Voronoi-based Variational Reconstruction of Unoriented Point Sets

AYUSHI SINHA

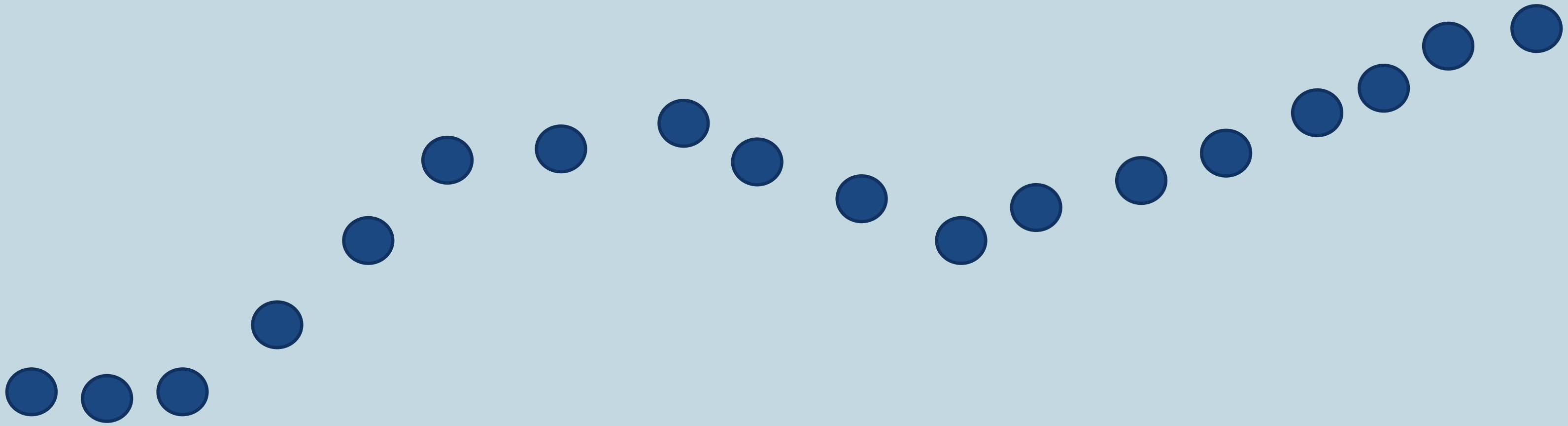
Algorithm Overview

Input: Oriented or unoriented point set.

Output: Smooth, watertight surface mesh.

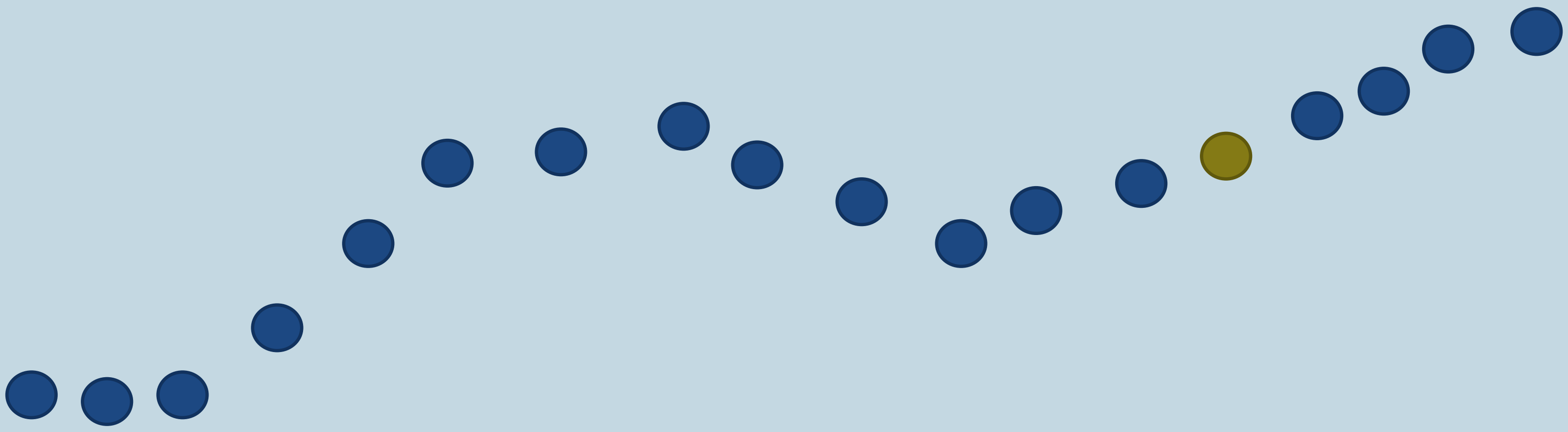
Estimating Unoriented Normals

PCA



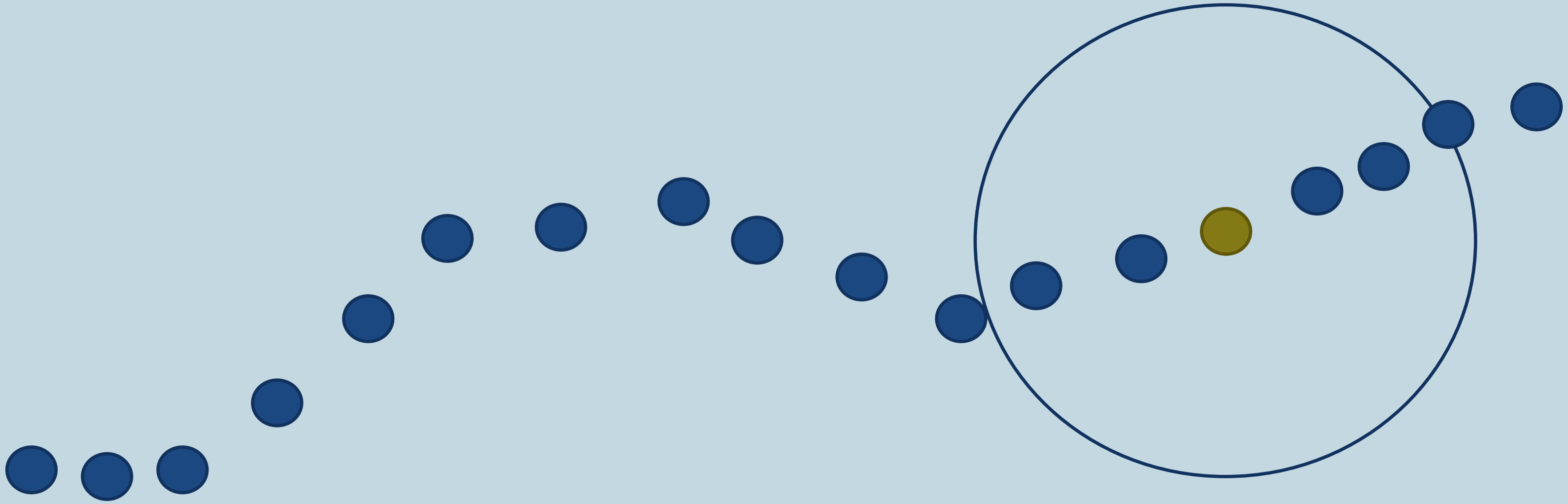
Estimating Unoriented Normals

PCA



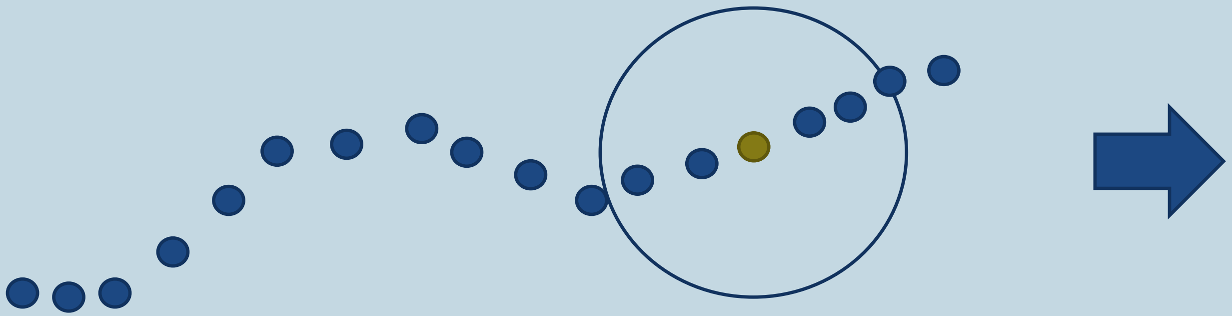
Estimating Unoriented Normals

PCA



Estimating Unoriented Normals

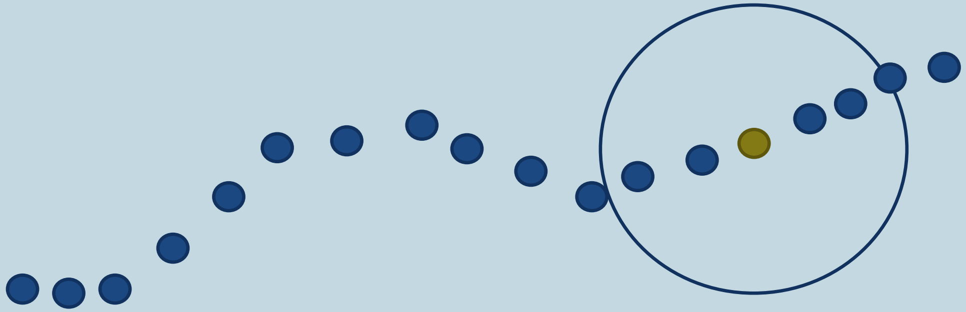
PCA



$$C = \begin{bmatrix} \; \end{bmatrix}$$

Estimating Unoriented Normals

PCA

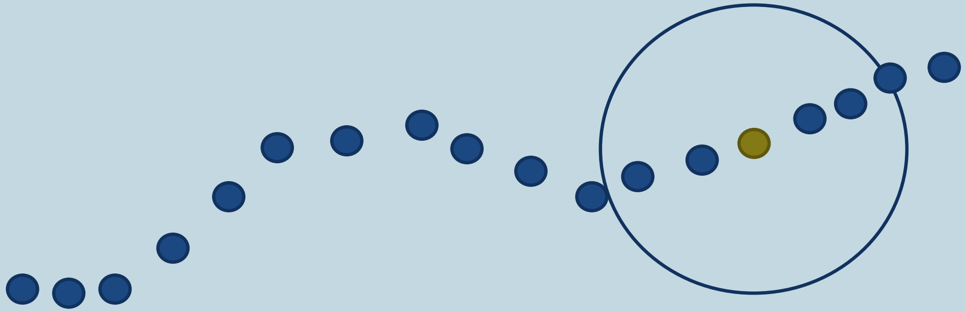


$$C = \begin{bmatrix} \; \end{bmatrix}$$

$$\lambda_1, v_1, \lambda_2, v_2, \lambda_3, v_3$$
$$\lambda_1 < \lambda_2 < \lambda_3$$

Estimating Unoriented Normals

PCA



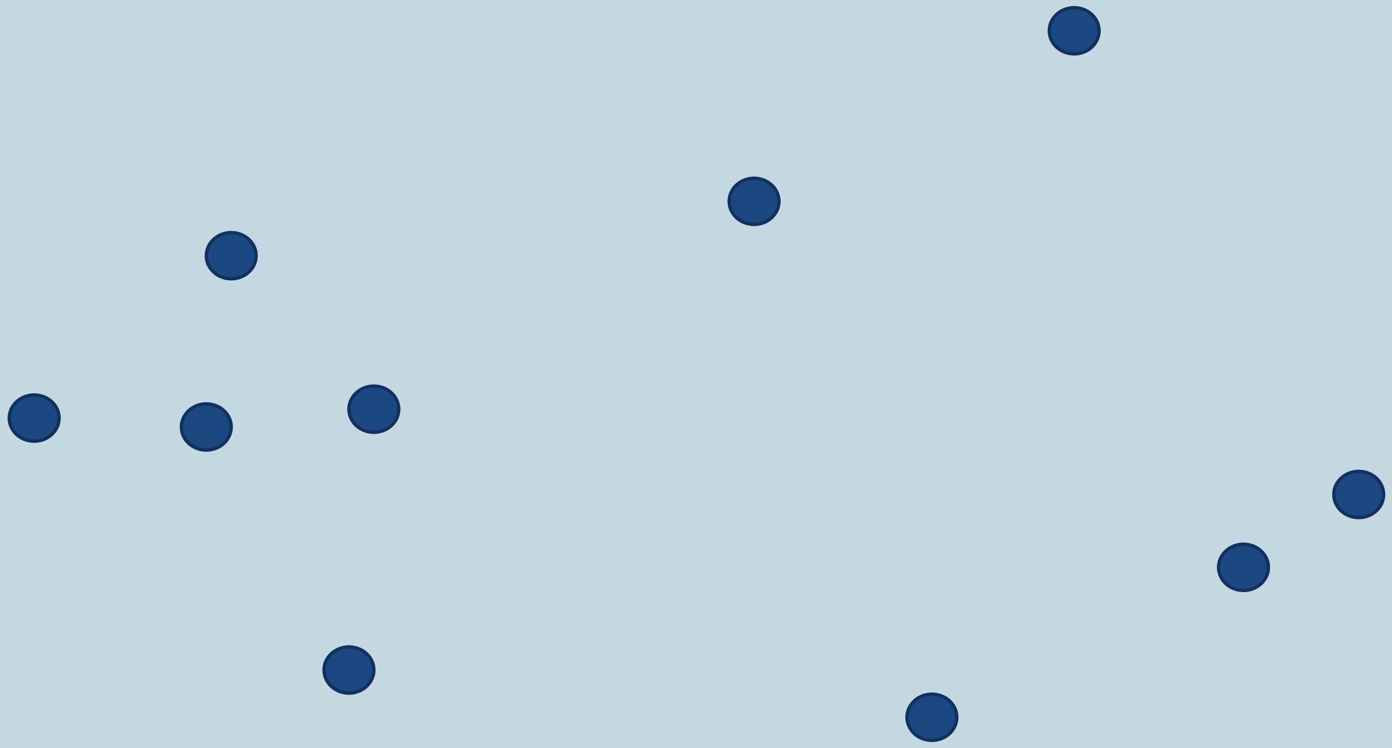
$$C = \begin{bmatrix} \quad & \quad & \quad \\ \quad & \quad & \quad \\ \quad & \quad & \quad \end{bmatrix}$$

$$\lambda_1, v_1, \lambda_2, v_2, \lambda_3, v_3$$
$$\lambda_1 < \lambda_2 < \lambda_3$$

