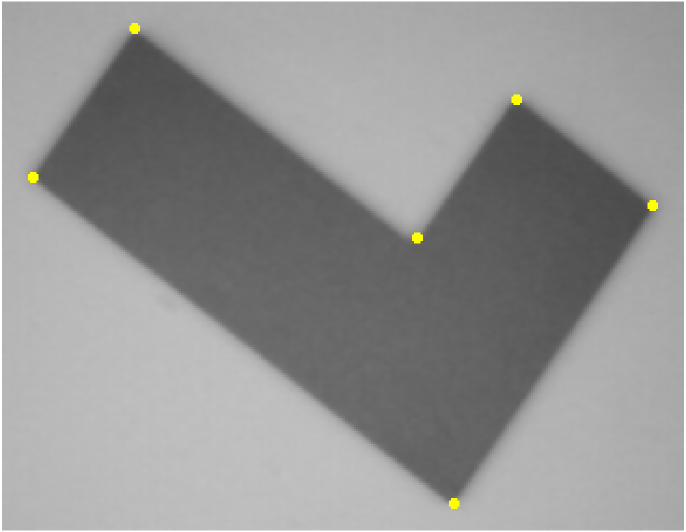
Points and Pointillism

A Computational Perspective

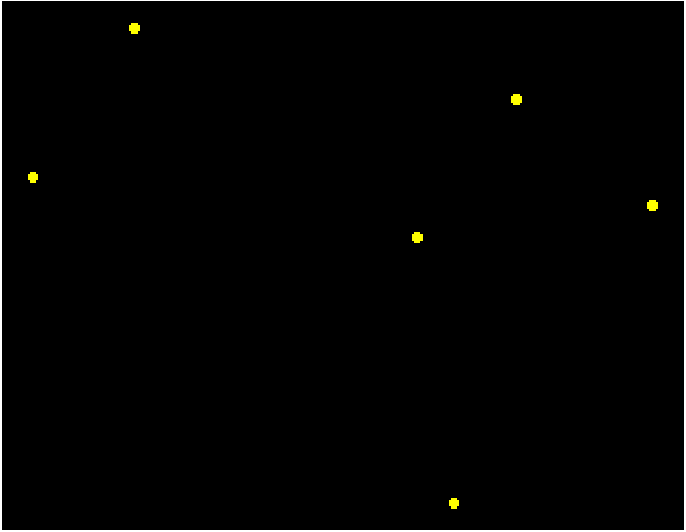
Partha Bhowmick

Associate Professor
CSE Department
IIT Kharagpur

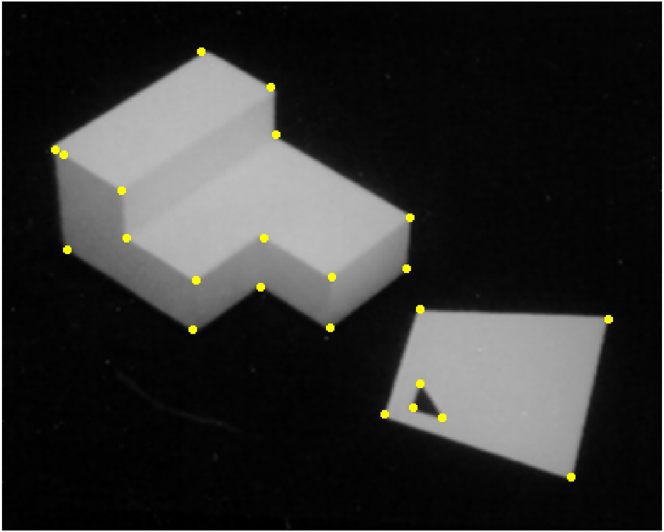
Corners as Points



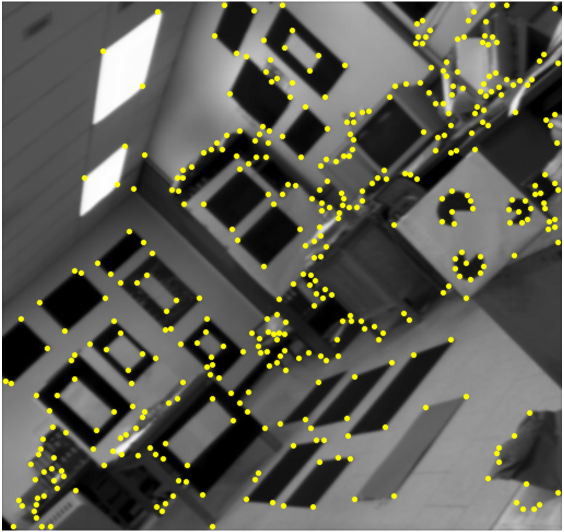
Corners as Points



Corners as Points



Corners as Points



Corners as Points

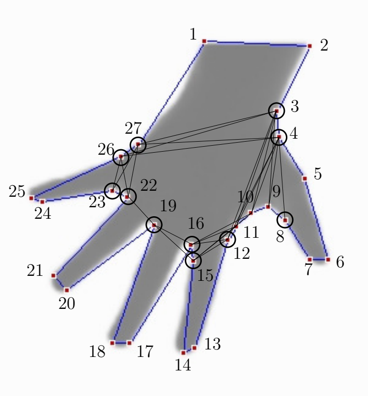
Applications

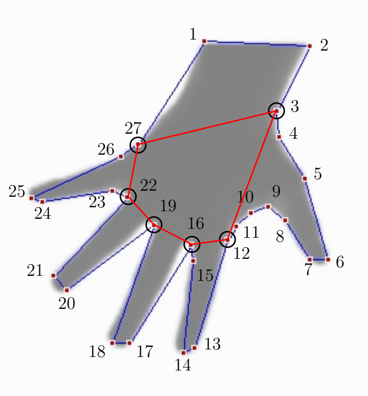
1. shape analysis
2. tracking and classification of moving vehicles
3. optical flow computation
4. 3D scene analysis and reconstruction from stereo image pairs
5. face tracking and face recognition
6. retrieval of images and videos

etc.

