Outline and Shape Reconstruction in 2D

ECCV 2022 TUTORIAL

Stefan Ohrhallinger, Jiju Peethambaran, Amal Dev Parakkat

Technische Universität Wien, Austria
Saint Mary's University, Canada
LTCI- Telecom Paris, Institut Polytechnique de Paris, France

Tutorial Outline

Intro & Proximity Graphs	Stefan Ohrhallinger - 25 minutes
Curve Reconstruction	Stefan Ohrhallinger - 25 minutes, Q&A 5 minutes
Benchmark & Demo	Amal Dev Parakkat - 25 minutes, break 15 minutes
Sketch Reconstruction	Amal Dev Parakkat - 25 minutes, Q&A 5 minutes
Visual Perception of Shapes	Jiju Peethambaran - 25 minutes
Shape Characterization	Jiju Peethambaran - 25 minutes, Q&A 5 minutes

Topic: Intro & Proximity Graphs

Motivation

Proximity Graphs

Presenter:

Stefan OHRHALLINGER

Researcher

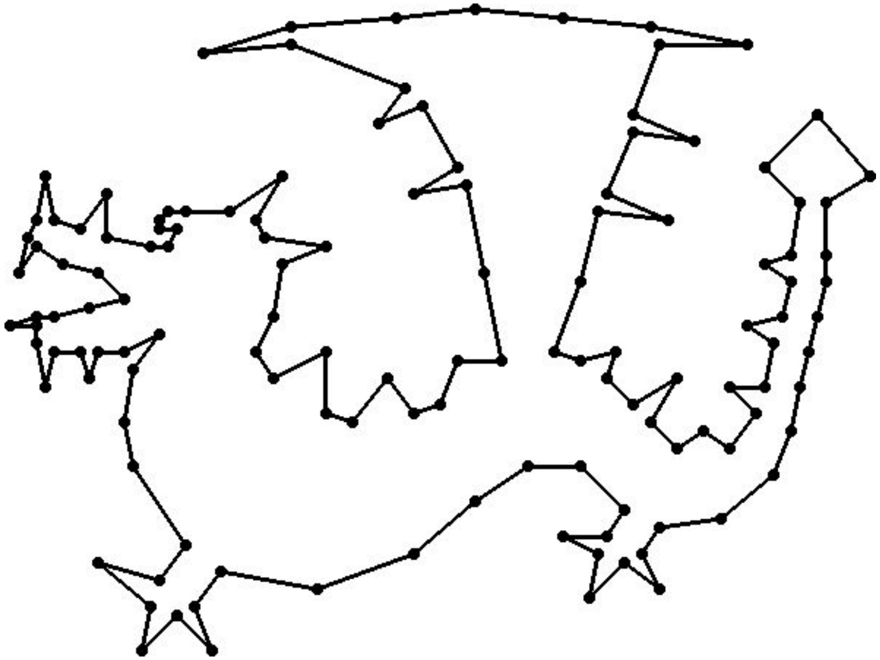
Institute of Visual Computing &
Human-Centered Technology