Normal Direction

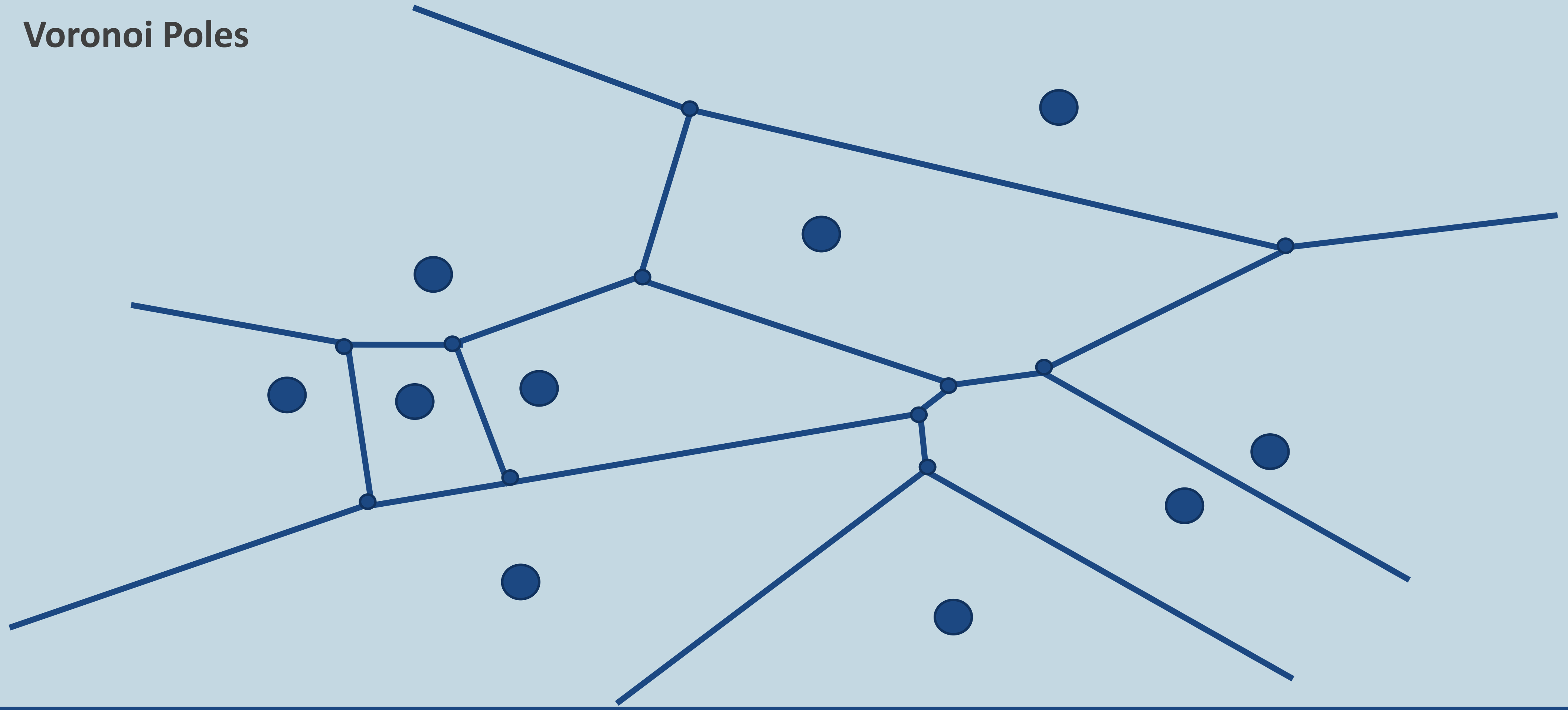
Estimating Unoriented Normals

Voronoi Poles



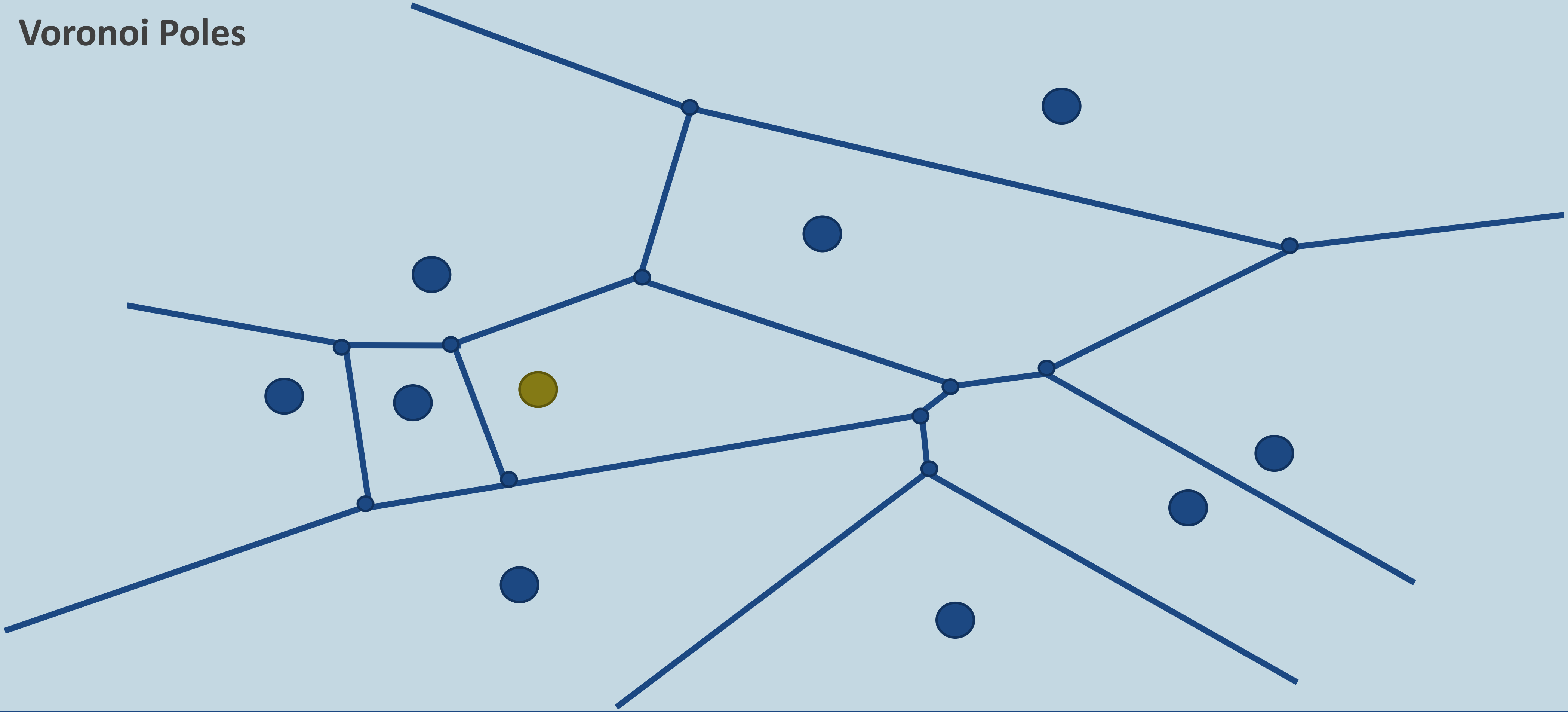
Estimating Unoriented Normals

Voronoi Poles



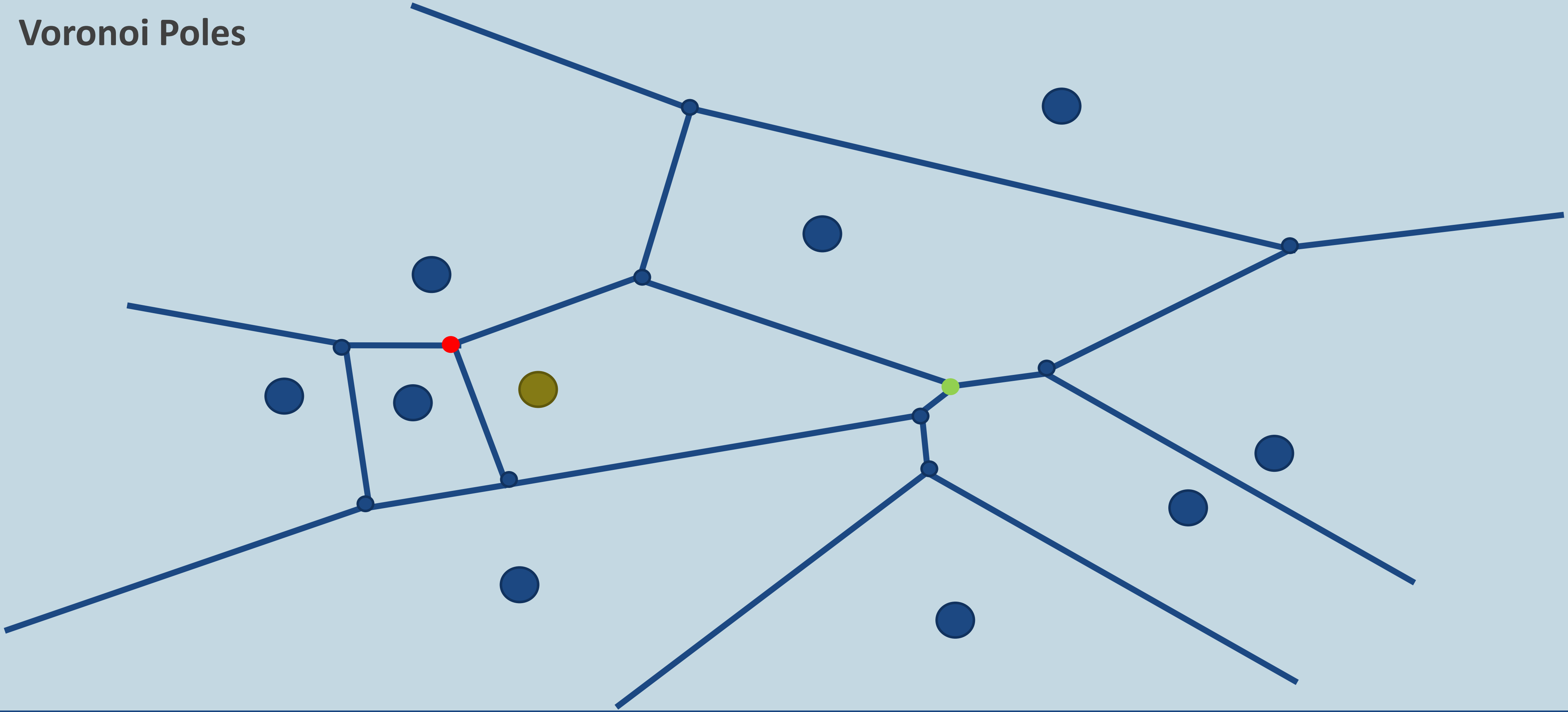
Estimating Unoriented Normals

Voronoi Poles



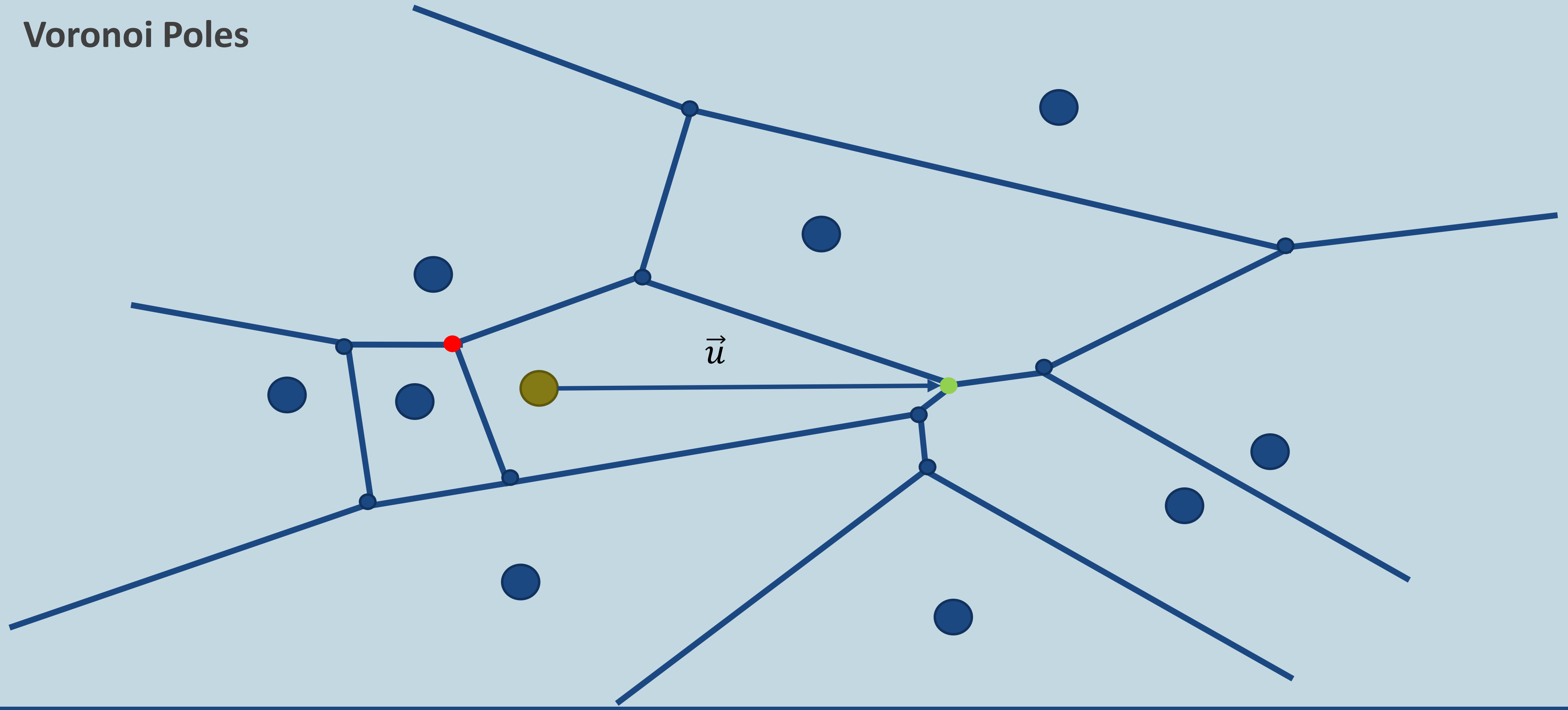
Estimating Unoriented Normals

Voronoi Poles



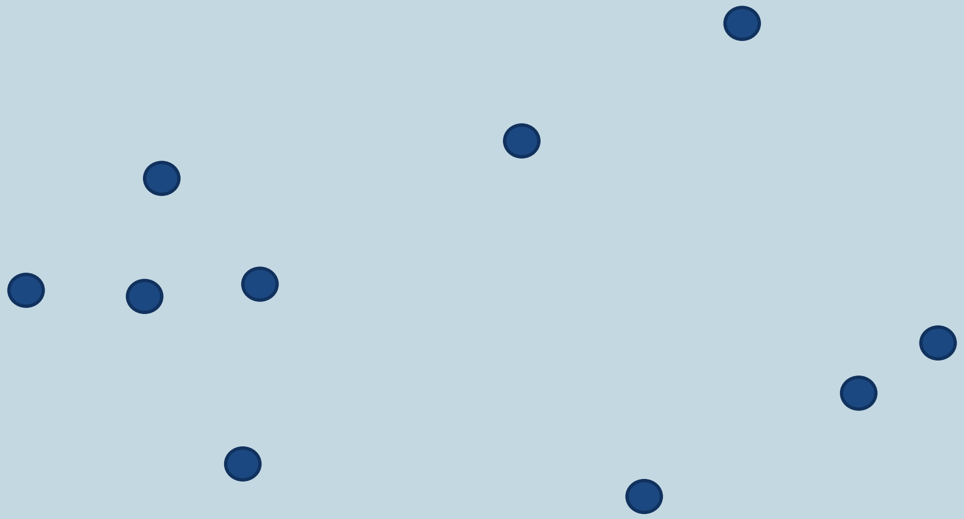