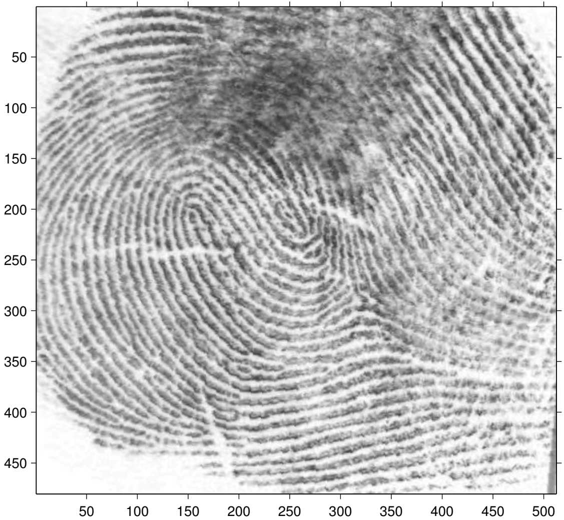
Corners as Points

Shape Analysis

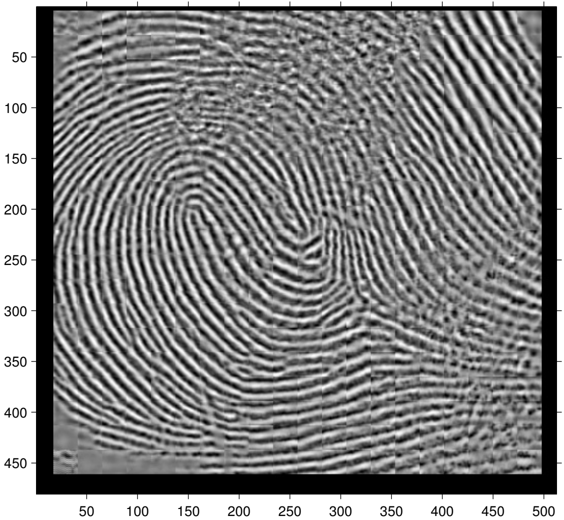




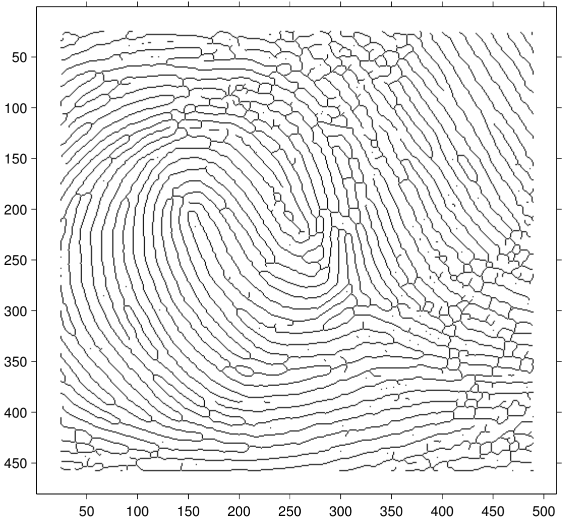
Minutiae as Points



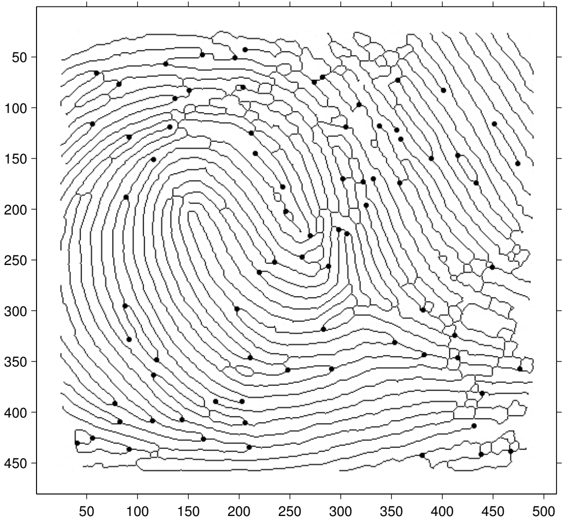
Minutiae as Points



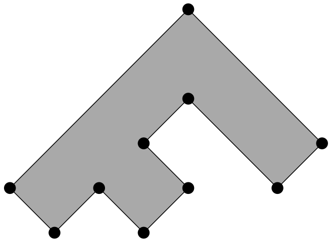
Minutiae as Points



Minutiae as Points

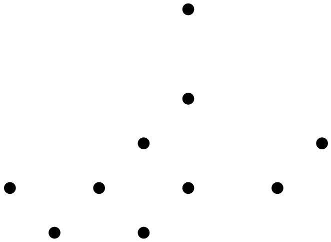


Point Set Pattern Matching