Introduction

The Problem

Connect the Dots

Now try without the numbers



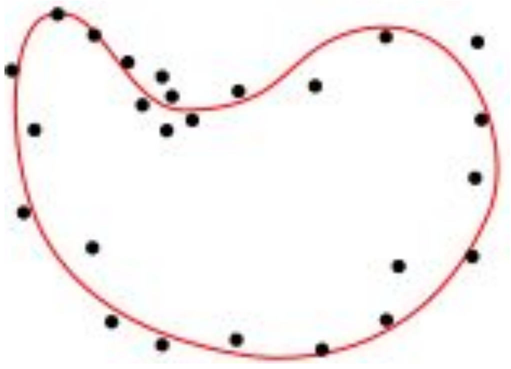
Reconstructed polygon

Challenges for Curve Reconstruction

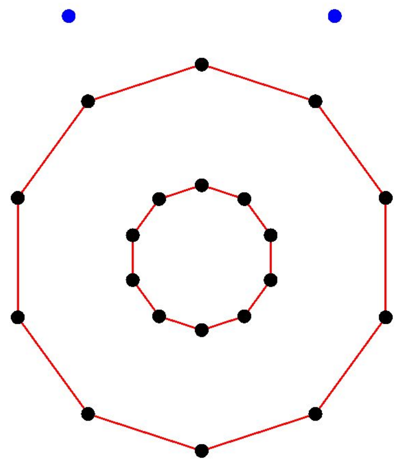
Non-uniform sampling

Sparse sampling

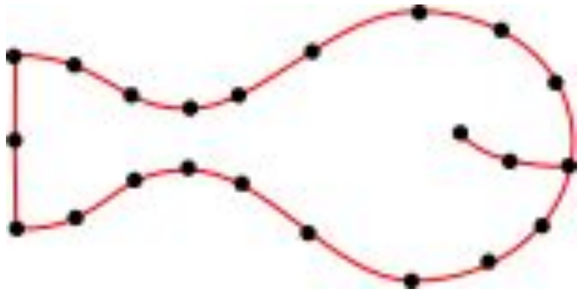
Sharp Corners



Noisy sampling



Outliers, multiple curves

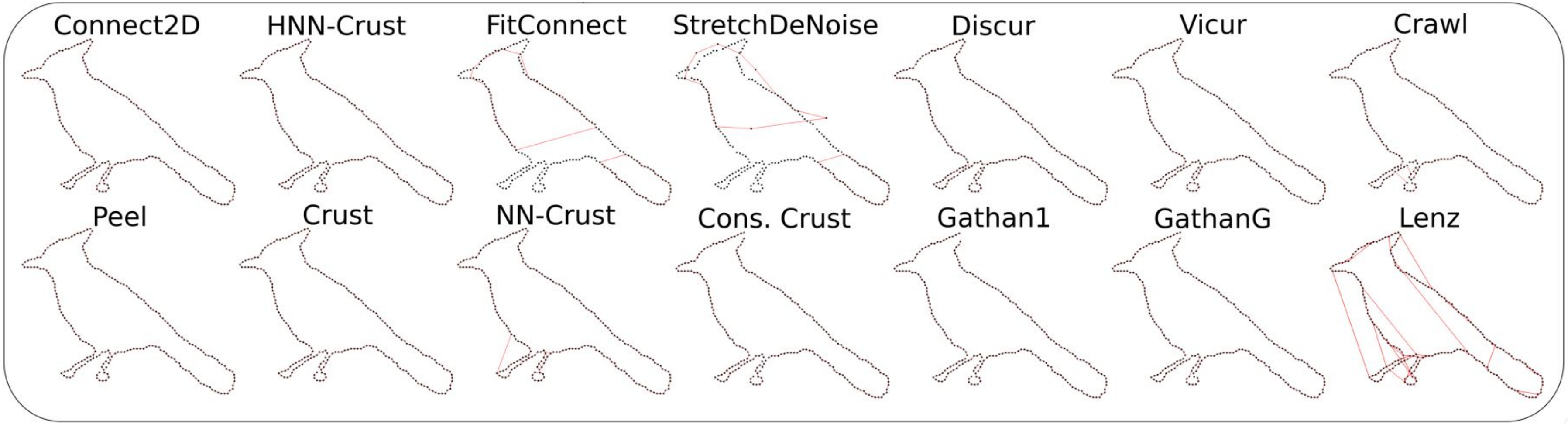


Non-manifold curves

How to Choose a Suitable Algorithm?

DISCUR
VICUR
StretchDenoise
[Lee00a]
α-shapes
Shape-hull Graph
γ-neighborhood
ec-Shape
[AM00]
[Len06]
Connect2D
GathanG
NN-Crust
Crust
Peel
FitConnect
[CFG*05]
[Rup14]
β-skeleton
Crawl
r-regular shapes
[WYZ*14]
Gathan
HNN-Crust
Concorde
Conservative Crust
[FR01]
EMST
Ball-pivoting
Optimal Transport
Robust HPR
[Gie99]
[Aro98]
[Hiy09]
Voronoi Labeling
Edge exchanging

A Benchmark Helps to Decide [OPP*21]

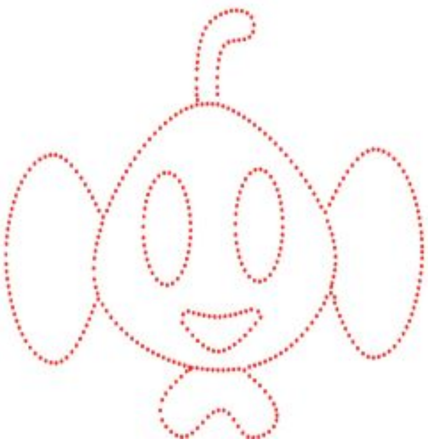


Evaluating algorithms on challenging curves, highlighting strengths & weaknesses

Quantitative analysis on: reconstruction quality & run-time

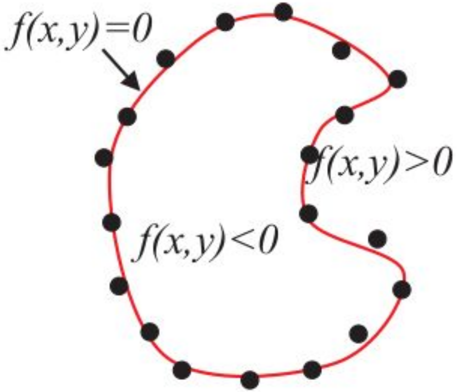
Scope of this Tutorial

We categorize 36 curve reconstruction algorithms:



