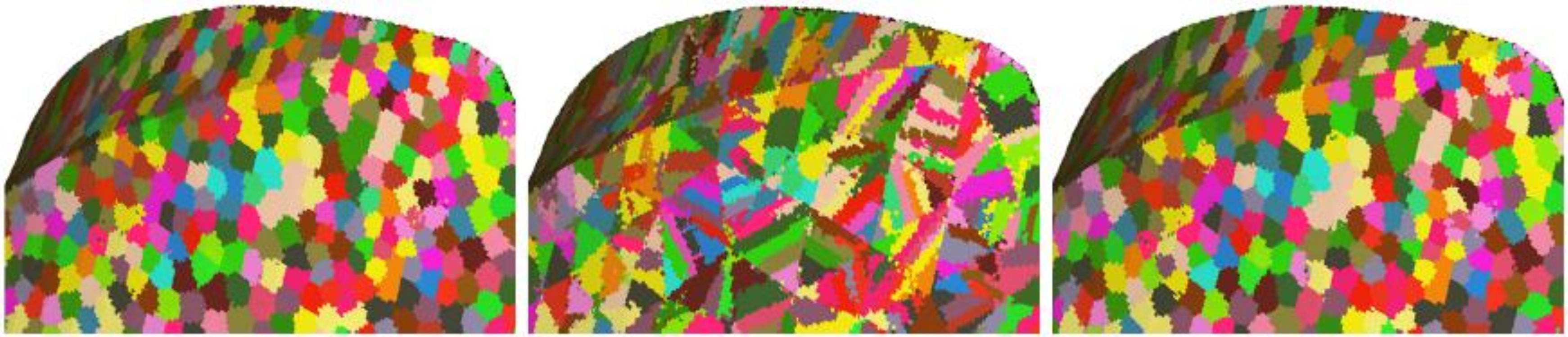
- Lloyd's Algorithm for CVD, which is iteratively performed by the following two steps
  - Compute the centroid of each cluster as the representative point of cluster
  - Form the new partition by assigning each data point to its closest representative point in Euclidean space
- Can be speed up by local updating



Local Update for Cluster Optimization

Algorithm 3: Clustering Optimization

```
repeat
    for each cluster C_i in parallel do
        Update the site point to the average position of all points in C_i;
        Find the boundary points in C_i;
    end
    for each cluster C_i in parallel do
        for each boundary point x_b ∈ C_i do
            for neighbors x_j ∈ C_j of x_b do
                if moving x_b to C_j reduces the energy then
                    Update the cluster ID of x_b;
                end
            end
        end
    end
until the change of E_global is less than 1% ;
```

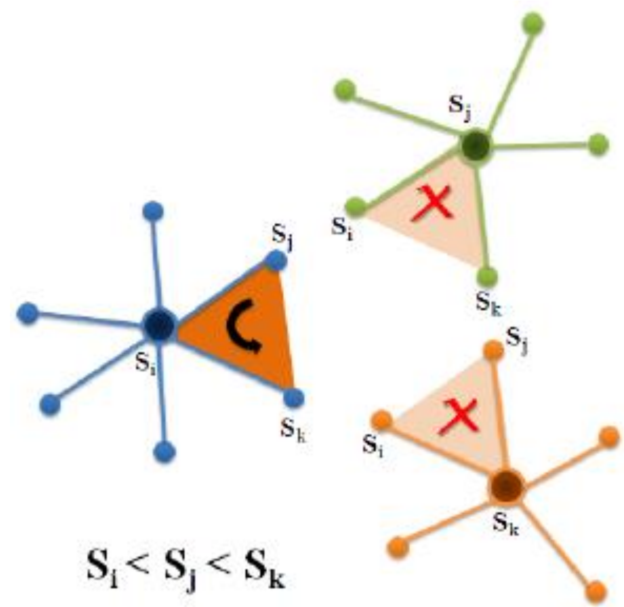


$w_1 = 1, w_2 = 0$ $w_1 = 0, w_2 = 1$ $w_1 = 0.1, w_2 = 0.9$

- After iteration, one Hermite data is retained for each cluster
  - Using the site point as the down-sampled point
  - Using the normal at the closest sample point to the site point
- The down-sampled points can then be triangulated

Topology Reconstruction

- To obtain the connectivity information of site points, we first build a neighboring cluster table
  - Constructed by checking the boundary samples
  - The neighboring site points of every site point *s* are then projected onto the tangent plane and sorted radially according to the angles to a reference vector
  - Triangles are only created if the index of *s* among them is the smallest
  - Cannot ensure water-tight!
  - Alternative: *Tight-CoCone*



Topology Reconstruction (cont.)

Algorithm 4: Local Triangulation

for each site point s_i **in parallel do**
    for each neighboring site point s_j **do**
        Project points to tangent plane forming $\vec{t_j}$ by Eq.(5.1);
    end for
    $\vec{v_r} \leftarrow \vec{t_o}$;
    $\theta_0 \leftarrow -1$;
    for $j = 1$ to $(neighNum - 1)$ **do**
        Compute θ_j by Eq.(5.2);
    end for
    Sort s_j according to θ_j;
    for each pair of consecutive neighboring points s_j, s_k **do**
        if index of s_i is the smallest **then**
            Create triangle $\triangle s_j s_i s_k$;
        end if
    end for
end for