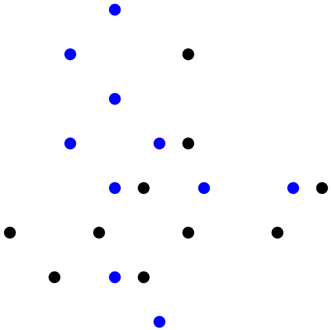
Object corners

Point Set Pattern Matching



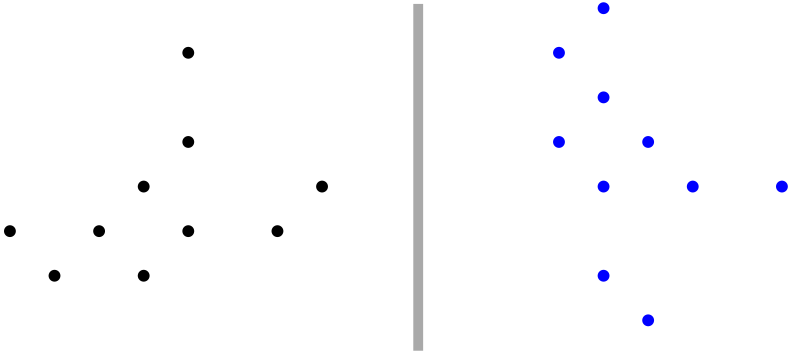
Point set

Point Set Pattern Matching



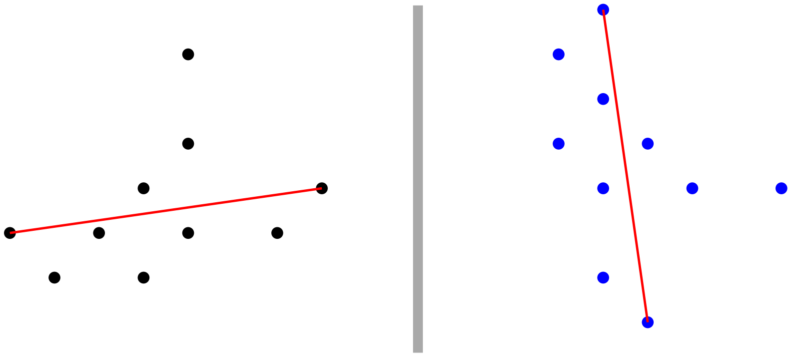
Q: Does the blue point set match the black one? What's the transformation?

Point Set Pattern Matching



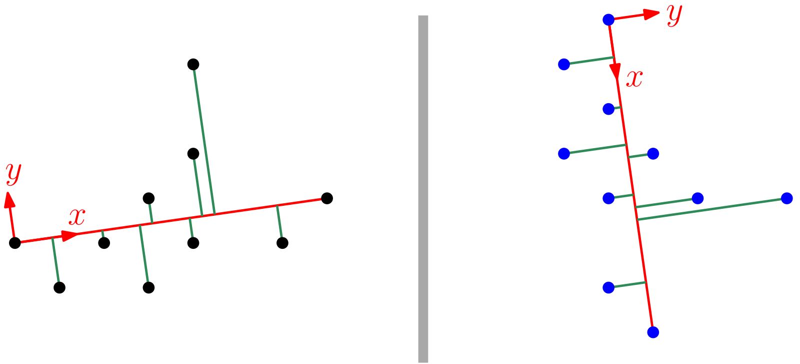
Treat them separately in proper local coordinate system.