Boundary samples

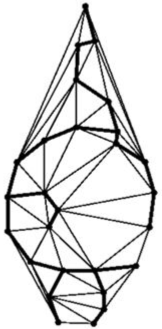
Area samples



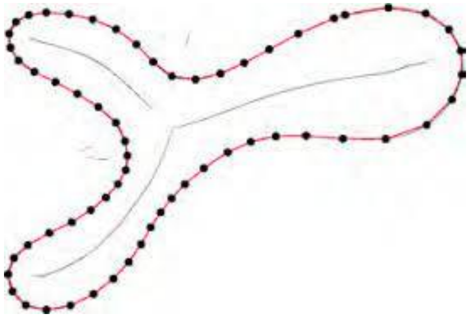
Implicit curve

Polygonal curve

Taxonomy of Algorithms

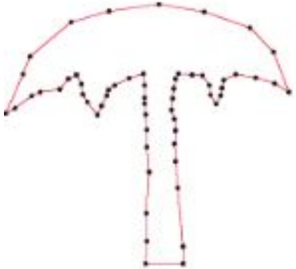


Graph-based

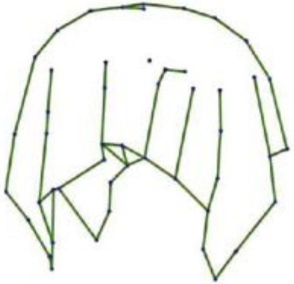


Feature size based

Noisy fitting

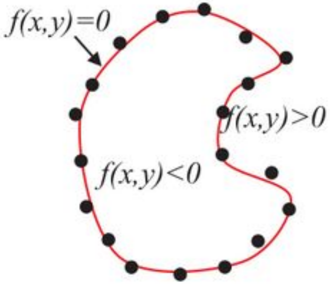


Sharp corners



Non-manifold

HVS-based



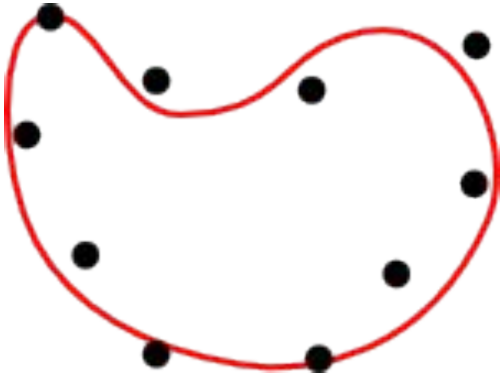
Implicit

Region Reconstruction

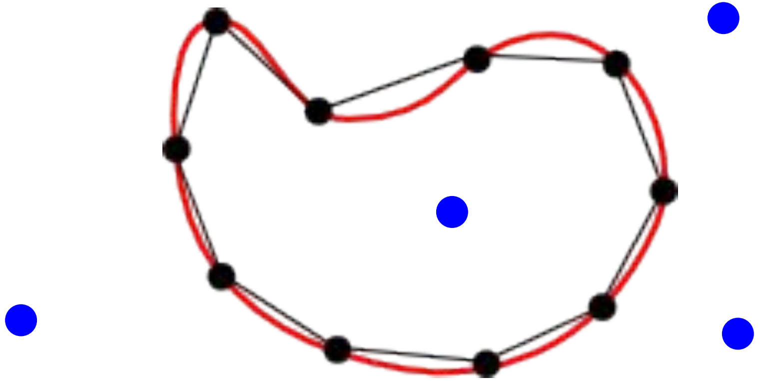
10

Input Data: Properties

Non-uniform sampling: determines feature size

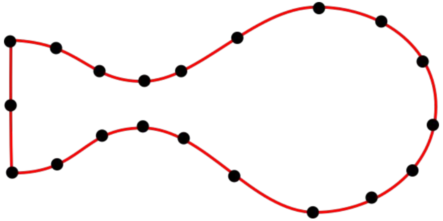


Noisy sampling: needs fitting



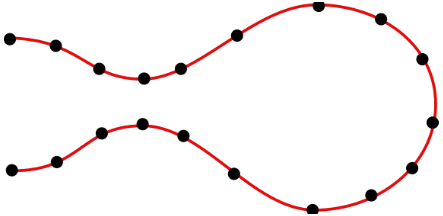
Outliers: needs filtering

Reconstruction Output: Properties



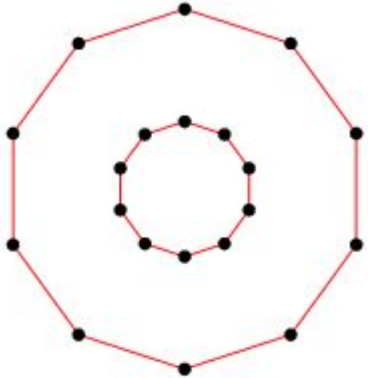
deg(v)=2

Manifold

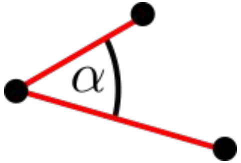


deg(v)≤2

Open curves



Multiply Connected



$\alpha < 90^o$

Sharp Corners

$\varepsilon < 0.5$

Guarantees

$O(n \log n)$

Time Complexity

Input capabilities: e.g., noise, outliers, non-uniformity

DISCUR
VICUR
StretchDenoise
[Lee00a]
α-shapes
Shape-hull Graph
γ-neighborhood
ec-Shape
[AM00]
[Len06]
Connect2D
GathanG
NN-Crust
Crust
Peel
FitConnect
[CFG*05]
[Rup14]
β-skeleton
Crawl
r-regular shapes
[WYZ*14]
Concorde
Gathan
HNN-Crust
Conservative Crust
[FR01]
EMST
Ball-pivoting
Optimal Transport
Robust HPR
[Gie99]
[Aro98]
Voronoi Labeling
[Hiy09]
Edge exchanging

Output capabilities: e.g., manifold, sharp, O(n log n)

TEL AVIV 2022

DISCUR

VICUR

StretchDenoise

[Lee00a]

α-shapes

Shape-hull Graph

γ-neighborhood

ec-Shape

[AM00]

[Len06]