Estimating Unoriented Normals

Voronoi Poles



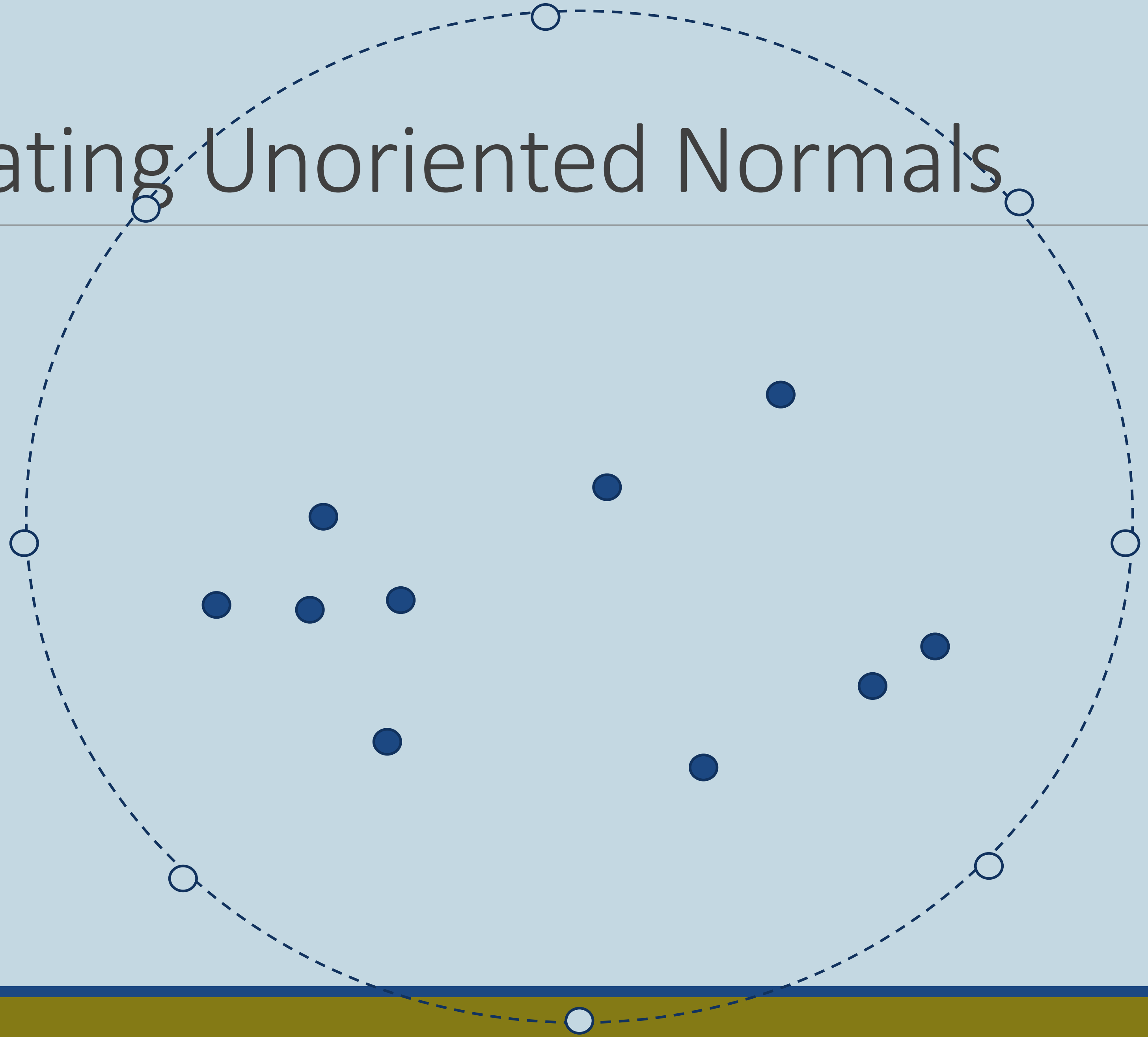
Estimating Unoriented Normals

Voronoi-PCA



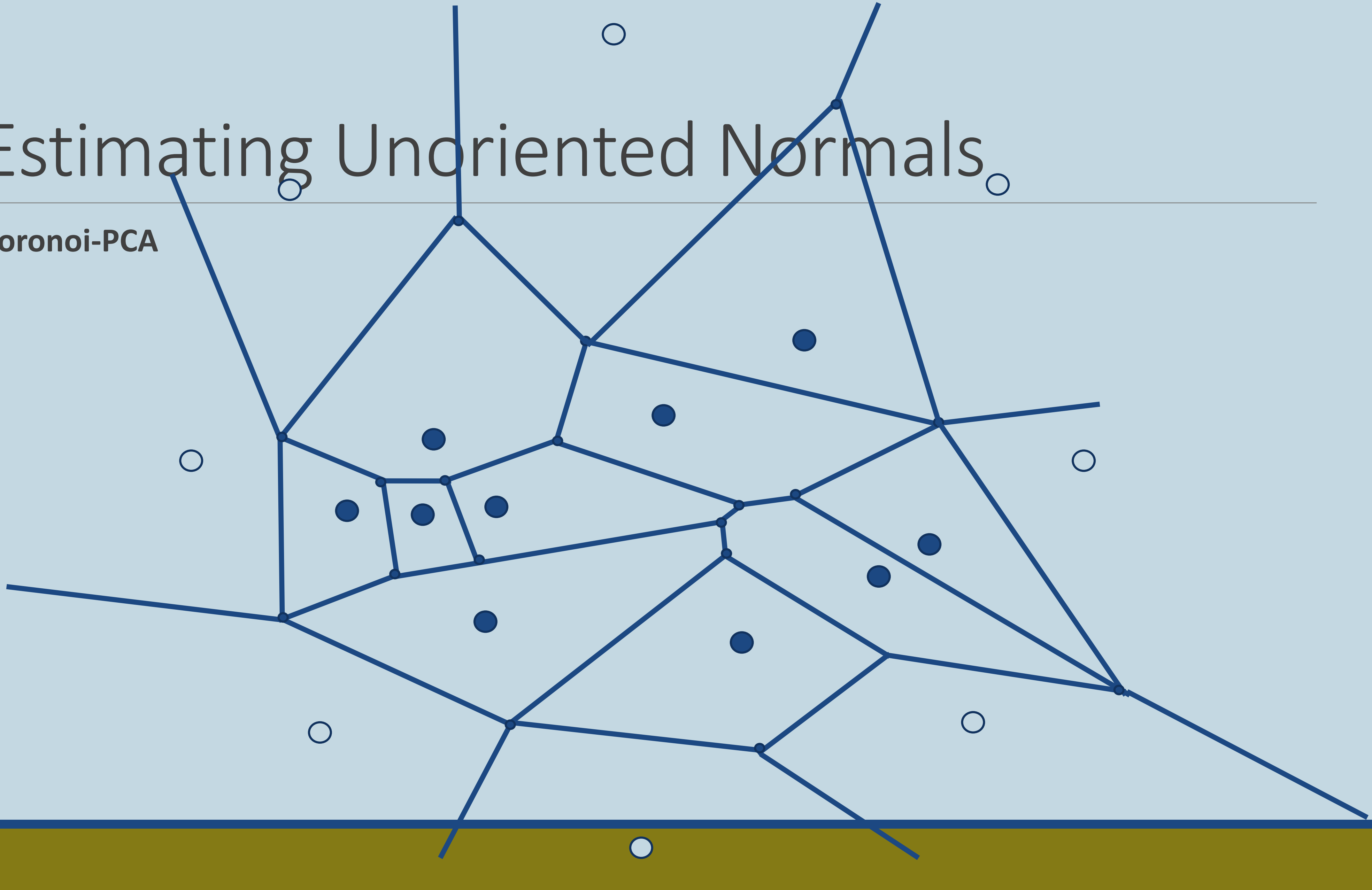
Estimating Unoriented Normals

Voronoi-PCA



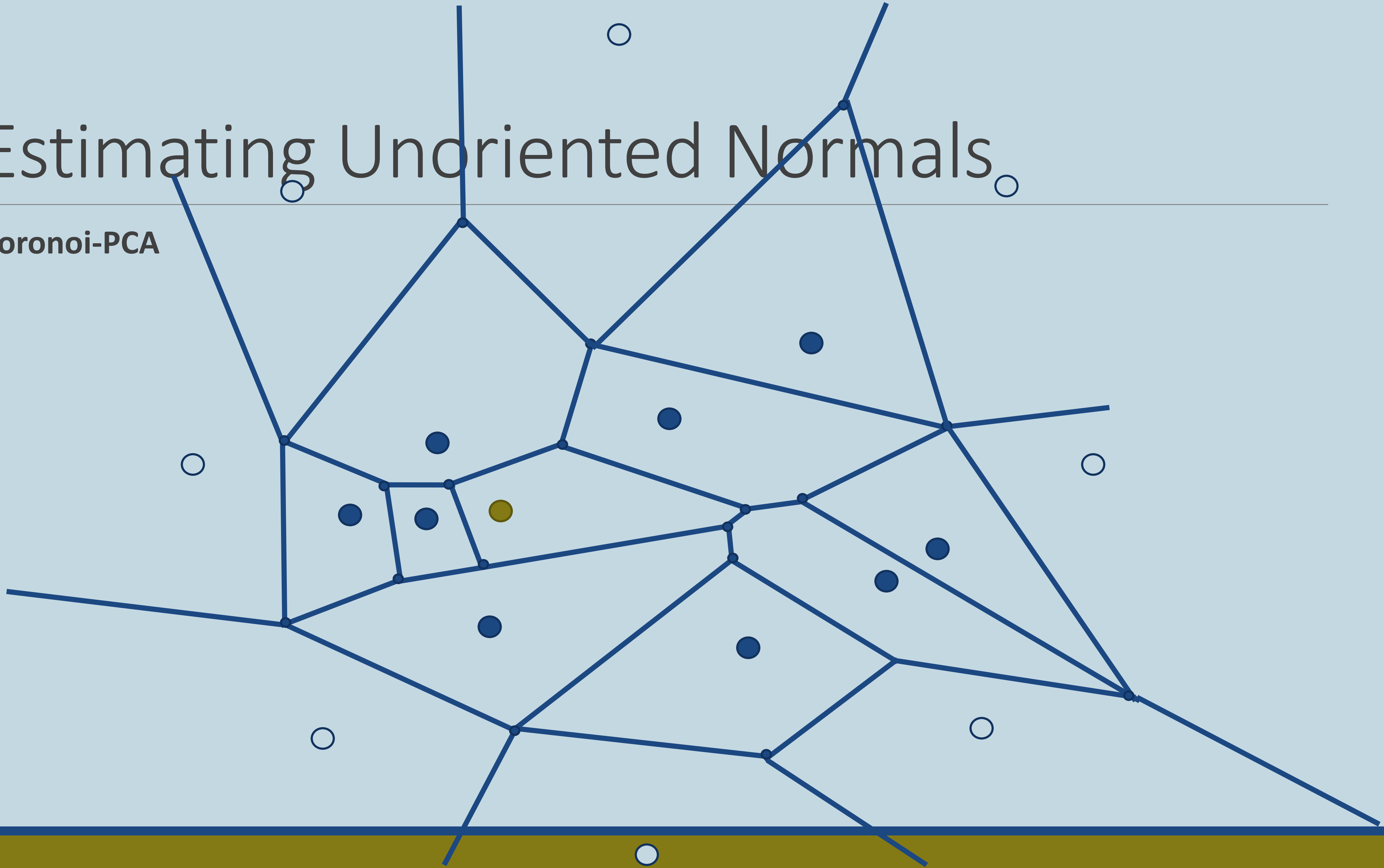
Estimating Unoriented Normals

Voronoi-PCA



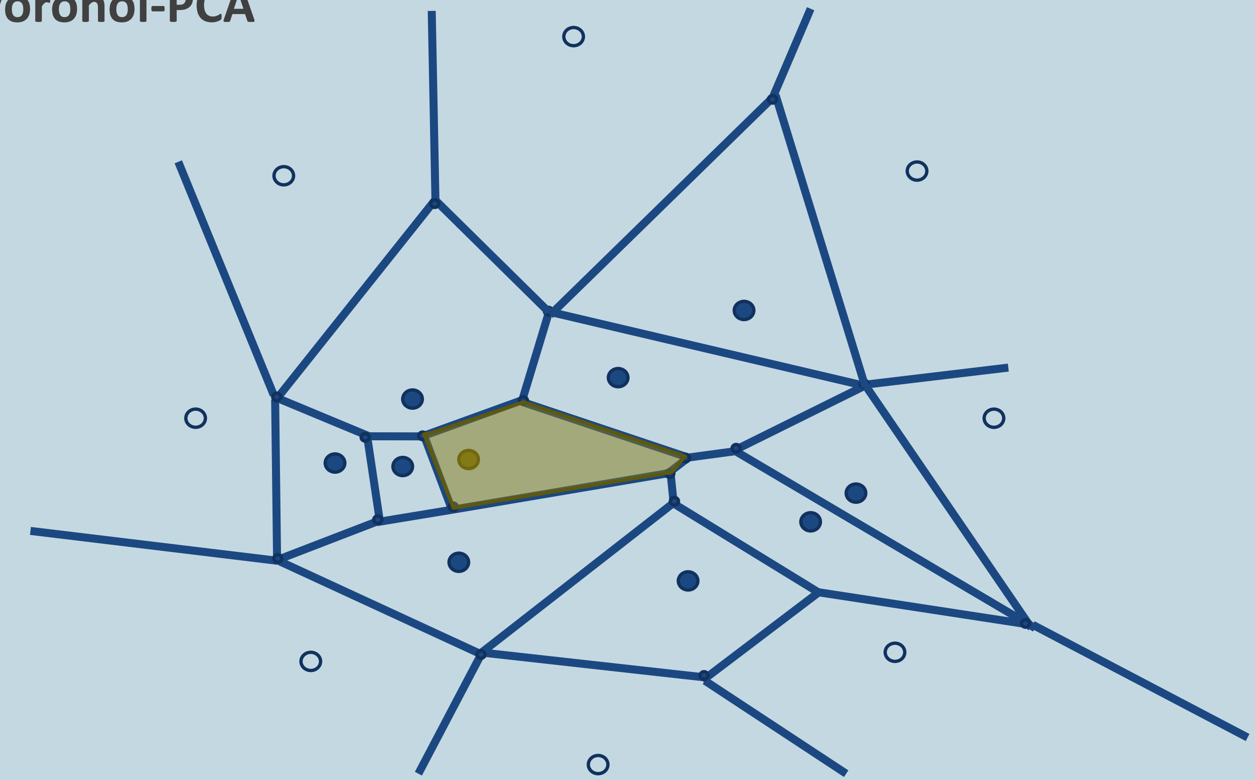
Estimating Unoriented Normals