Point Set Pattern Matching



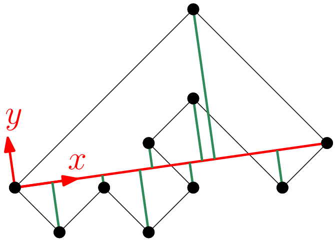
Consider the longest vectors (red lines).

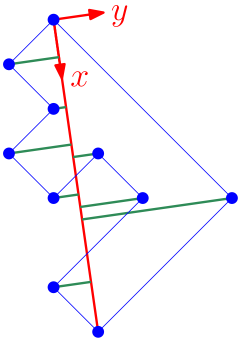
Point Set Pattern Matching



Define the local coordinate systems and compare the recomputed point coordinates.

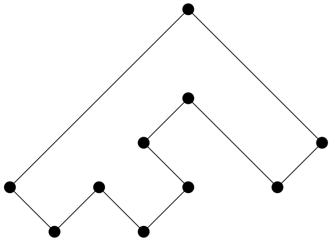
Point Set Pattern Matching

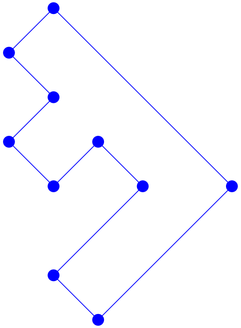




Report the match.

Point Set Pattern Matching



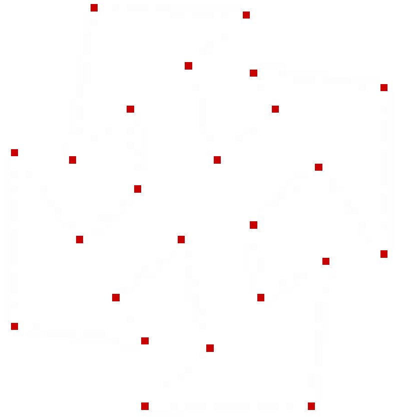


Redraw the objects if needed.

Pointillism

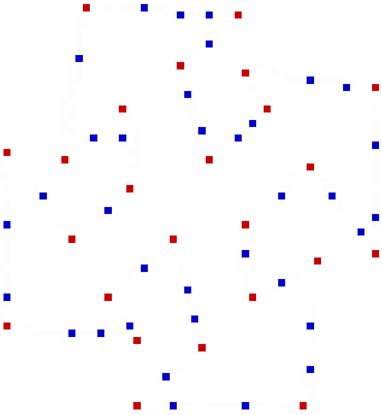
Our algorithmic artwork *(in progress)*

Unordered Point Set



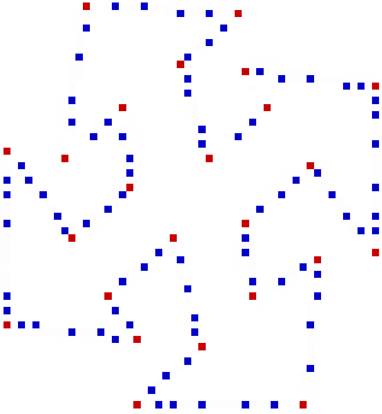
Object corners — Too few to reconstruct

Unordered Point Set



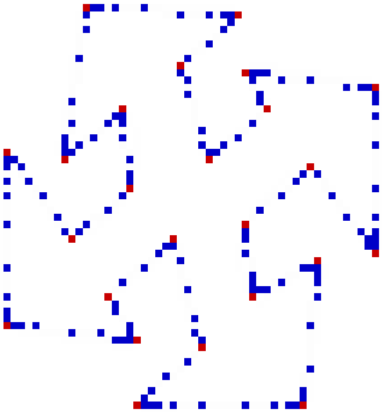
Sufficient?

Unordered Point Set



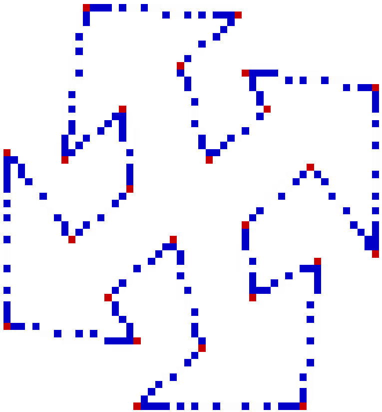
Sufficient???