Connect2D

GathanG

NN-Crust

Crust

Peel

FitConnect

[CFG*05]

[Rup14]

β-skeleton

Crawl

r-regular shapes

[WYZ*14]

Concorde

Gathan

HNN-Crust

Conservative Crust

[FR01]

EMST

Ball-pivoting

Optimal Transport

Robust HPR

[Gie99]

[Aro98]

Voronoi Labeling

[Hiy09]

Edge exchanging

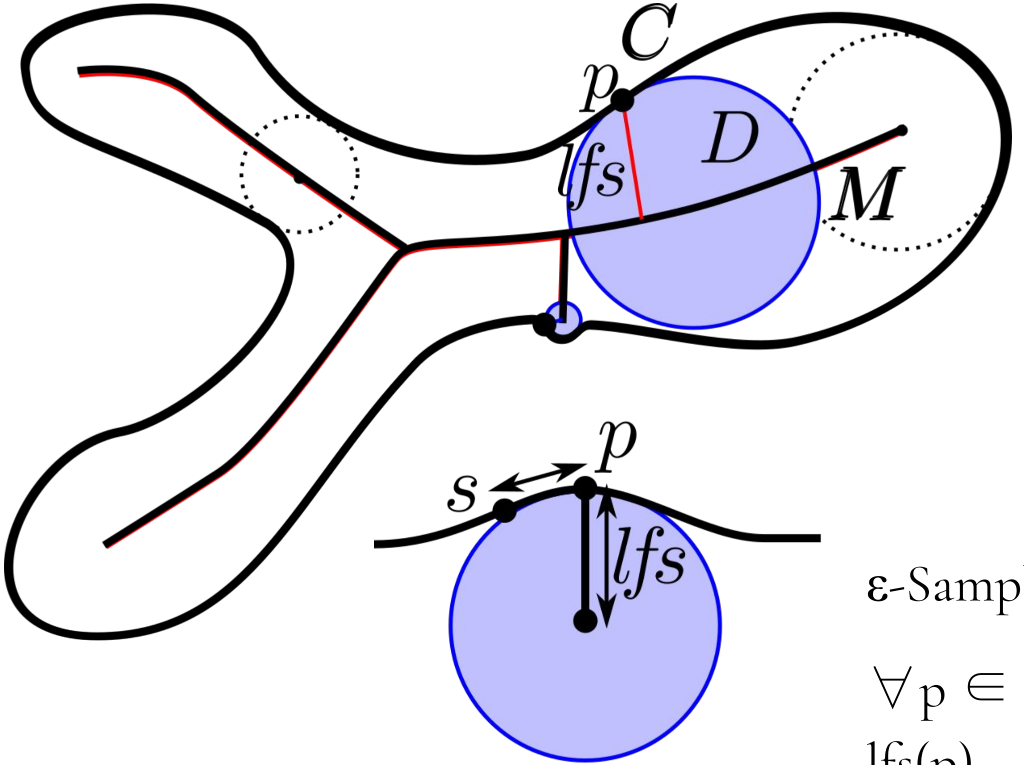
Definitions Curve

Curve Σ: Simple closed and planar

Smooth curve C: (collection of) twice-differentiable bounded 1-manifolds $\in \mathbb{R}^2$

Sample set P: n points sampled on Σ or C

Definitions Sampling



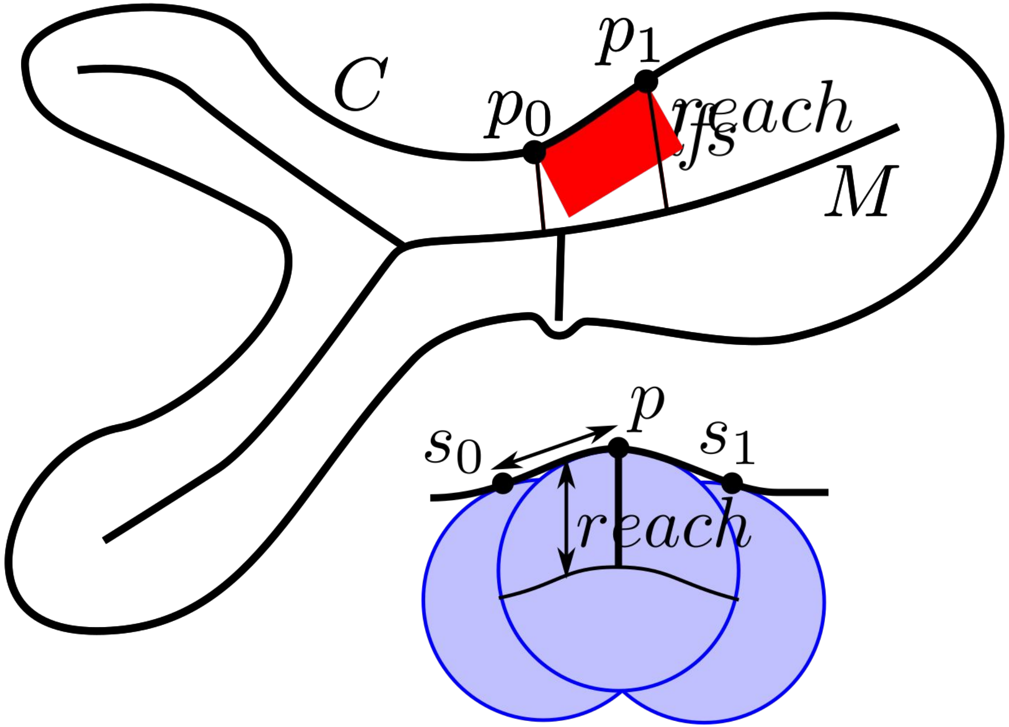
Medial axis M for C [Blum67]:
Closure of all points in $\mathbb{R}^2$
with ≥2 closest points in C