Voronoi-PCA



Estimating Unoriented Normals

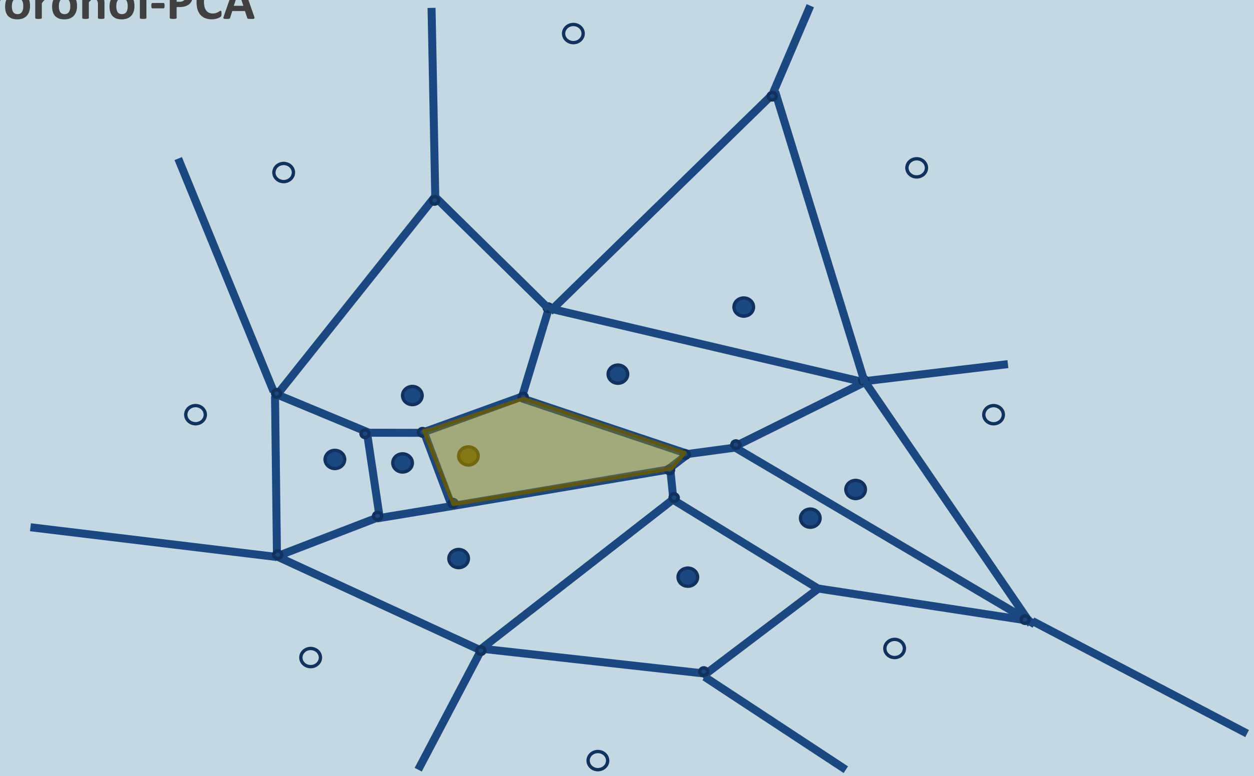
Voronoi-PCA



$$C = \begin{bmatrix} \quad & \quad \\ \quad & \quad \end{bmatrix}$$

Estimating Unoriented Normals

Voronoi-PCA

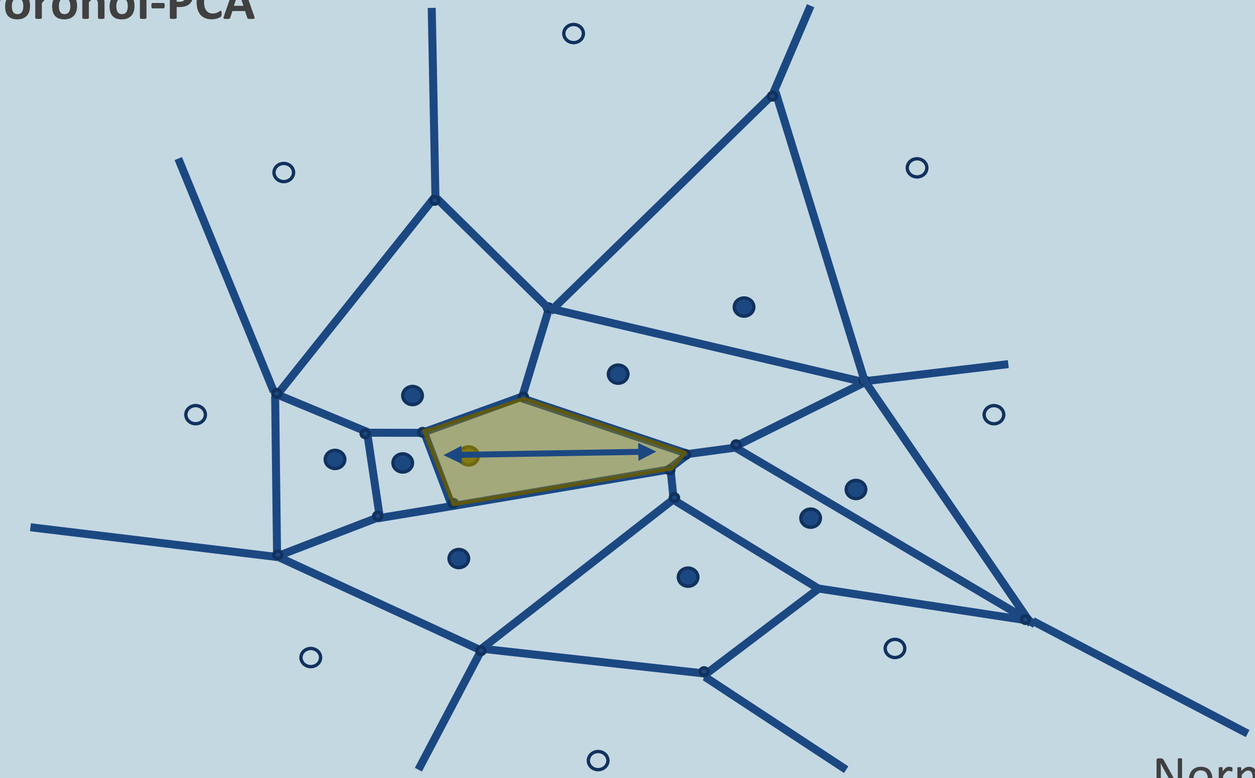


$$C = \begin{bmatrix} \\ \\ \end{bmatrix}$$

$$\lambda_1, v_1, \lambda_2, v_2, \lambda_3, v_3$$
$$\lambda_1 < \lambda_2 < \lambda_3$$