Unordered Point Set



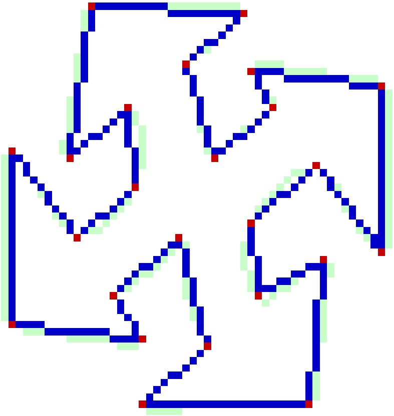
Yes, sufficient! (Pointillist factor $\phi = 1$)

Unordered Point Set



More than sufficient (Pointillist factor $\phi = 2$)

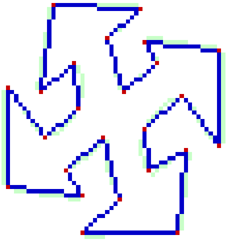
Unordered Point Set

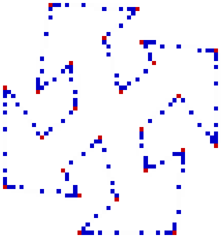


Reconstruction

The idea

- Use the nearest neighbor (NN) rule.
- NN mimics our psycho-visual mechanism.
- Pick an optimal or suboptimal set of points so that reconstruction is possible.

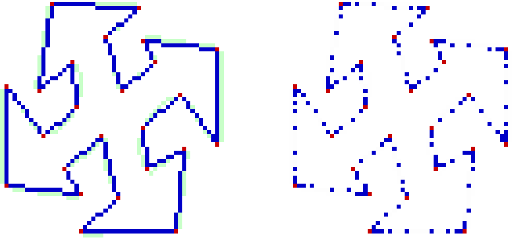




Edge processing

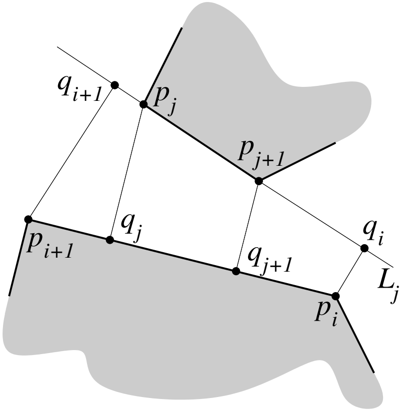
Procedure

Find the minimum distance between two edges e_i and e_j of (same or different) polygon(s).



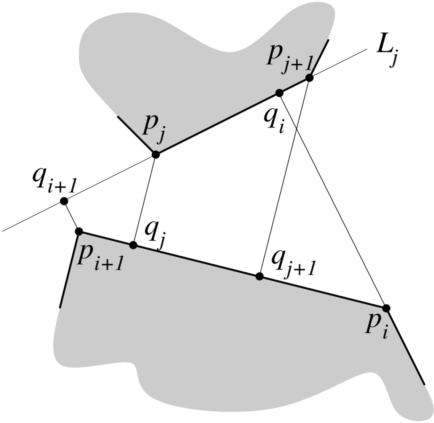
Edge processing

Case 1



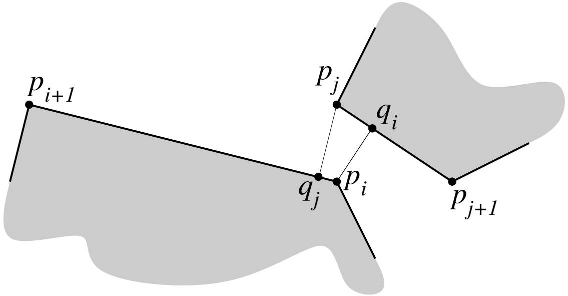
Edge processing

Case 2



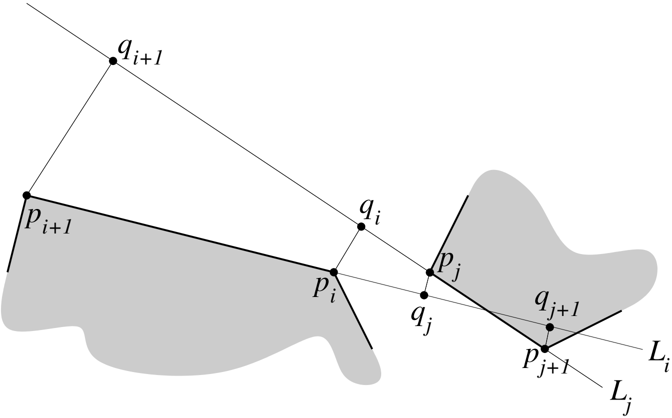
Edge processing

Case 3



Edge processing

Case 4



Reconstruction idea

Facts about **Delaunay triangulation** $DT(S)$ of *any* point set S:

- Each pair of nearest neighbors in S are neighbors in $DT(S)$.
- For the **Euclidean graph**[1] $EG(S)$ of S, the **minimum spanning tree** $MST(EG(S))$ is a subgraph of $DT(S)$.

[1] If S consists of m points, then the vertices of $EG(S)$ are the points in S and the edges are all $\binom{n}{2}$ undirected pairs of distinct points, the weight of each edge being given by the Euclidean distance between the corresponding points.