Sharp Feature Reconstruction

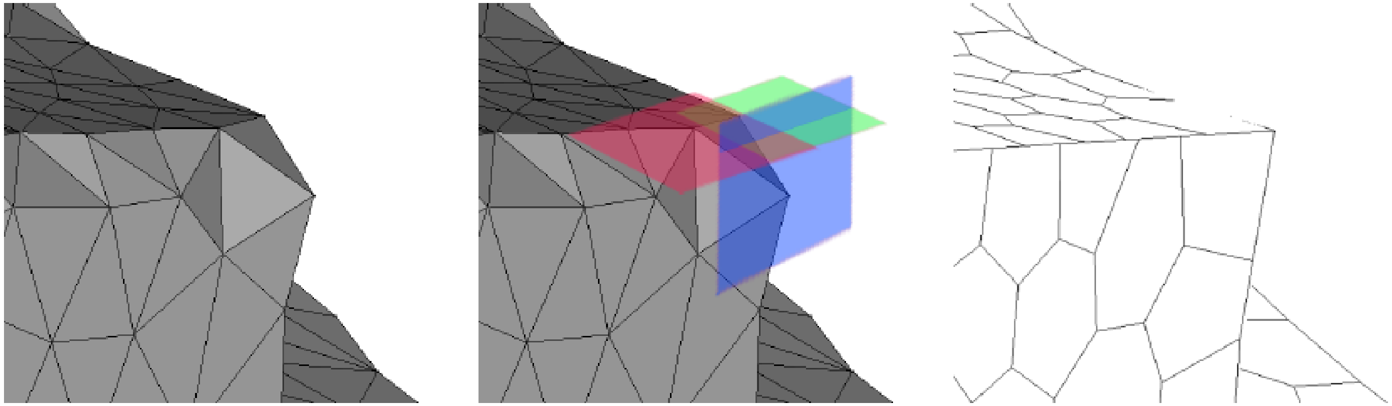
- Compute the dual-graph of current triangular mesh
- Position of the new vertex is computed by minimizing the *Quadratic Error Function* (QEF)

  $$E(x) = \sum_i (n_i^T x - d_i)^2$$

  by solving

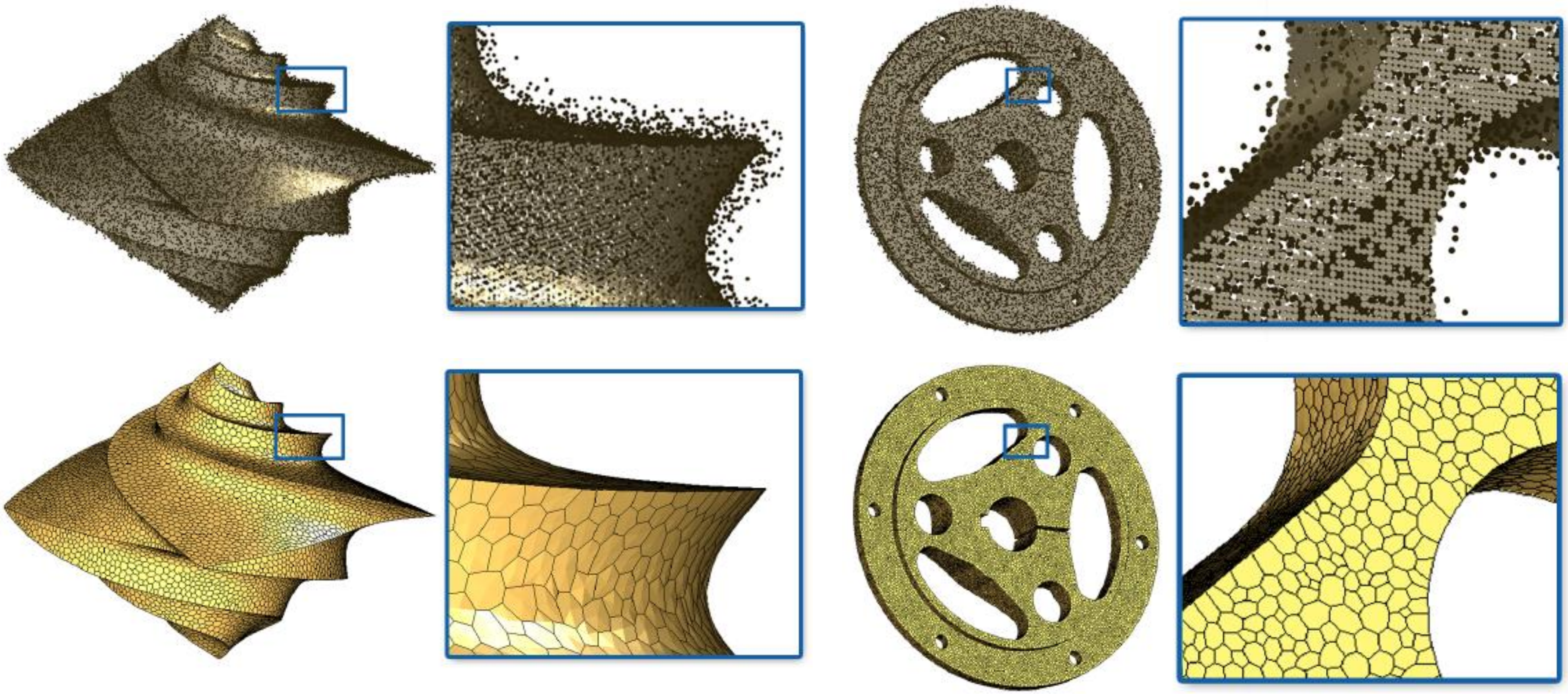
  $$(\sum_i n_i n_i^T)x = (\sum_i n_i d_i)$$

- To be robust, the *Singular Value Decomposition* (SVD) will be used



Robust Surface Reconstruction

- Work together with the robust normal estimation



Hoi Sheung, and Charlie C.L. Wang, "Robust mesh reconstruction from unoriented noisy points", ACM Symposium on Solid and Physical Modeling 2009, pp.13-24, San Francisco, California, October 5-8, 2009.