Local feature size lfs(p) [Rup93]:
Euclidean distance from p to
its closest point m ∈ M

ε-Sampling [ABE98]:

$\forall p \in C, \exists s \in S: \|p, s\| < \varepsilon\, \mathrm{lfs}(p)$

Definitions Sampling

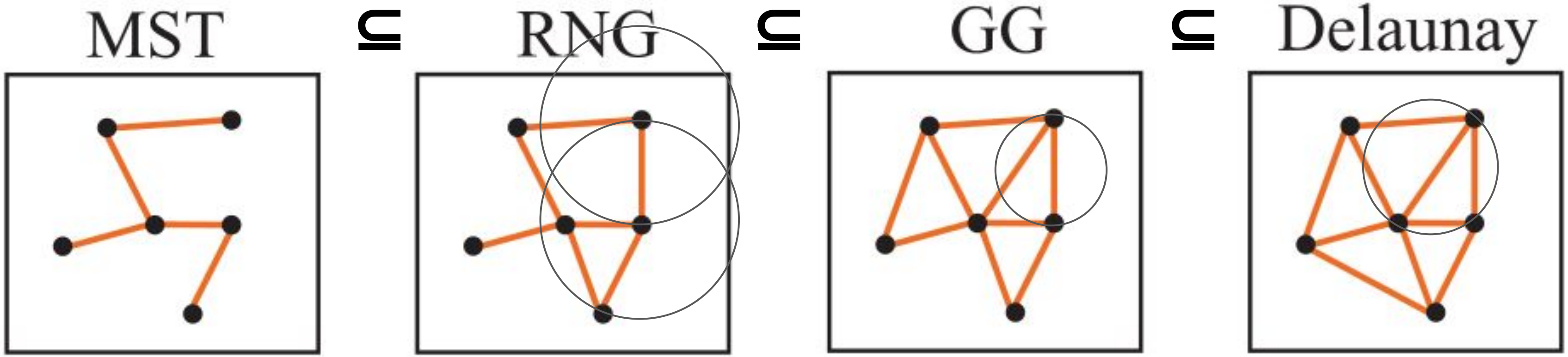


Reach of a curve interval I:
inf lfs(p) : p ∈ I [OMW16]

ρ-Sampling [OMW16]:

∀ p ∈ C, ∃ s ∈ S : ‖p, s‖ < ρ reach(p)

Proximity Graphs for a Point Set



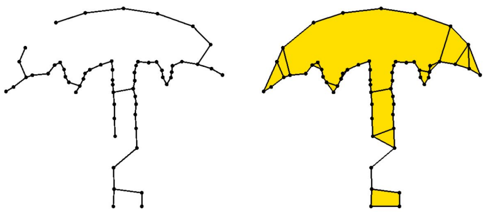
Minimum Spanning Tree: cycle-free graph spanning P with minimum edge weights

Relative Neighborhood Graph: $\forall$(p,q): d(p, q)≤d(p, x), d(p, q)≤d(q, x) $\forall$x $\in$ P, x≠p,q

Gabriel Graph: All (p,q) with p,q $\in$ empty ball centered at (p,q)

Delaunay Triangulation: circumcircles empty of P

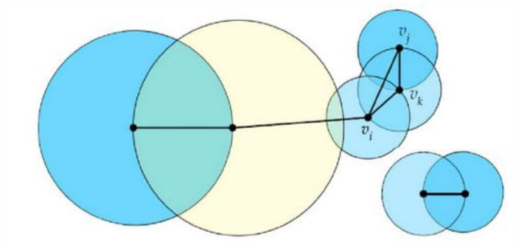
More Proximity Graphs



EMST (d≥1) -> BC_0 (d≥2)

Boundary Complex

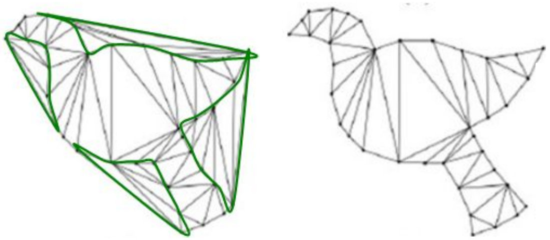
Connect2D [OM13]



SIG edges: r=|v,NN1| overlap

Sphere-of-Influence Graph

[Toussaint88]



DT ∖ divergent concave

Shape-Hull Graph

[PM15]

Topic: Curve Reconstruction

Graph-based Algorithms

Feature size based Algorithms

Presenter:

Stefan OHRHALLINGER

Researcher

Institute of Visual Computing &
Human-Centered Technology

Algorithms Based on Graphs - Overview

α-shapes [EKS83], Ball-pivoting [BB97]

β-skeleton [KR85]