Estimating Unoriented Normals

Voronoi-PCA



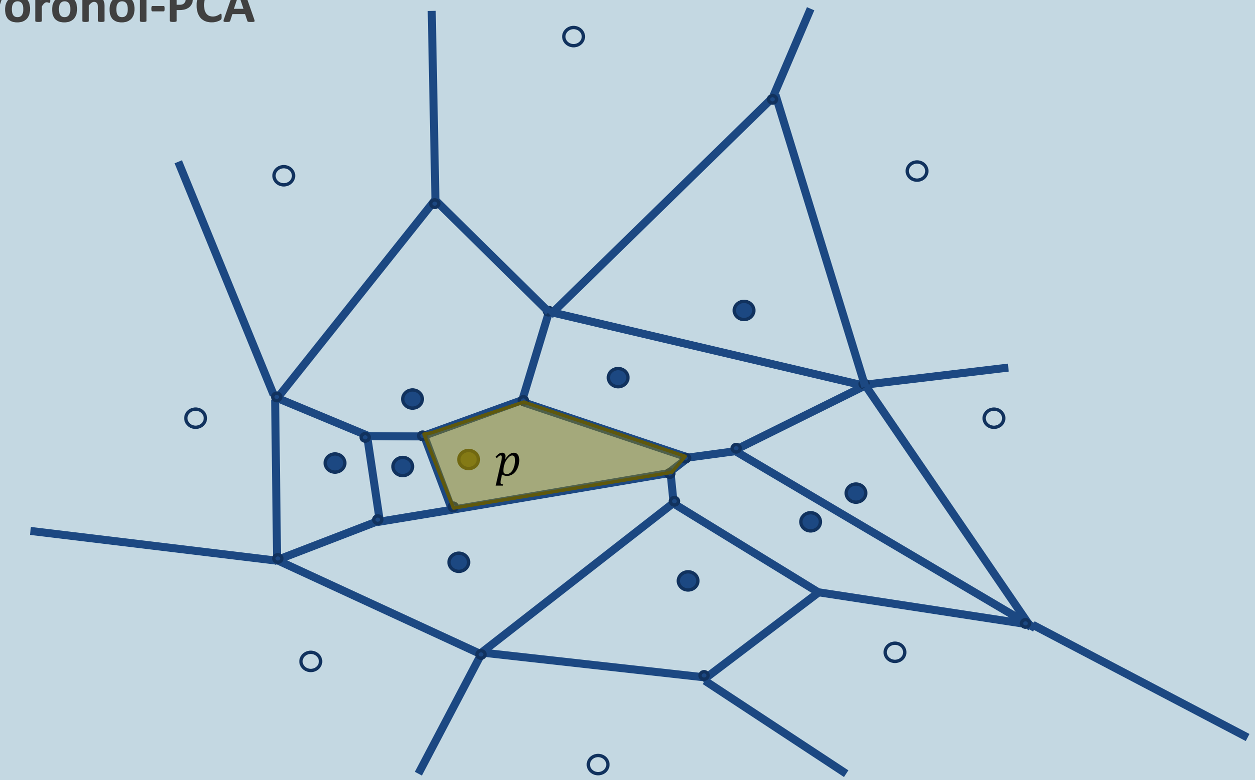
$C =$

$\lambda_1, v_1, \lambda_2, v_2, \lambda_3, v_3$
$\lambda_1 < \lambda_2 < \lambda_3$

Normal Direction

Estimating Unoriented Normals

Voronoi-PCA



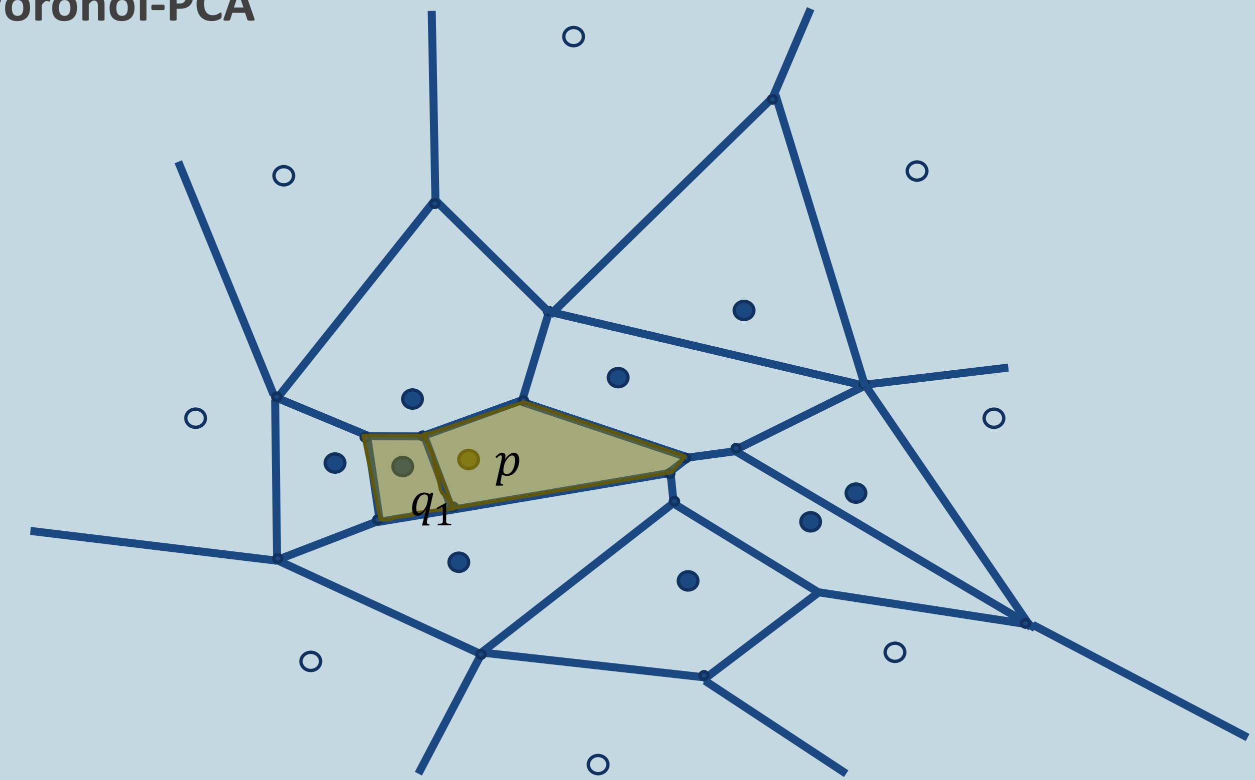
$$C = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

$$\lambda_1, v_1, \lambda_2, v_2, \lambda_3, v_3$$
$$\lambda_1 < \lambda_2 < \lambda_3$$

Anisotropy $(V_p) = 1 - \text{Isotropy}(V_p) = 1 - \frac{v_1}{v_3}$

Estimating Unoriented Normals

Voronoi-PCA



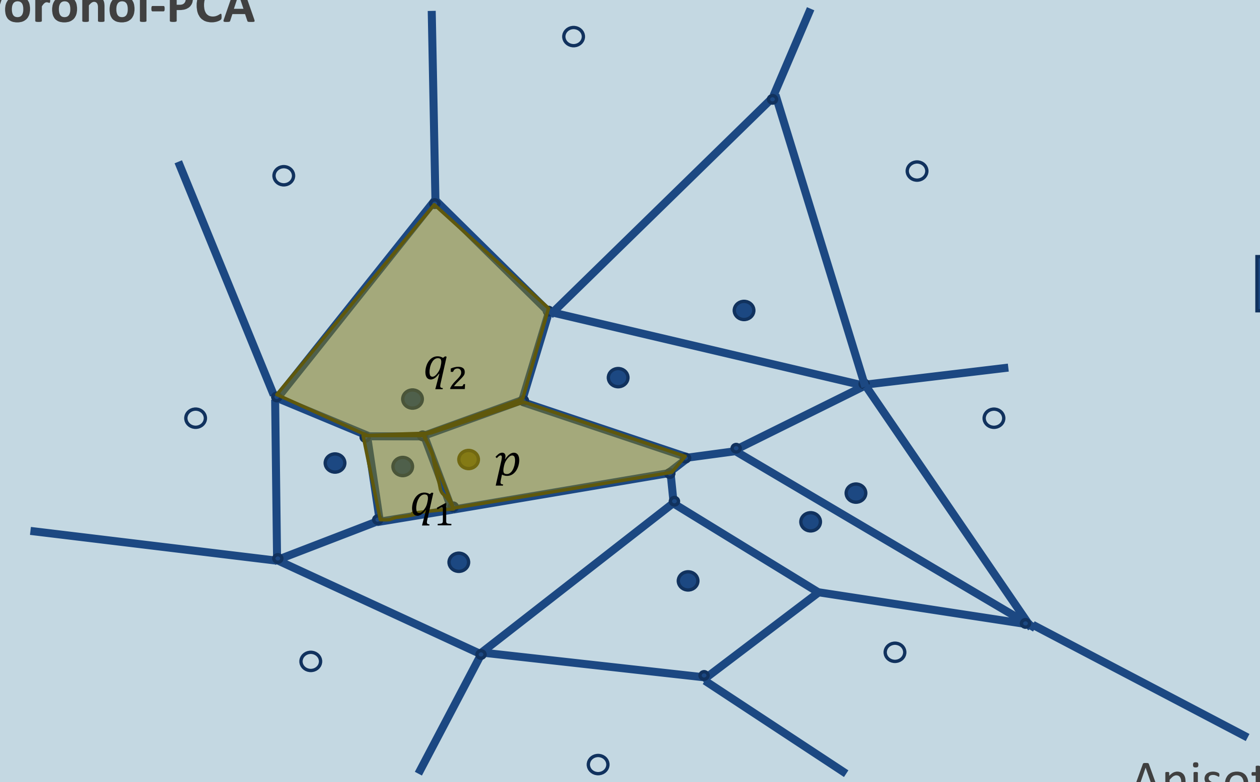
$$C = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

$$\lambda_1, v_1, \lambda_2, v_2, \lambda_3, v_3$$
$$\lambda_1 < \lambda_2 < \lambda_3$$

Anisotropy $(V_p \cup V_{q_1}) >$ Anisotropy(V_p)

Estimating Unoriented Normals

Voronoi-PCA



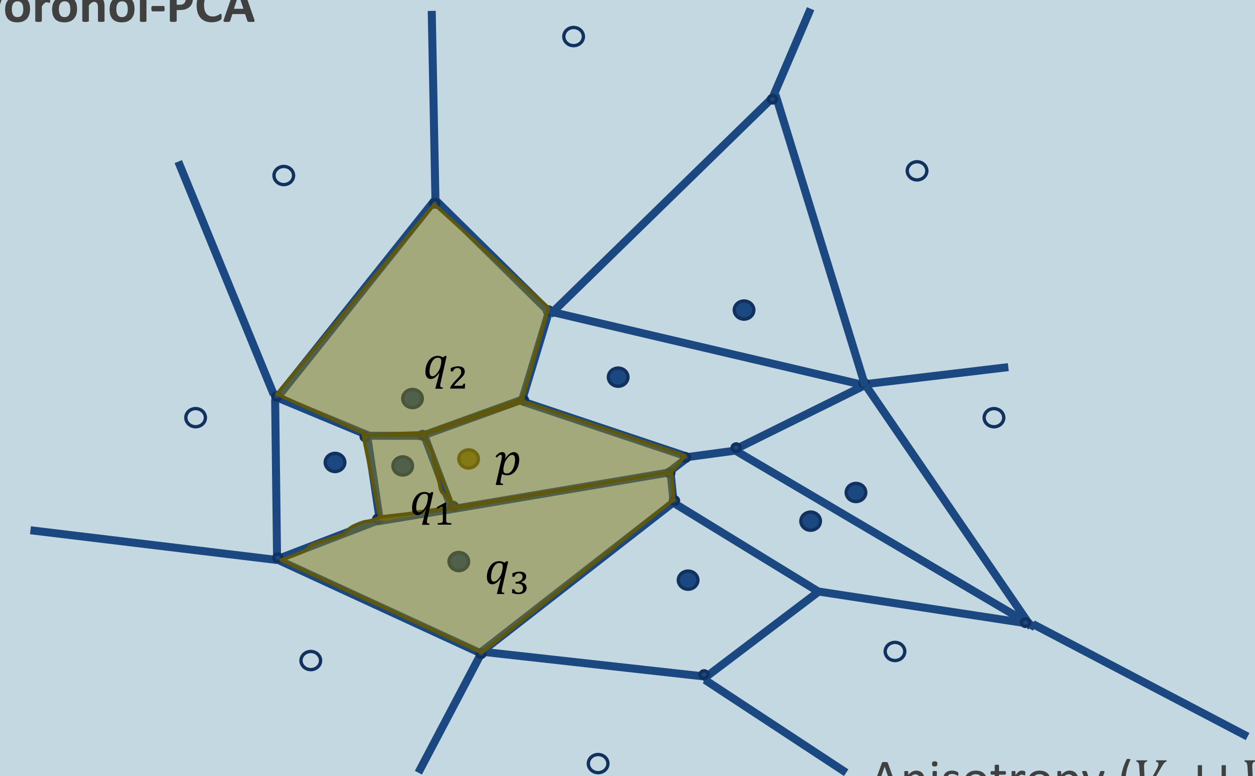
$C =$

$$\lambda_1, v_1, \lambda_2, v_2, \lambda_3, v_3$$
$$\lambda_1 < \lambda_2 < \lambda_3$$