Adaptive Spherical Cover

- Every point is assigned with a weight: $w_i = \frac{1}{k}\sum_{j=1}^{k}\|\mathbf{p}_i - \mathbf{p}_j\|^2$
- Also preliminary normals by the covariance based method
  - Orientation is not important at this moment
- Generate *m* spheres (*m* < *n*) by starting with all *uncovered* points
  - Random select an uncovered point as the center
  - For each sphere if the radius *r* was known

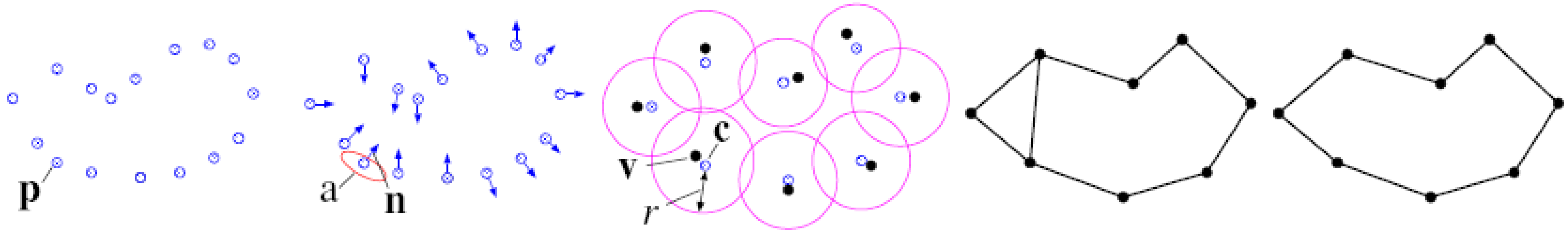
$$Q_{\mathbf{c}_i,r}(\mathbf{x}) = \sum_j w_j G_\sigma(\|\mathbf{p}_j - \mathbf{c}_j\|)(\mathbf{n}_j \cdot (\mathbf{x} - \mathbf{p}_j))^2$$

$$G_\sigma(\rho) = \begin{cases} exp(-8(\rho/\sigma)^2), & |\rho| \in [0, \sigma/2] \\ 16(1-\rho/\sigma)^4/e^2, & |\rho| \in (\sigma/2, \sigma] \\ 0. & |\rho| \in (\sigma, \infty] \end{cases} \qquad \sigma = 2r$$

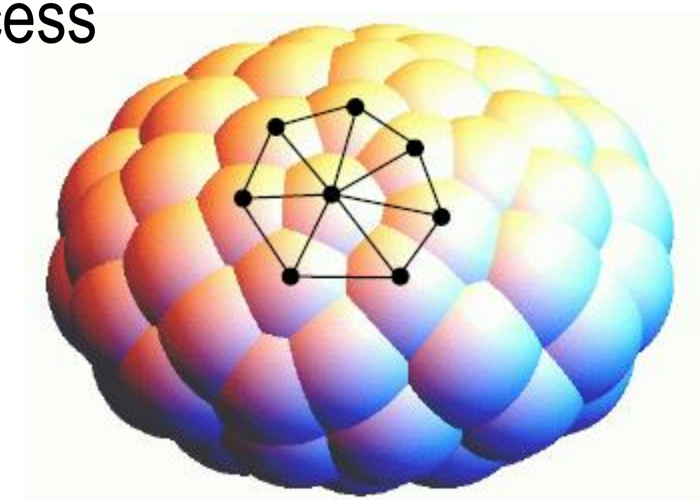
$$\partial Q_{\mathbf{c}_i,r}(\mathbf{x})/\partial\mathbf{x} = 0 \xrightarrow{\text{SVD}} \mathbf{x}_{\min} \xrightarrow{\text{Determine } r} Q_{\mathbf{c}_i,r}(\mathbf{x}_{\min}) = (\varepsilon L)^2 \qquad \varepsilon = 10^{-5}$$

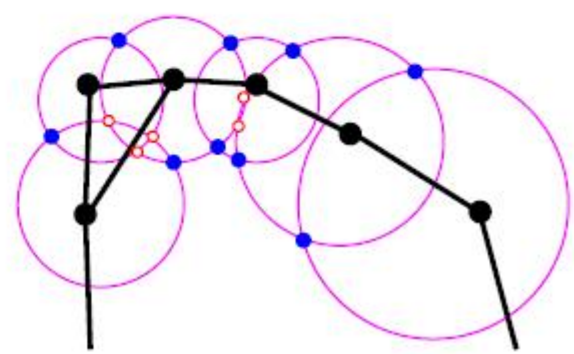
L is the length of the main diagonal of the bounding box of the whole point set *S*

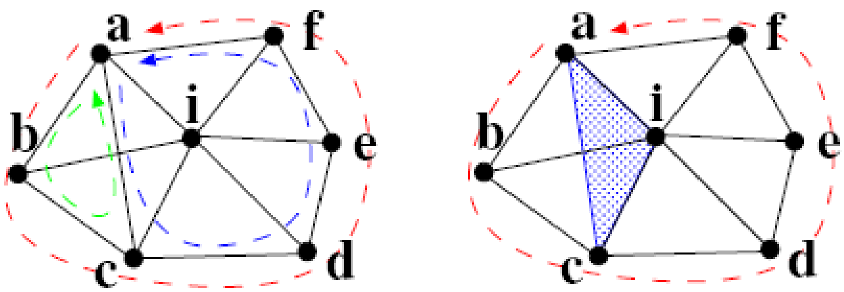
  - Check if $\boldsymbol{x}_{\min}$ is a good auxiliary points; if not, simply assign the center as aux.
  - Projecting points inside sphere and *exclude* pnts NOT inside 2D convex hull



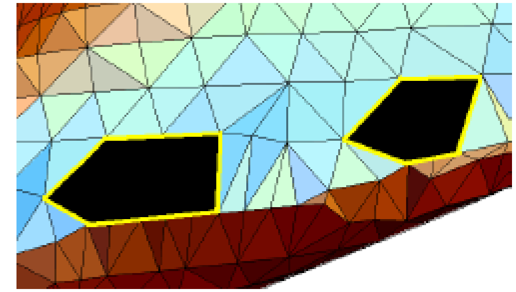
- Triangle {$\mathbf{v}_i$, $\mathbf{v}_j$, $\mathbf{v}_k$} is added if there exist two intersection points of three spheres associated with $\mathbf{v}_i$, $\mathbf{v}_j$, $\mathbf{v}_k$ and at least one of the intersection point is not contained inside other spheres of the cover
- This is a subset of the so-called *nerve complex*
- Cleaning process is needed