γ-neighborhood [Vel92]

Sculpting [Boi84a]

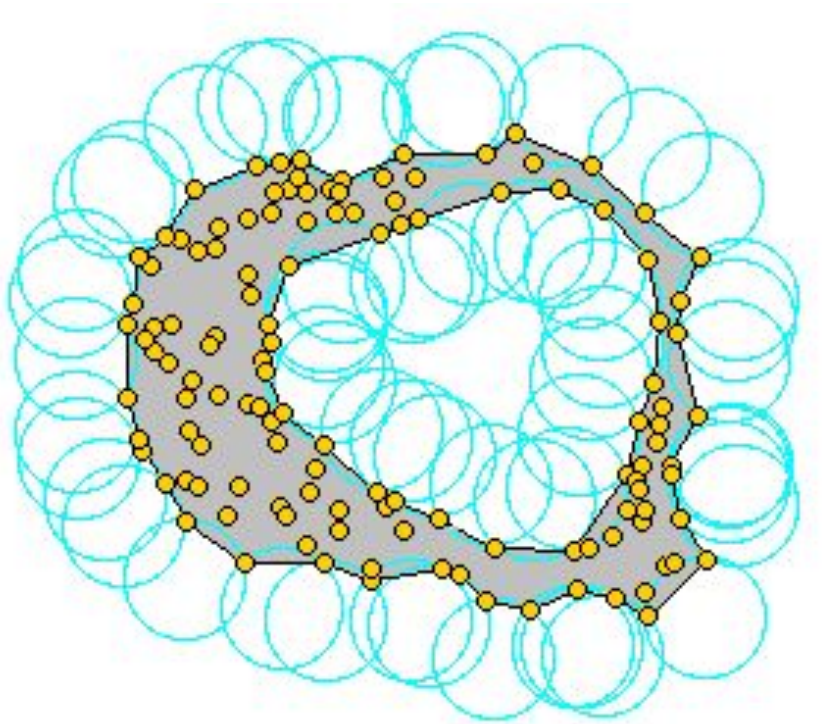
EMST [FMG94], edge exchange [OM11] and inflating [OM13]

r-regular shape [Att97]

Shape-hull graph [PM15b], Voronoi labeling [PPT*19]

Crawl thru neighbors [PM16]

α-Shapes [EKS83]



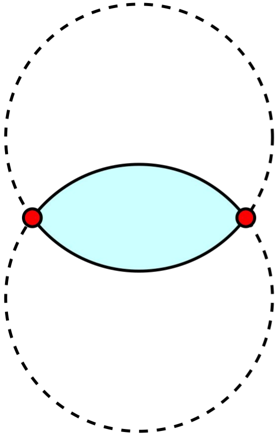
Disks of radius 1/α

Generalization of convex hull (α=0)

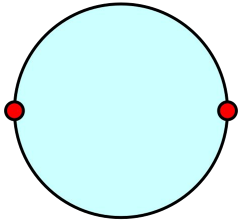
Extracting manifolds [BB97]

Later: Ball-pivoting algorithm [BMR*99]

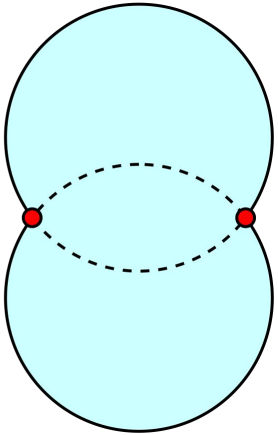
β-Skeleton [KR85]



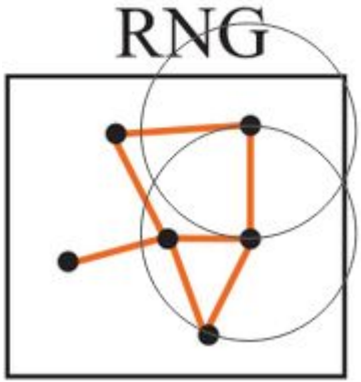
β<1



β=1



β>1



β=2: Relative neighborhood graph

Empty lens formed with ∡prq<θ

= Intersection/Union of disks

$$\theta = \begin{cases} \sin^{-1}\frac{1}{\beta}, & \text{if } \beta \geq 1 \\ \pi - \sin^{-1}\beta, & \text{if } \beta \leq 1 \end{cases}$$

γ-Neighborhood [Vel92]

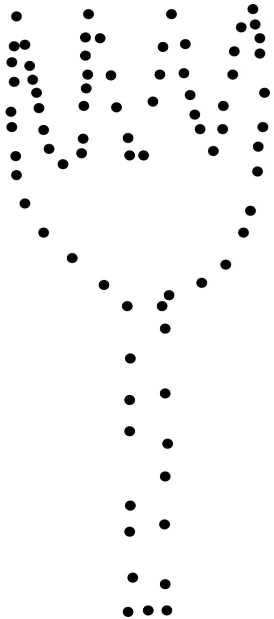
A unification of 12 graphs including convex hull, Delaunay triangulation, Gabriel graph, RNG, MST, nearest neighbor graph, α-shapes and β-skeletons.