Anisotropy $(V_p \cup V_{q_1} \cup V_{q_2}) >$ Anisotropy$(V_p \cup V_{q_1})$

Estimating Unoriented Normals

Voronoi-PCA



$$C = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

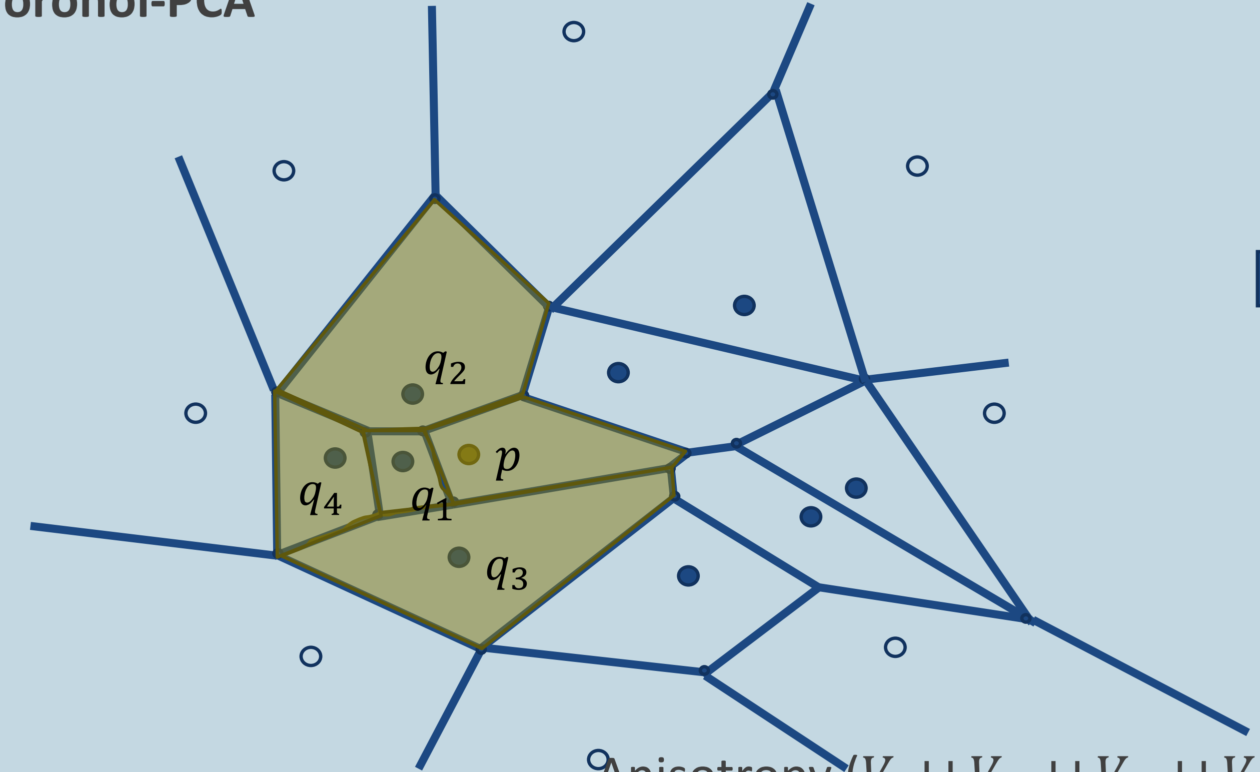
$\lambda_1, v_1, \lambda_2, v_2, \lambda_3, v_3$

$\lambda_1 < \lambda_2 < \lambda_3$

Anisotropy $(V_p \cup V_{q_1} \cup V_{q_2} \cup V_{q_3}) >$ Anisotropy$(V_p \cup V_{q_1} \cup V_{q_2})$

Estimating Unoriented Normals

Voronoi-PCA



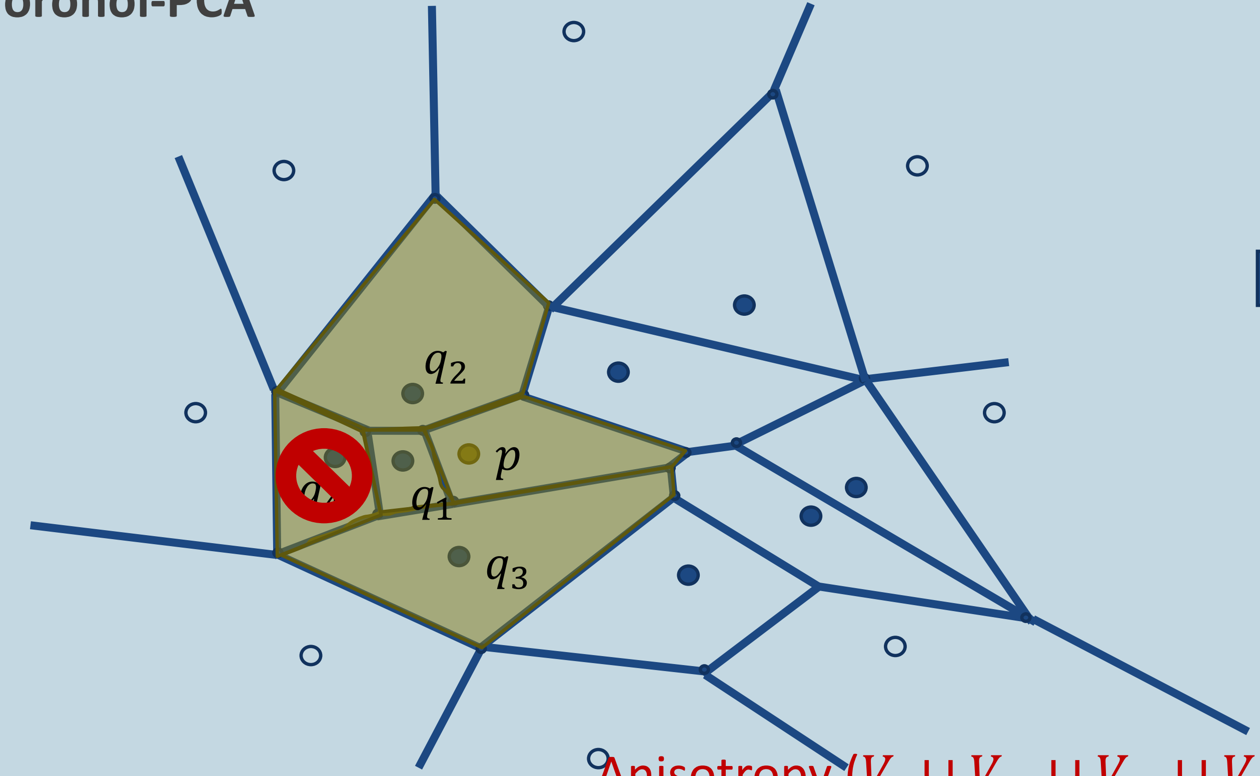
$C =$

$\lambda_1, v_1, \lambda_2, v_2, \lambda_3, v_3$
$\lambda_1 < \lambda_2 < \lambda_3$

$\text{Anisotropy}\,(V_p \cup V_{q_1} \cup V_{q_2} \cup V_{q_3} \cup V_{q_4}) < \text{Anisotropy}(V_p \cup V_{q_1} \cup V_{q_2} \cup V_{q_3})$

Estimating Unoriented Normals

Voronoi-PCA



$$C =$$

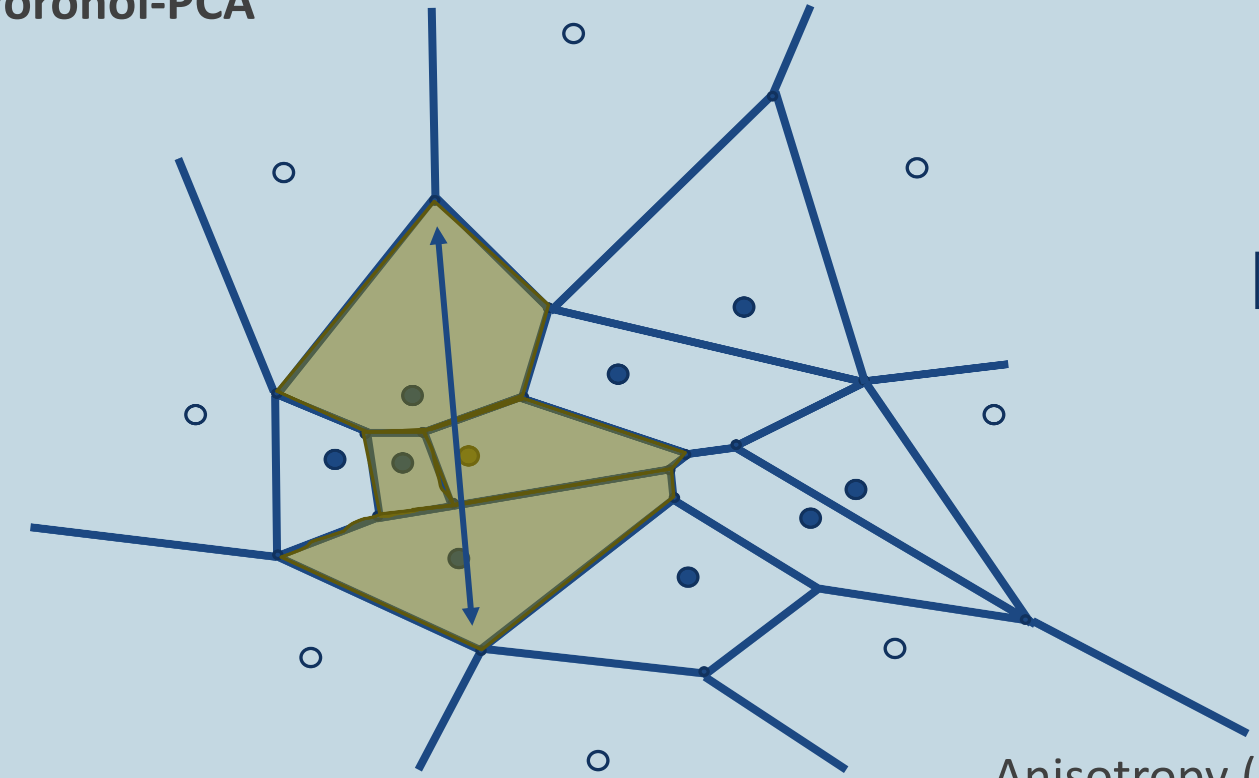
$\lambda_1, v_1, \lambda_2, v_2, \lambda_3, v_3$

$\lambda_1 < \lambda_2 < \lambda_3$

Anisotropy $(V_p \cup V_{q_1} \cup V_{q_2} \cup V_{q_3} \cup V_{q_4}) <$ Anisotropy$(V_p \cup V_{q_1} \cup V_{q_2} \cup V_{q_3})$

Estimating Unoriented Normals

Voronoi-PCA



$$C = \begin{bmatrix} \quad \end{bmatrix}$$

$$\lambda_1, v_1, \lambda_2, v_2, \lambda_3, v_3$$
$$\lambda_1 < \lambda_2 < \lambda_3$$

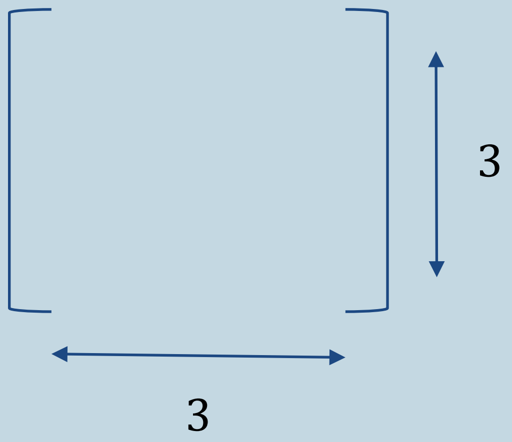
$$\text{Anisotropy}\,(V_p \cup V_{q_1} \cup V_{q_2} \cup V_{q_3}) = \text{Maximum Anisotropy}$$

Generalized Eigenvalue Problem

Want: f such that ∇f is everywhere best aligned to the maximum eigenvalue direction of C.

Given:

$C =$



Positive definite
Symmetric

Eigenvectors
v_1, v_2, v_3
Eigenvalues
$\lambda_1, \lambda_2, \lambda_3$
$\lambda_1 < \lambda_2 < \lambda_3$

Generalized Eigenvalue Problem

Want: f such that ∇f is everywhere best aligned to the maximum eigenvalue direction of C.