Reconstruction idea

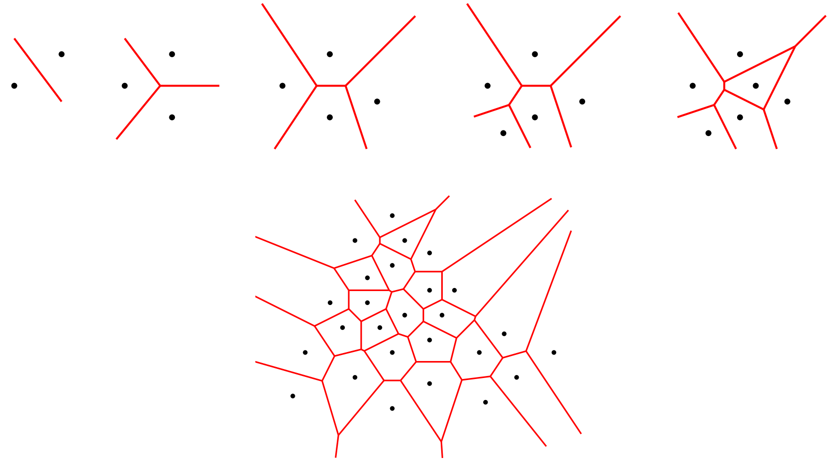
Facts about **Delaunay triangulation** $DT(S)$ of *any* point set S:

- Each pair of nearest neighbors in S are neighbors in $DT(S)$.
- For the **Euclidean graph**[1] $EG(S)$ of S, the **minimum spanning tree** $MST(EG(S))$ is a subgraph of $DT(S)$.

---

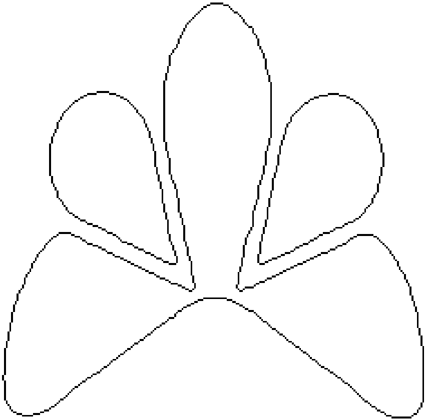
[1] If S consists of m points, then the vertices of $EG(S)$ are the points in S and the edges are all $\binom{n}{2}$ undirected pairs of distinct points, the weight of each edge being given by the Euclidean distance between the corresponding points.

Reconstruction – by *Voronoi diagram*



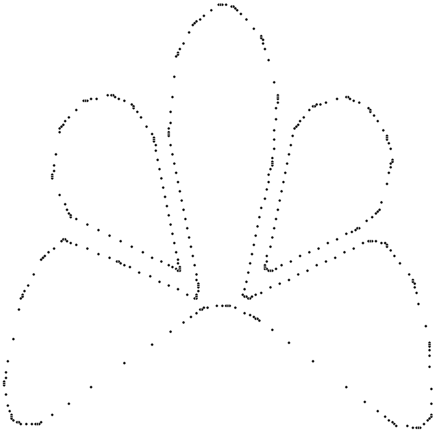
Voronoi diagrams for different point sets

Reconstruction – by *Voronoi diagram*



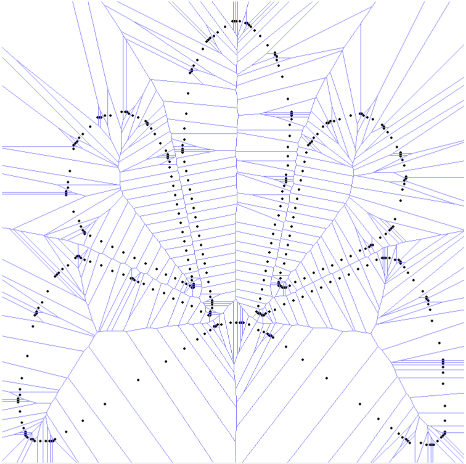
Original contour $\mathcal{C}$

Reconstruction – by *Voronoi diagram*



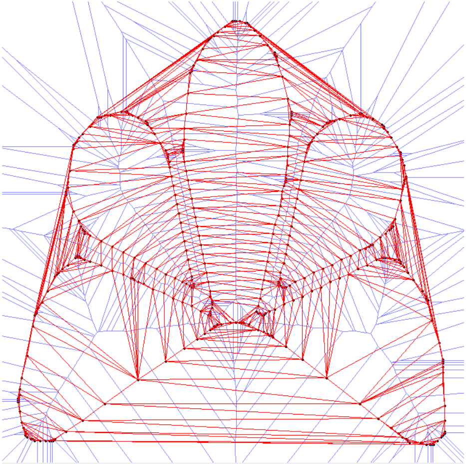
Pointillist ensemble $\widehat{\mathcal{C}}$