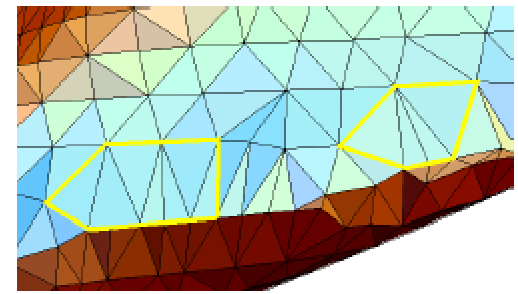


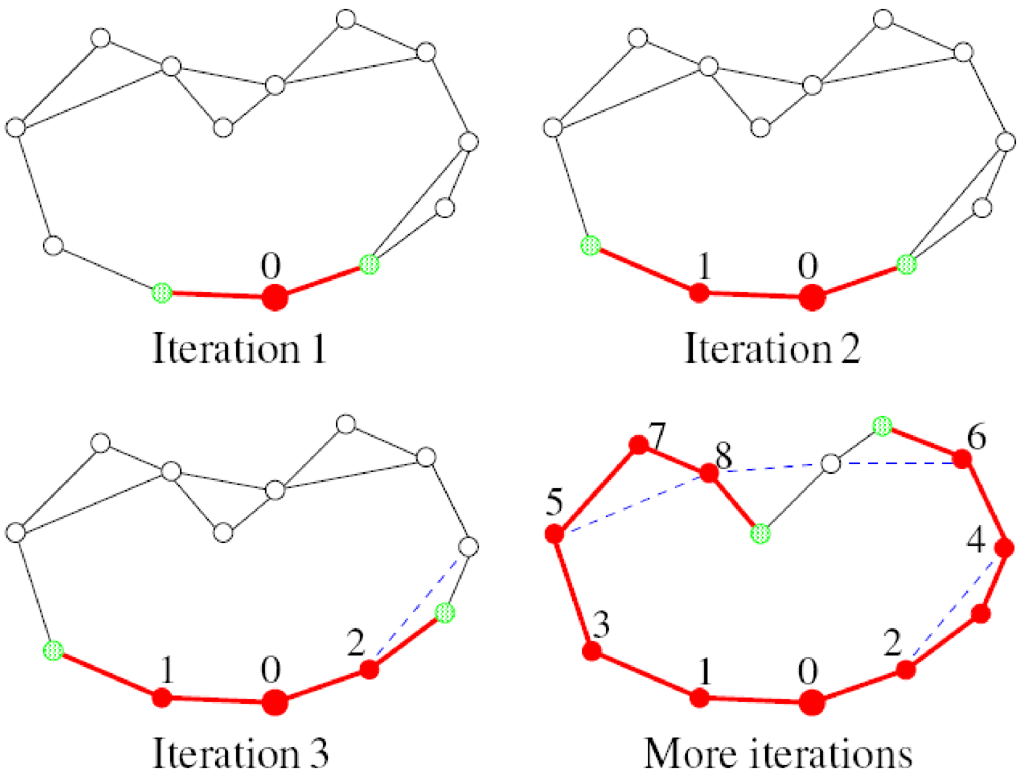


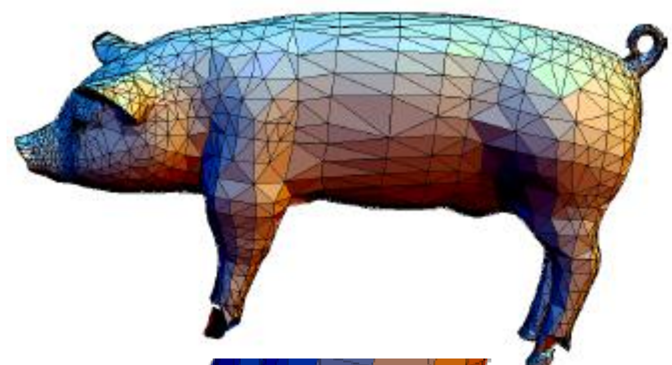


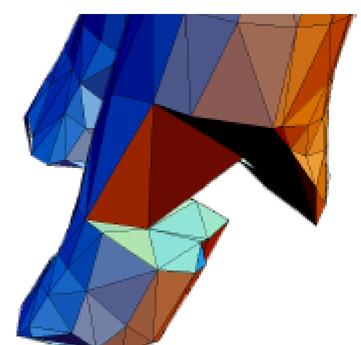
Left: three possibilities to choose a disk-shaped 1-ring neighborhood for vertex **i**. Right: redundant triangle $\{\mathbf{a},\mathbf{i},\mathbf{c}\}$ is detected after the minimum curvature disk is selected.





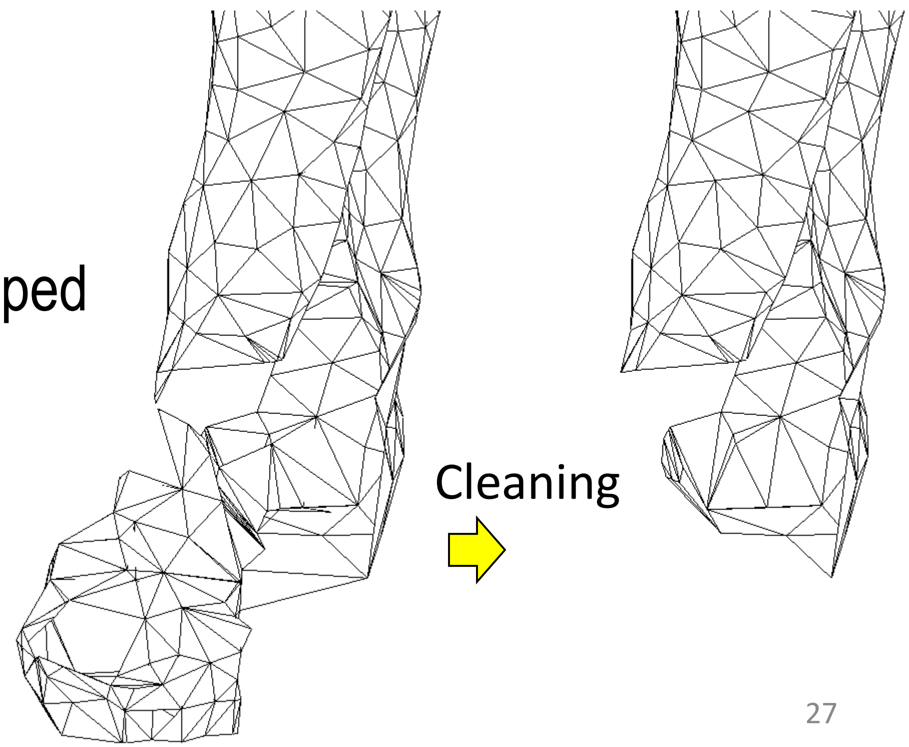






Adaptive Spherical Cover (cont.)