Assume that:

$$C = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

$$\nabla f = a_1 v_1 + a_2 v_2 + a_3 v_3$$

Generalized Eigenvalue Problem

Want: f such that ∇f is everywhere best aligned to the maximum eigenvalue direction of C.

Assume that:

$$C = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

$$\nabla f = a_1 v_1 + a_2 v_2 + a_3 v_3$$

We get:

$$\begin{aligned} \int_\Omega \nabla f^t C \nabla f &= \int_\Omega (a_1 v_1 + a_2 v_2 + a_3 v_3)^t \mathrm{C} (a_1 v_1 + a_2 v_2 + a_3 v_3) \\ &= \int_\Omega (\lambda_1 a_1^2 + \lambda_2 a_2^2 + \lambda_3 a_3^2)\,, \end{aligned}$$

$$v_i v_i^t = 1, v_i v_j^t = 0, i \neq j, i, j = 1,2,3$$

Generalized Eigenvalue Problem

Want: f such that ∇f is everywhere best aligned to the maximum eigenvalue direction of C.

Assume that:

$$C = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

$$\nabla f = a_1 v_1 + a_2 v_2 + a_3 v_3$$

We get:

$$\int_\Omega \nabla f^t C \nabla f = \int_\Omega (a_1 v_1 + a_2 v_2 + a_3 v_3)^t \mathrm{C} (a_1 v_1 + a_2 v_2 + a_3 v_3)$$
$$= \int_\Omega (\lambda_1 a_1^2 + \lambda_2 a_2^2 + \underbrace{\lambda_3 a_3^2}) \,,$$

f is maximized when aligned with the eigenvector corresponding to the largest eigenvalue of C

Generalized Eigenvalue Problem

Want: f such that ∇f is everywhere best aligned to the maximum eigenvalue direction of C.

Solution: Find f that maximizes

$$E_C^D(f) = \int_\Omega \nabla f^t C \nabla f$$

Subject to $\int_\Omega (|\Delta f|^2 + \epsilon |f|^2) = 1,$

Where Ω is the domain, and Δ is the Laplacian operator.

Generalized Eigenvalue Problem

Want: f such that ∇f is everywhere best aligned to the maximum eigenvalue direction of C.

Regularization constraint:

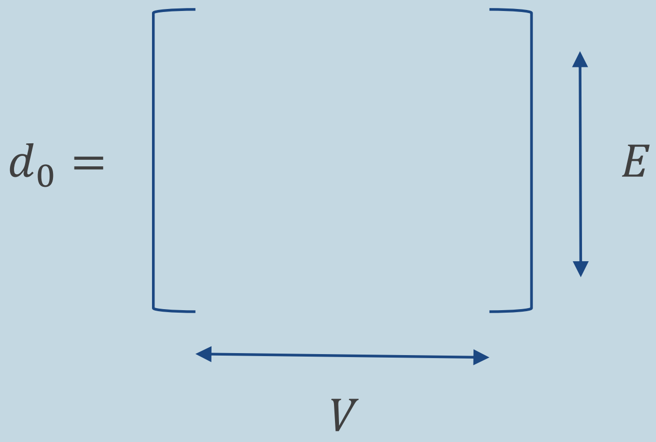
$$\int_\Omega (|\Delta f|^2 + \epsilon |f|^2) = 1$$

Ensures that maximization of the energy term is not being achieved by scaling the function f

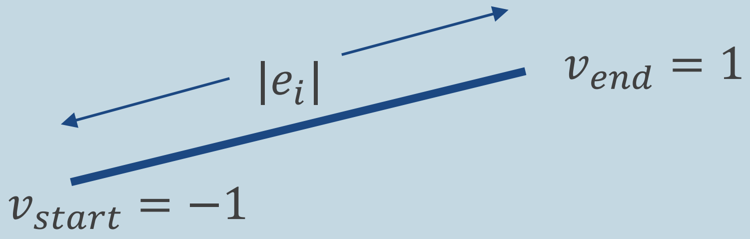
Discrete Formulation

Want: $F = (f_1, f_2, \ldots, f_V)^t$ that satisfies the constrained maximization (from previous slide)

Let:



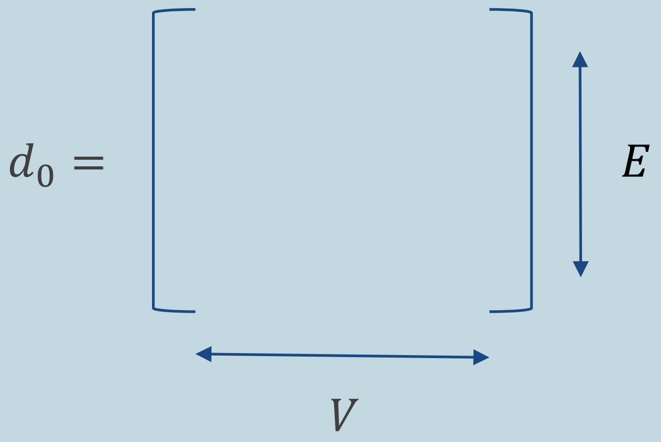
Where:



Discrete Formulation

Want: $F = (f_1, f_2, \dots, f_V)^t$ that satisfies the constrained maximization

Let:



Then:

$$F^t d_0^t = f_{end} - f_{start}$$

$$F^t d_0^t d_0 F = (f_{end} - f_{start})^2$$

Discrete Formulation

Want: $F = (f_1, f_2, \dots, f_V)^t$ that satisfies the constrained maximization

But:

$$\nabla f = \frac{f_{end} - f_{start}}{|e_i|}$$

$$\Rightarrow (\nabla f)^2 = \frac{(f_{end} - f_{start})^2}{|e_i|^2}$$

Discrete Formulation

Want: $F = (f_1, f_2, \ldots, f_V)^t$ that satisfies the constrained maximization

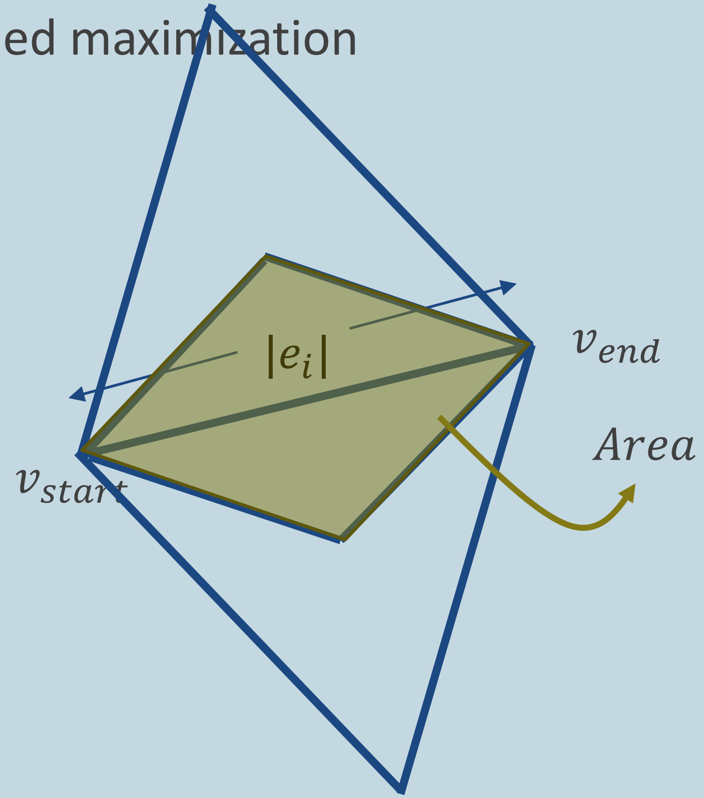
Therefore:

If $C = I_{3x3}$,

Then, we want to maximize:

$$\frac{(f_{end} - f_{start})^2}{|e_i|^2} Area$$

Where:



Discrete Formulation

Want: $F = (f_1, f_2, \dots, f_V)^t$ that satisfies the constrained maximization

Therefore:

If $C = I_{3x3}$,

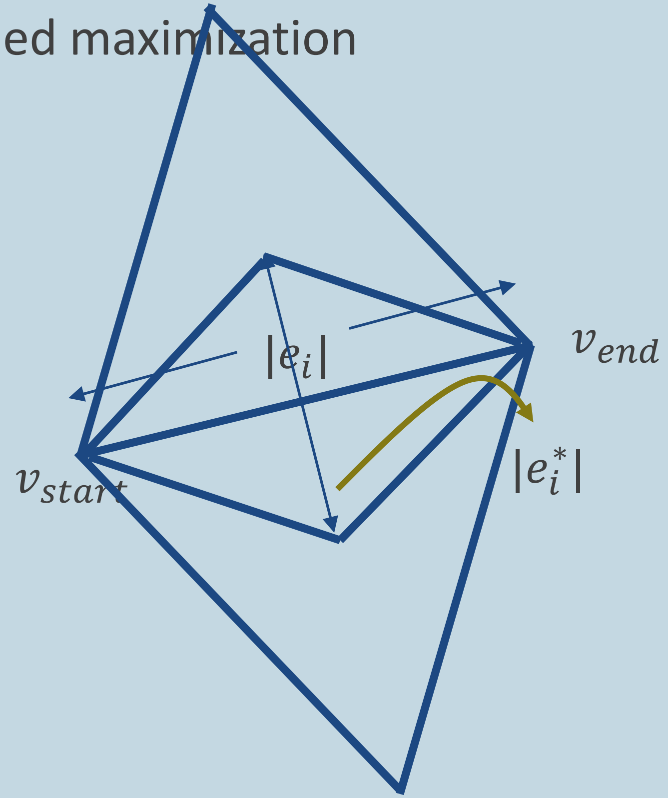
Then, we want to maximize:

$$\frac{(f_{end} - f_{start})^2}{|e_i|^2} |e_i||e_i^*|$$

$$= \frac{(f_{end} - f_{start})^2}{|e_i|} |e_i^*|$$

$$= (f_{end} - f_{start})^2 \, (\star^1)_{ii}$$

Where:



Discrete Formulation

Want: $F = (f_1, f_2, \dots, f_V)^t$ that satisfies the constrained maximization (from previous slide)

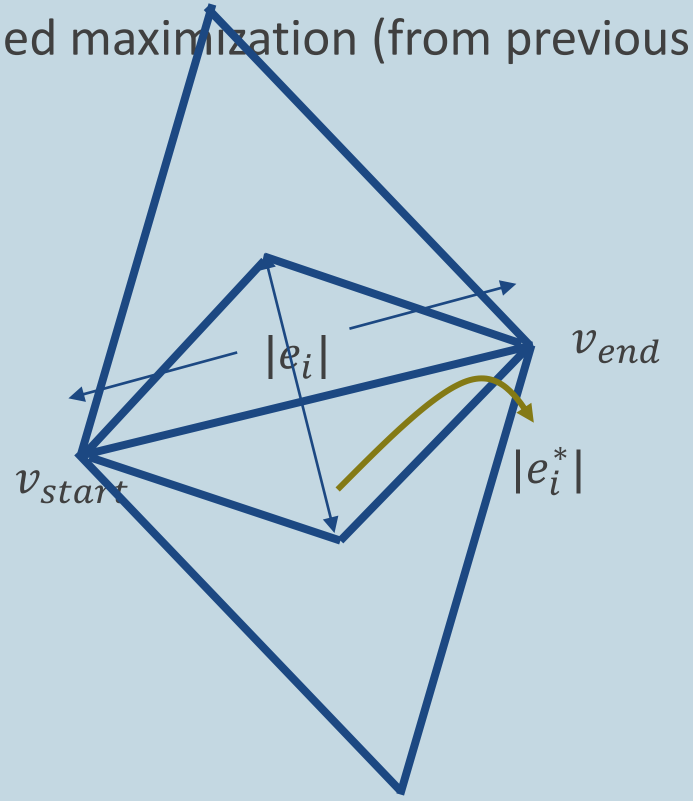
Therefore:

If $C \neq I_{3x3}$,

Then, we want to maximize:

$$\begin{aligned}(f_{end} - f_{start})^2 \frac{e_i^t C\, e_i}{e_i^t e_i} (\star^1)_{ii} \\ = (f_{end} - f_{start})^2 \, (\star_C^1)_{ii} \\ = F^t d_0^t (\star_C^1)\, d_0 F \\ = F^t A\, F \\ \approx E_C^D(F)\end{aligned}$$

Where:



Discrete Formulation

Want: $F = (f_1, f_2, \ldots, f_V)^t$ that satisfies the constrained maximization (from previous slide)

Solution: Find *F* that maximizes

$$E_C^D(F) \approx F^t A\, F \text{ with } A = d_0^t \star_C^1 d_0,$$

Where d_0 is the transpose of the signed vertex/edge incidence matrix of the mesh,

$\star_C^1$ is the Hodge star operator for the metric induced by *C*, such that

$$(\star_C^1)_{ii} = \frac{e_i^t C e_i}{e_i^t e_i} (\star^1)_{ii} \text{ with } (\star^1)_{ii} = \frac{|e_i^*|}{|e_i|}, \forall i = 1, \ldots, E$$

Discrete Formulation

Want: $F = (f_1, f_2, \ldots, f_V)^t$ that satisfies the constrained maximization (from previous slide)

Similarly: Find *F* that maximizes

$$E^B(F) \approx F^t B\, F \text{ with } B = (d_0^t \star^1 d_0)^2,$$

For the biharmonic energy, where $(d_0^t \star^1 d_0)^2$ is the bilaplacian.