$\gamma(\gamma_0,\gamma_1)$ is defined for $-1<\gamma_0,\gamma_1<1$ and $|\gamma_0|\leq|\gamma_1|$

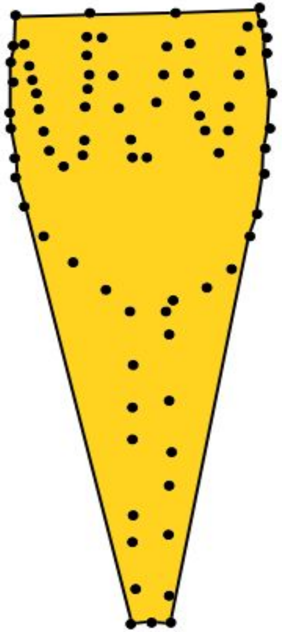
Contains edges with empty neighborhood defined by disks using γ_0,γ_1

It can also reconstruct shapes not in the Delaunay graph

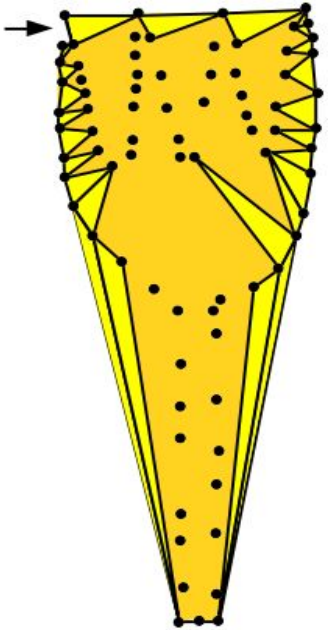
Sculpting [Boi84a]