- Auxiliary points are triangulated
- Two-manifold mesh surface – by a cleaning process
- Problematic in the regions with very spare points and the sparseness is *anisotropic*
- The connectivity between regions is very important
  - otherwise, normals can be flipped
  - How to avoid breaking the sphere connectivity of ASC in anisotropic sparse regions?



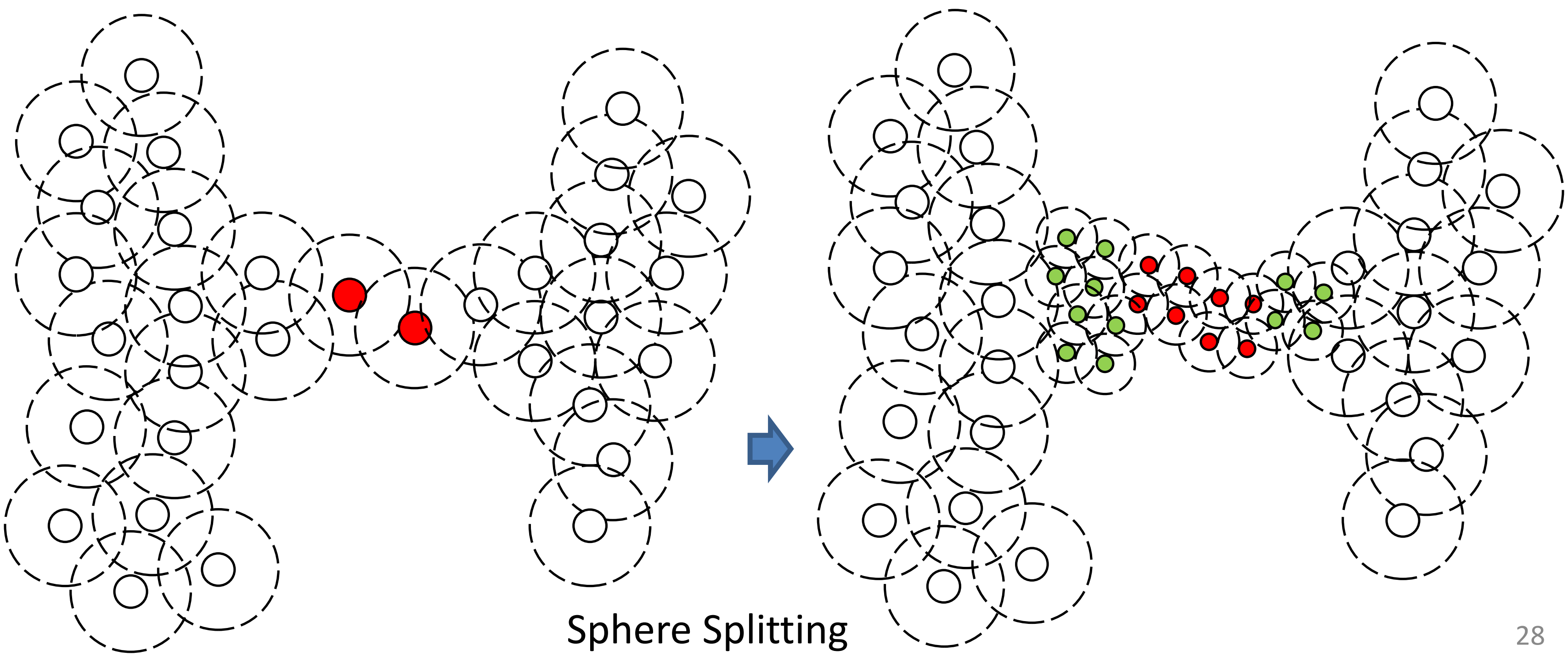
Modified Adaptive Spherical Cover

- Identify such regions by eigen values of the voting tensor

$$F_{\mathbf{c}_i} = \sum_j (\mathbf{c}_j - \mathbf{c}_i)(\mathbf{c}_j - \mathbf{c}_i)^T$$