Discrete Formulation

Want: $\nabla(F^t A\, F) \parallel \nabla(F^t B\, F)$

Solution:

We want to maximize $E = F^t A F + \lambda(1\, - F^t B F)$

$$\frac{\partial E}{\partial F} = \nabla(F^t A F) + 0\, - \lambda \nabla(F^t B F)$$

$$\frac{\partial E}{\partial F} = 0 \Rightarrow 2AF - 2\lambda BF = 0$$

$$\Rightarrow AF = \lambda BF$$

Discrete Formulation

Want: $AF = \lambda BF$

Solution:

$$B = LL^t$$

$$\Rightarrow AF = \lambda LL^t F$$

$$\Rightarrow L^{-1}AF = \lambda L^t F$$

$$\Rightarrow L^{-1}AIF = \lambda L^t F$$

$$\Rightarrow L^{-1}AL^{-t}L^t F = \lambda L^t F$$

$$\Rightarrow L^{-1}AL^{-t}G = \lambda G$$

Discrete Formulation

Want: $AF = \lambda BF$

Solution:

$$B = LL^t$$

$$\Rightarrow AF = \lambda LL^t F$$

$$\Rightarrow L^{-1}AF = \lambda L^t F$$

$$\Rightarrow L^{-1}AIF = \lambda L^t F$$

$$\Rightarrow L^{-1}AL^{-t}L^t F = \lambda L^t F$$

$$\Rightarrow L^{-1}AL^{-t}G = \lambda G$$

$$\boxed{L^t F = G}$$

Delaunay Refinement

Recall: $F^t A F = \underbrace{(f_{end} - f_{start})}_{\text{Variance along each edge}}{}^2 \, \frac{e_i^t C \, e_i}{e_i^t e_i} (\star^1)_{ii}$

Delaunay Refinement

Recall: $F^t A F = \underbrace{(f_{end} - f_{start})}_{\text{Variance along each edge}}{}^2 \; \frac{e_i^t C \, e_i}{e_i^t e_i} (\star^1)_{ii}$

What if?

- Bad edges
- Non-uniformly distributed edges

Delaunay Refinement

Recall: $F^t A F = \underbrace{(f_{end} - f_{start})^2}_{\text{Variance along each edge}} \frac{e_i^t C\, e_i}{e_i^t e_i} (\star^1)_{ii}$

What if?

- Bad edges
- Non-uniformly distributed edges

- Edges are not likely to align with the principal direction
- Therefore, not likely to capture direction of largest variance

Delaunay Refinement

Recall: $F^t A F = \underbrace{(f_{end} - f_{start})^2}_{\text{Variance along each edge}} \frac{e_i^t C\, e_i}{e_i^t e_i} (\star^1)_{ii}$

What if?

- Bad edges
- Non-uniformly distributed edges

- Edges are not likely to align with the principal direction
- Therefore, not likely to capture direction of largest variance

Want:

- Similarly sized edges, i.e., roughly equilateral triangles
- Uniformly sampled edges

Delaunay Refinement

Recall: $F^t A F = \ (\underbrace{f_{end} - f_{start}}_{\text{Variance along each edge}})^2 \ \frac{e_i^t C \ e_i}{e_i^t e_i} (\star^1)_{ii}$

Want:

- Similarly sized edges, i.e., roughly equilateral triangles
- Uniformly sampled edges

Solution:

$$\frac{radius}{edge} > 2 \Rightarrow \text{refine tetrahedra}$$

Generalizations

Recall: $E^B(F) \approx F^t B\, F, B = \underbrace{(d_0^t \star^1 d_0)}_{\text{Laplacian}}{}^2$

Data Fitting: Replace B with $(d_0^t \star^1 d_0)^2 + \mu_{fit} D$

Tunes
- Amount of data fitting
- Separation between connected components

Splines-under-tension Energy: Replace B with $(d_0^t \star^1 d_0)^2 + \mu_\Delta d_0^t \star^1 d_0$

Tunes
- Smoothness

Generalizations

Recall: $E^B(F) \approx F^t B\, F, B = \underbrace{(d_0^t \star^1 d_0)}_{\text{Laplacian}}{}^2$

Data Fitting: Replace B with $(d_0^t \star^1 d_0)^2 + \mu_{fit} D$

Tunes

- Amount of data fitting
- Separation between connected components

Splines-under-tension Energy: Replace B with $(d_0^t \star^1 d_0)^2 + \mu_\Delta d_0^t \star^1 d_0$

Tunes

- Smoothness

$$\boxed{B = (d_0^t \star^1 d_0)^2 + \mu_{fit} D + \mu_\Delta d_0^t \star^1 d_0}$$

Contouring

Evaluate F

Pick median value as isovalue

Perform isocontouring using marching-tetrahedra or Delaunay-based surface meshing algorithm

Contouring

Evaluate F

Pick median value as isovalue

Perform isocontouring using marching-tetrahedra or Delaunay-based surface meshing algorithm

Complexity

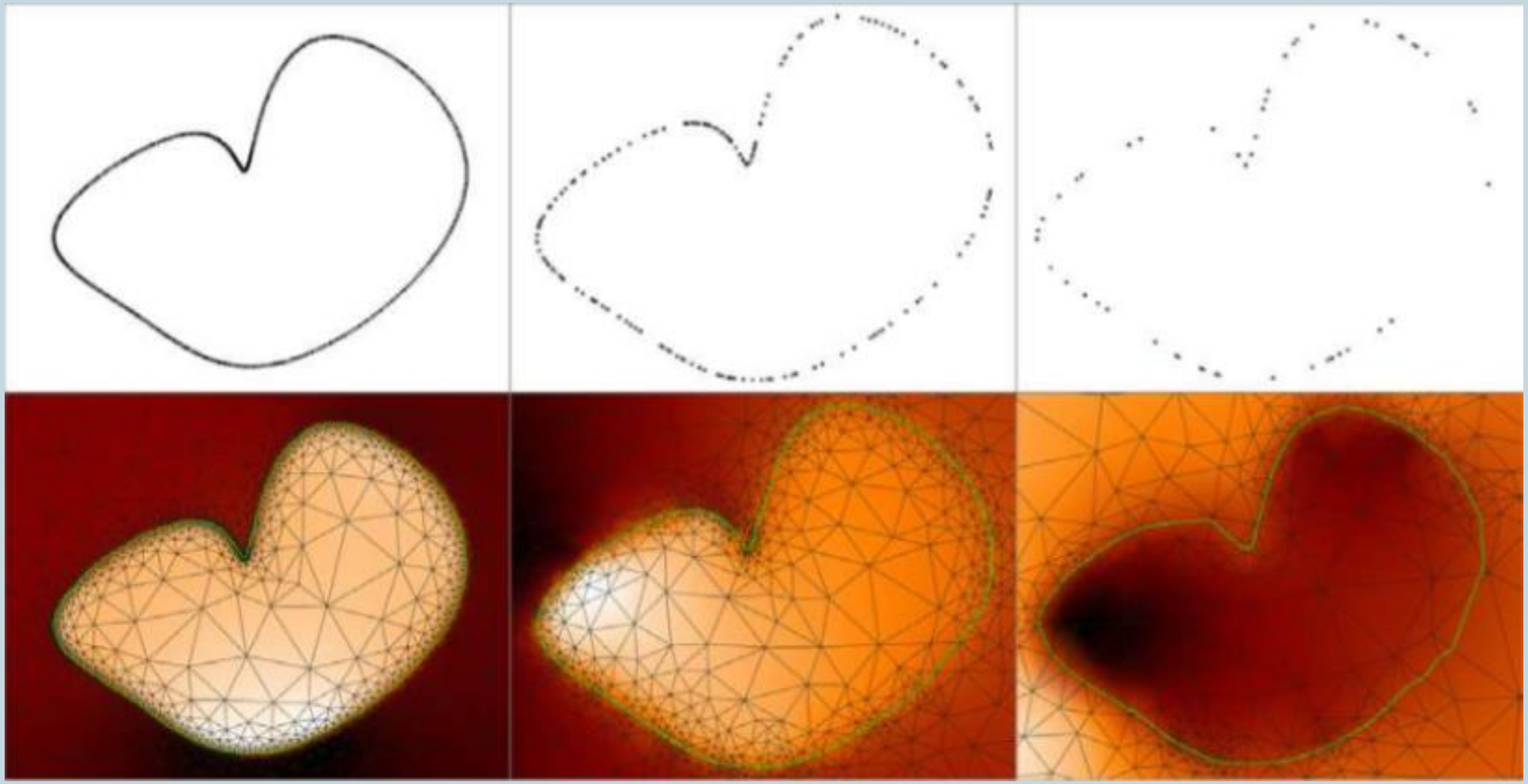
Time complexity is driven by the time required for the Implicitly Restarted Arnoldi Iteration to converge.

The data structures they use scale linearly in memory. The bottleneck in memory is caused by Cholesky Factorization.

$N =$ # points in the original point set

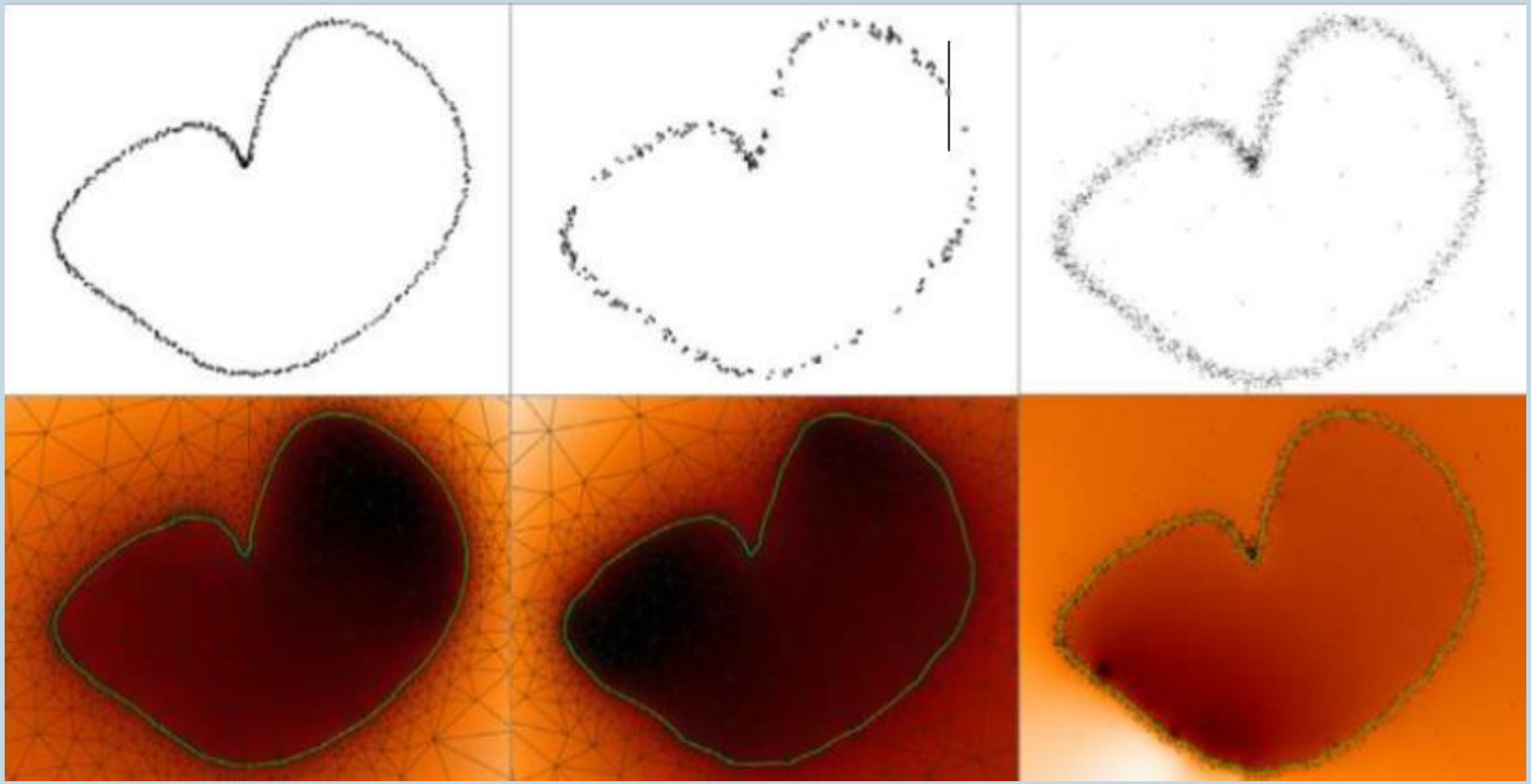
Results

Increasing sparsity



Results

Increasing noise



Results