Reconstruction – by *Voronoi diagram*



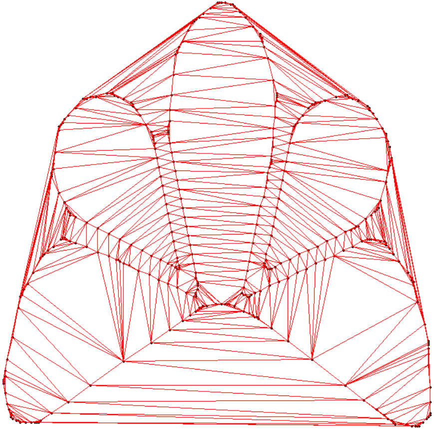
Voronoi diagram, $Vor(\widehat{\mathcal{C}})$

Reconstruction – by *Voronoi diagram*



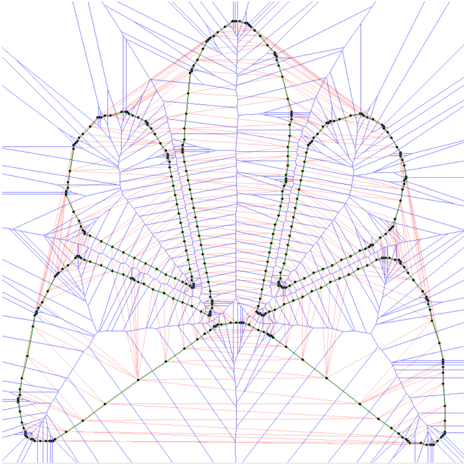
Delaunay triangulation (in red), $DT(\widehat{\mathcal{C}})$, from $Vor(\widehat{\mathcal{C}})$

Reconstruction – by *Voronoi diagram*



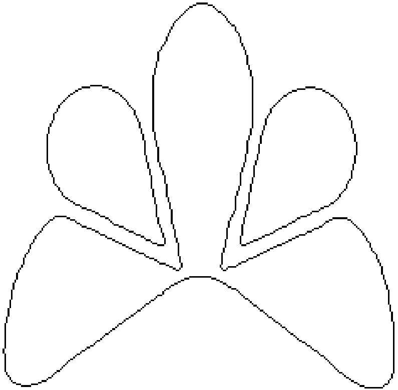
$$DT(\widehat{\mathcal{C}}) = \text{subgraph of Euclidean graph } EG(\widehat{\mathcal{C}})$$