- If $|\lambda_1| > \mu|\lambda_2|$, is considered as an anisotropic region

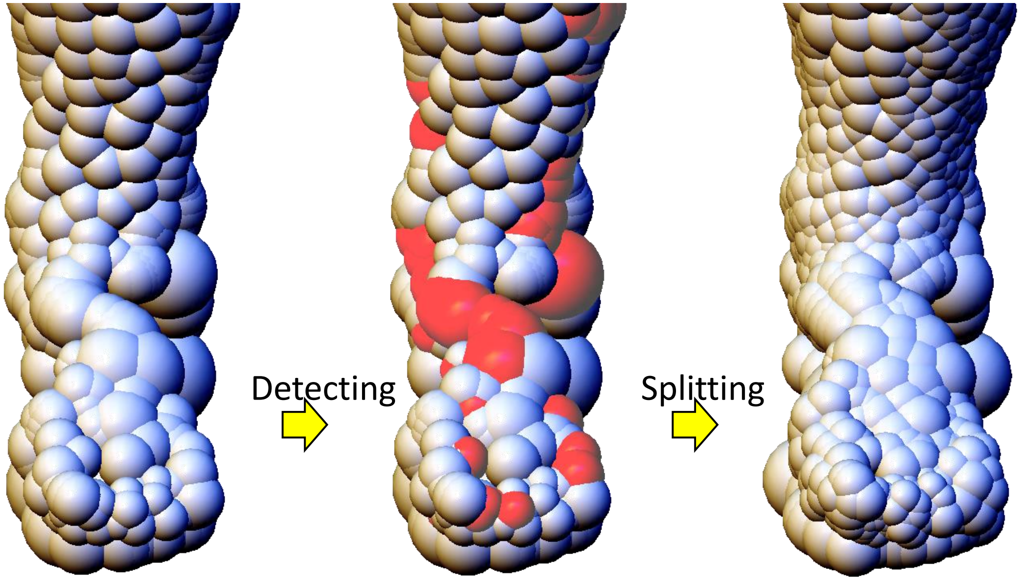
$$\mu = 3.0$$

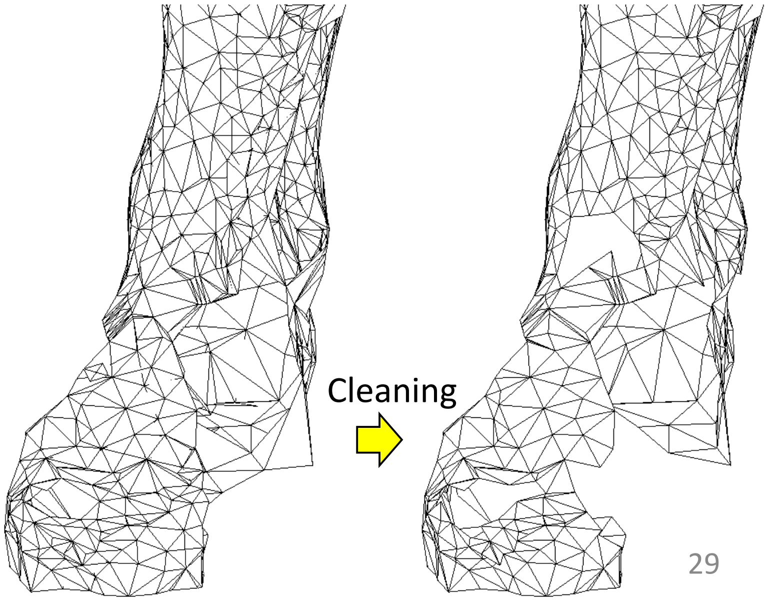


Sphere Splitting

Modified ASC (cont.)

- Splitting spheres in the anisotropic region
- Redistributing spheres on the plane defined by preliminary normals
- Along the direction perpendicular to the thin features





Orienting Unorganized Points

- Triangulating the auxiliary points, get a *rough mesh surface*
- An approximation of the surface represented by *points*
- How to assign normal vectors for points in S?
  - *Direct transfer*: assigned by the closest point's normal
  - Option 1: *Direct flipping* – flipped by the closest point

$$\mathbf{n}_i = \mathbf{n}_{c_{p_i}}$$

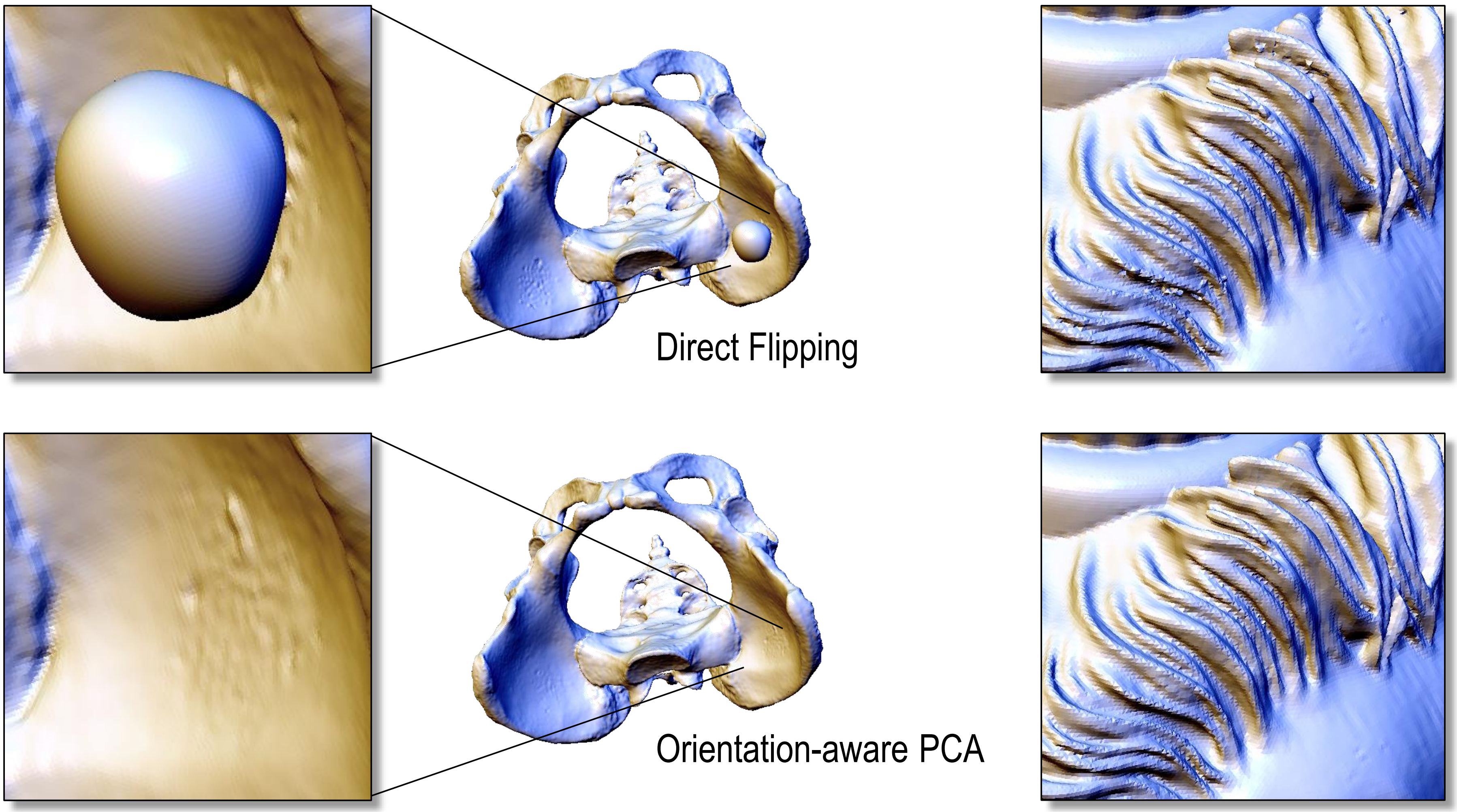
  - Option 2: *Orientation-aware PCA*, only including points that

$$\mathbf{n}_i = -\mathbf{n}_i \qquad \text{if } \mathbf{n}_{c_{p_i}} \cdot \mathbf{n}_i < 0$$

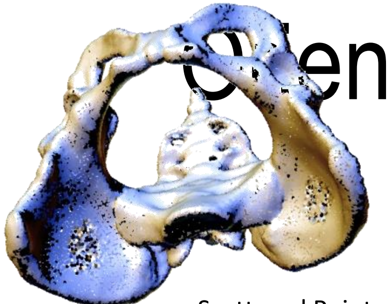
in a new run of covariant *Principal Component Analysis* (PCA)

$$\mathbf{n}_{c_{p_j}} \cdot \mathbf{n}_{c_{p_i}} \geq 0$$

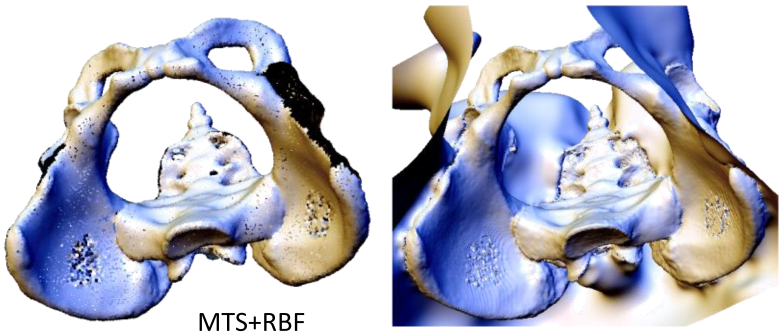
Orienting Unorganized Points (cont.)



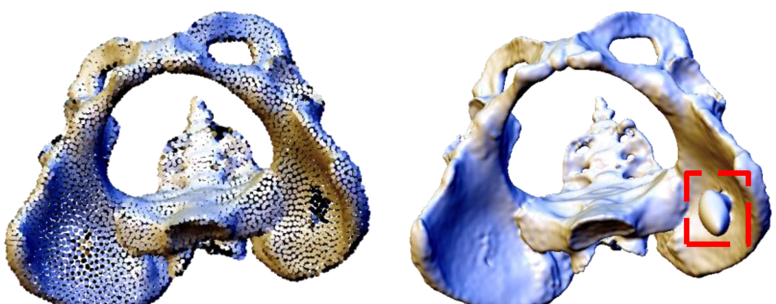
Orienting Unorganized Points for Surface Reconstruction



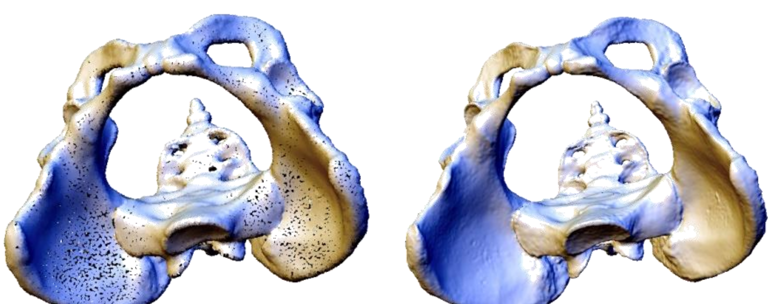
Scattered Points (50.7k)