Reconstruction – by *Voronoi diagram*

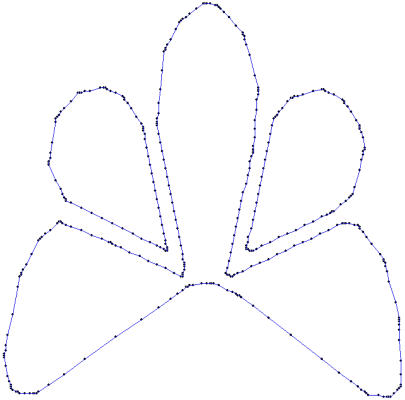


Reconstructed curve (in green) = $MST(DT(\widehat{\mathcal{C}}))$

Reconstruction – by *Voronoi diagram*

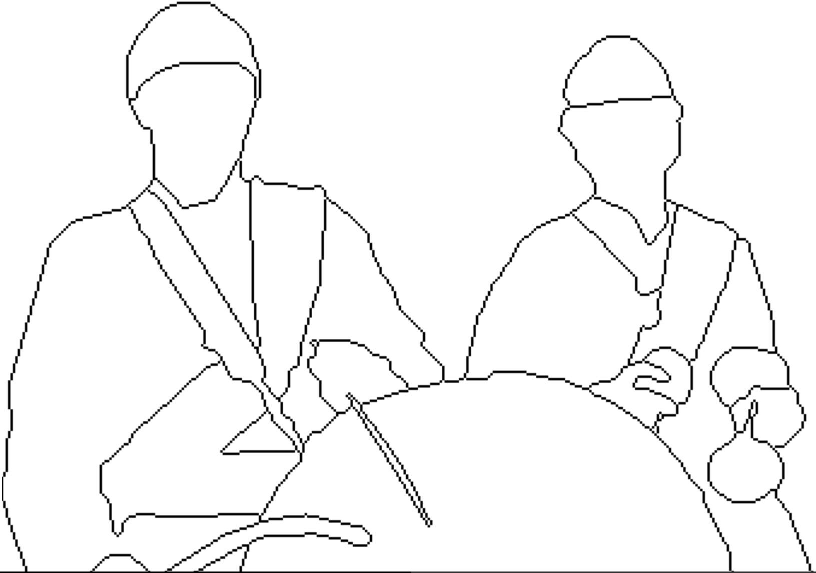


Original

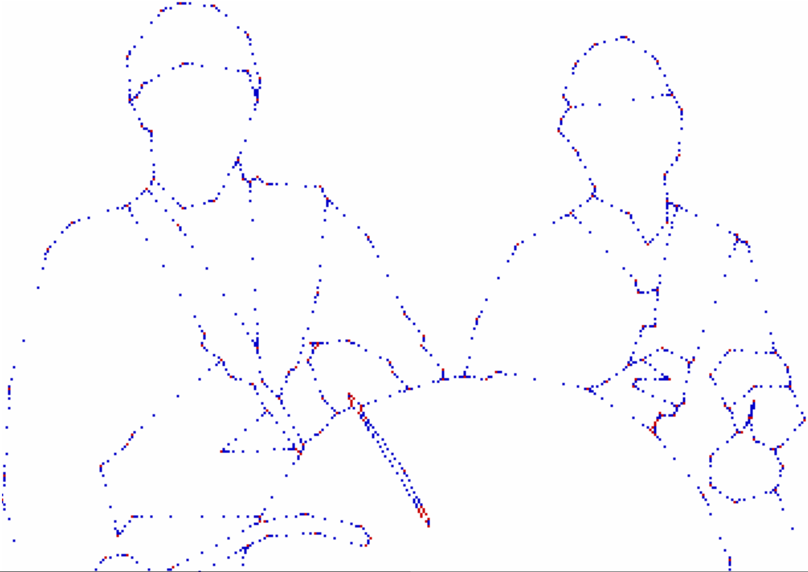


Reconstruction

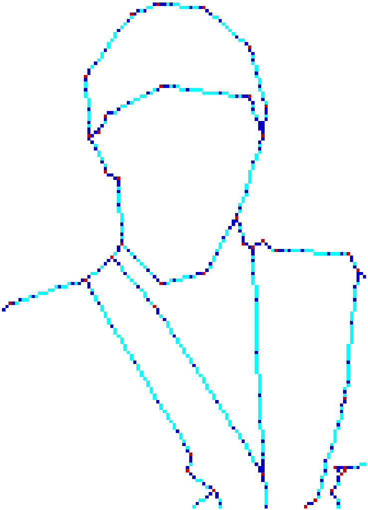
Reconstruction – by *Voronoi diagram*



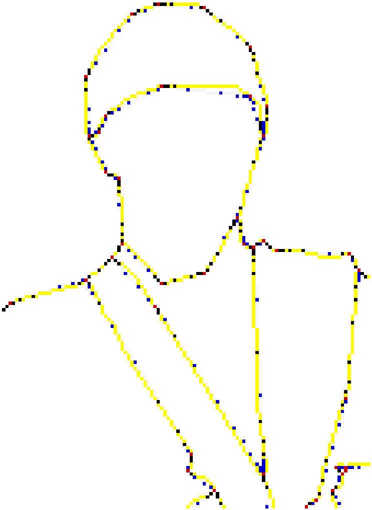
Reconstruction – by *Voronoi diagram*



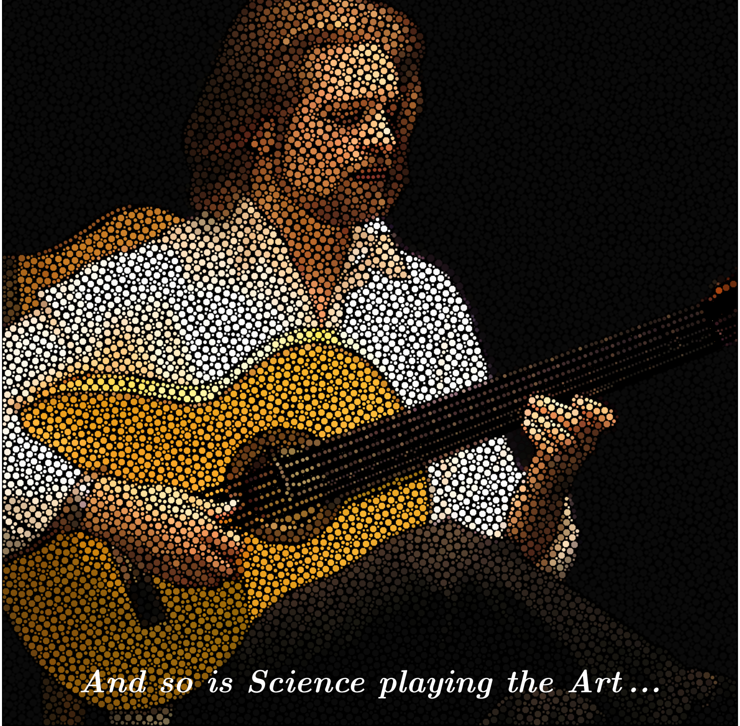
Reconstruction – by *Voronoi diagram*



reconstructed



original


And so is Science playing the Art ...