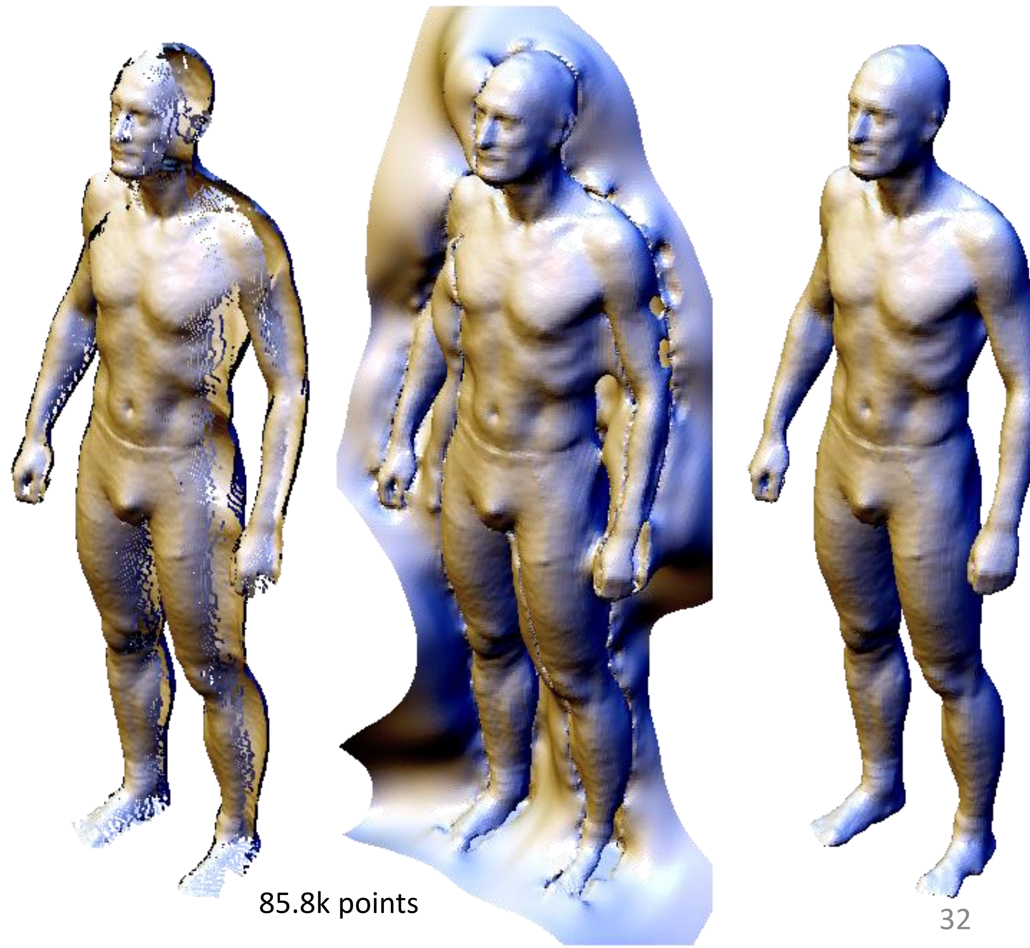
MTS+RBF



Cons+RBF

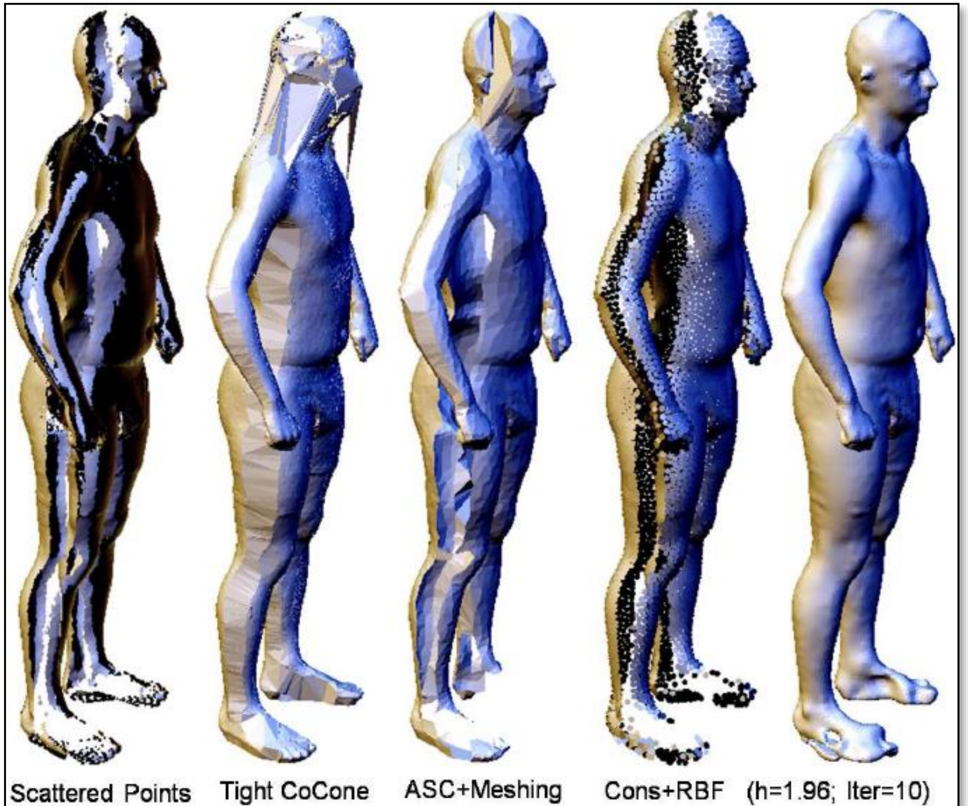


ORT+RBF

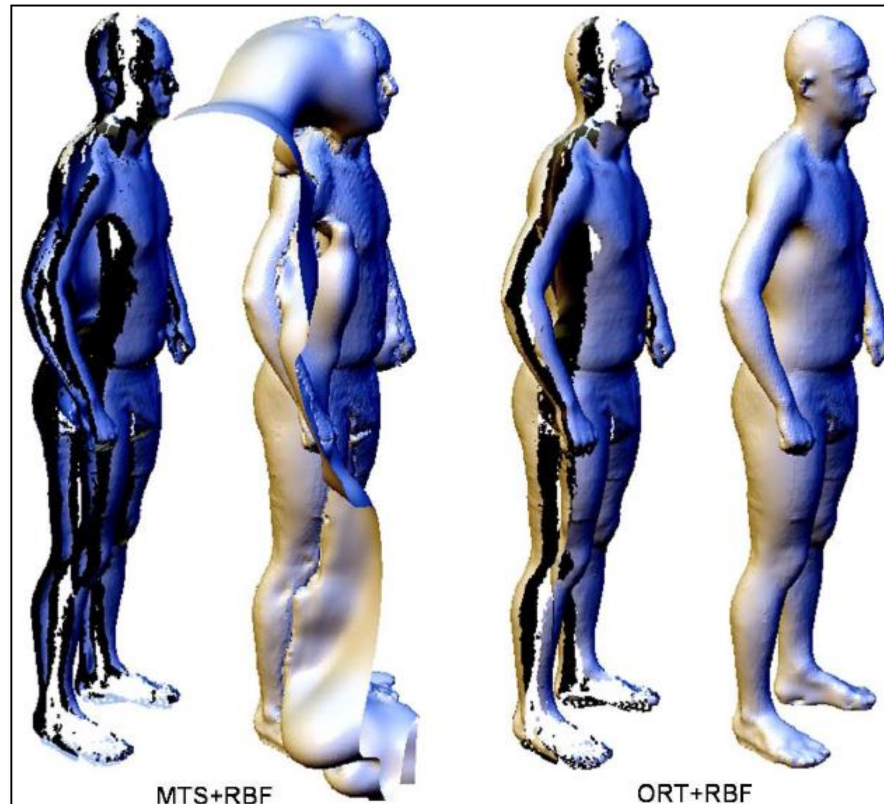


85.8k points

For Human Model Reconstruction



170k points



Shengjun Liu, and Charlie C.L. Wang, "Orienting unorganized points for surface reconstruction", Computers & Graphics, Special Issue of IEEE International Conference on Shape Modeling and Applications (SMI 2010), vol.34, no.3, pp.209-218, Arts et Metiers ParisTech, Aix-en-Provence, France, June 21-23, 2010.