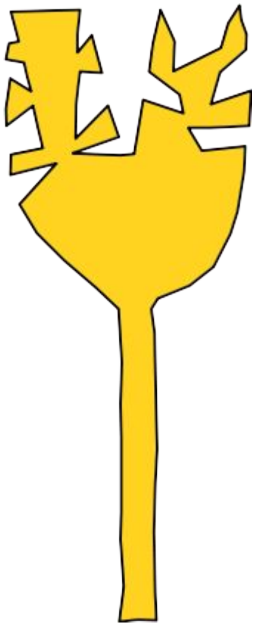
Points



Convex hull

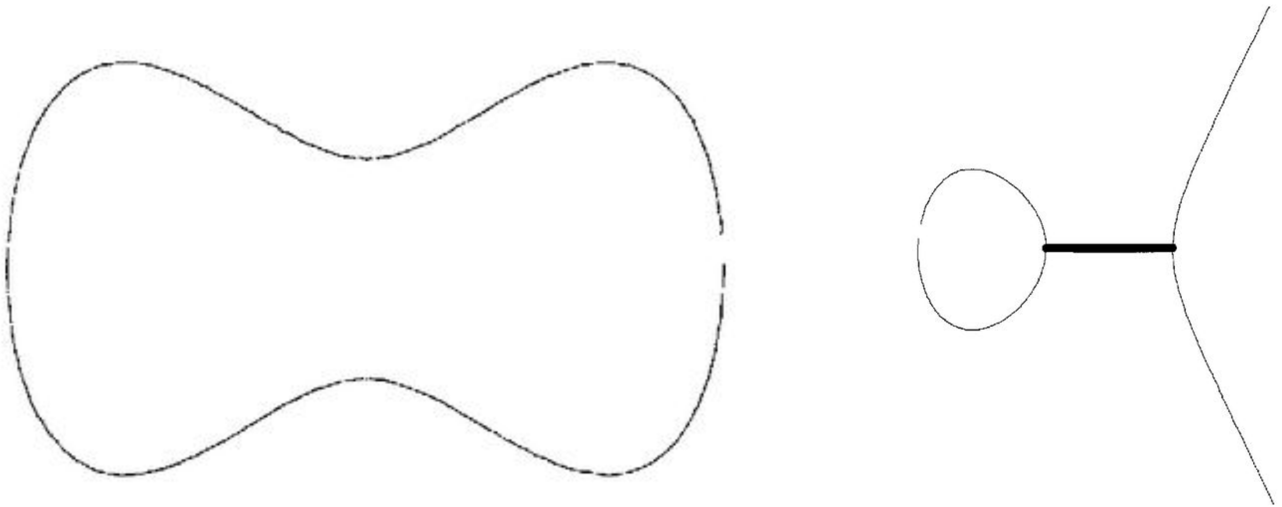


Sculpting step



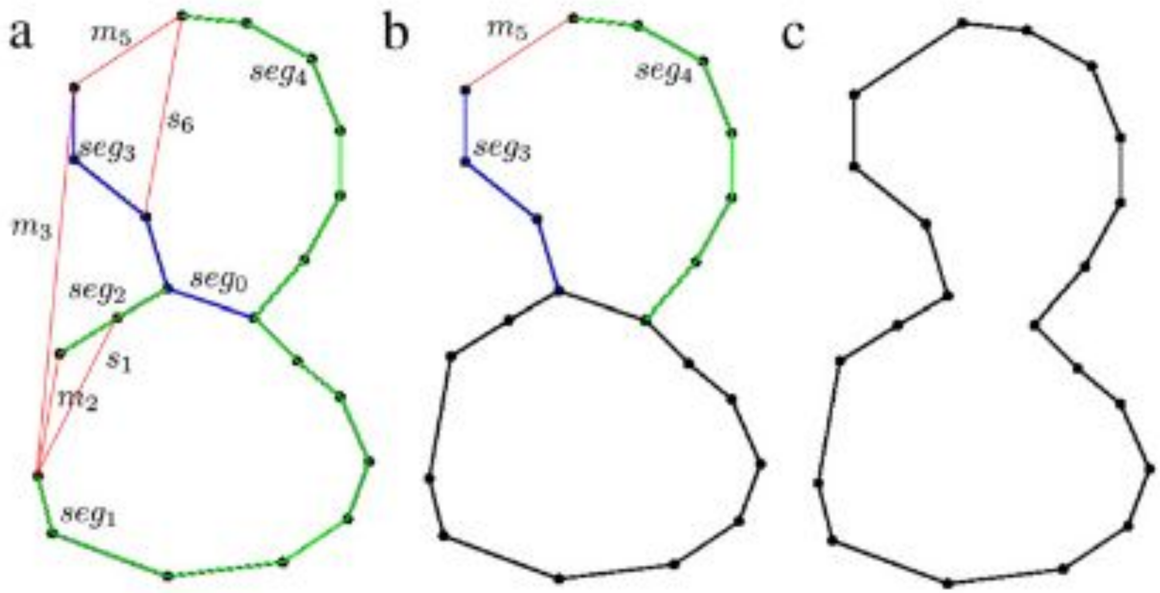
Final boundary

EMST-based Reconstruction [FMG94]



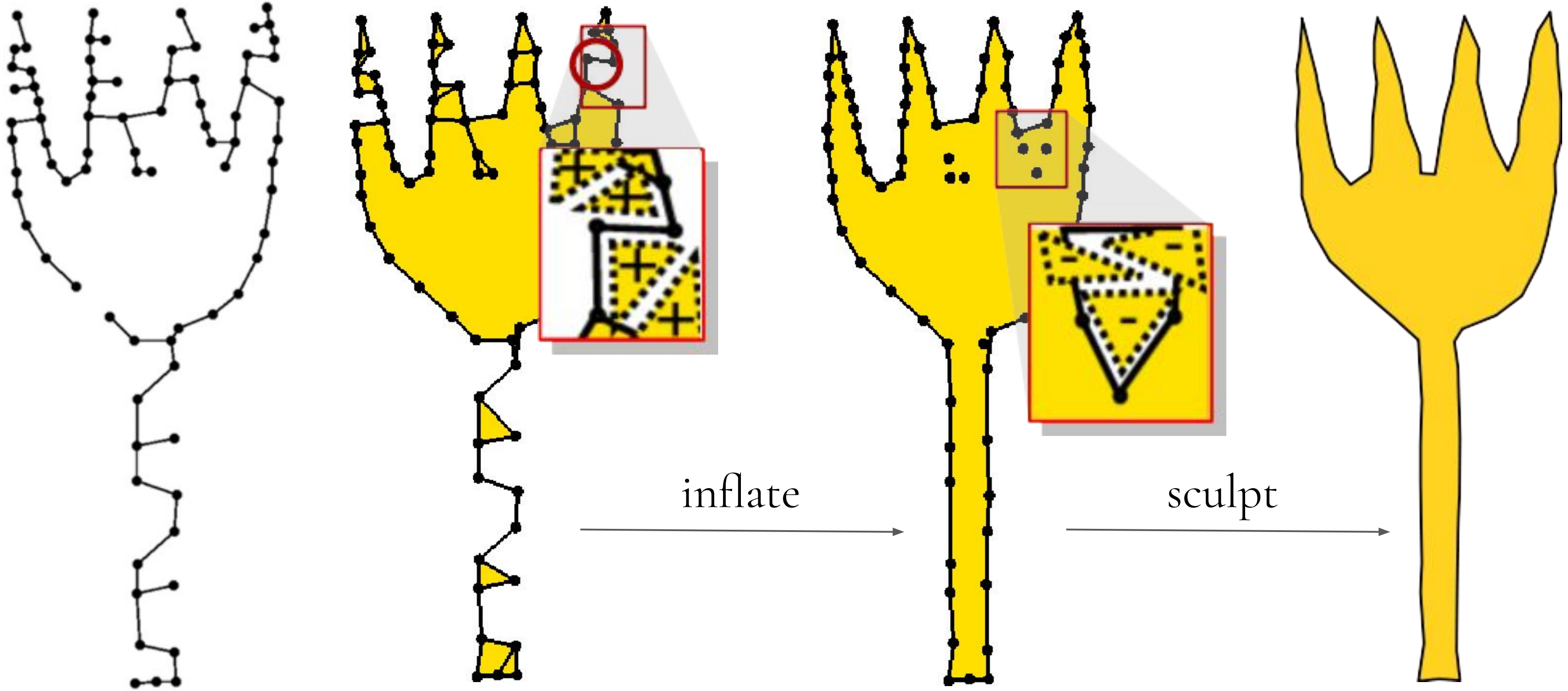
Proves that EMST reconstructs (open) curve from sufficiently dense samples

EMST-based Edge-Exchange Reconstruction [OM11]



Transform EMST with “snap” and “move” operations - combinatorial complexity

EMST-based Inflating Reconstruction [OM13]



EMST | BC: deg(v)≥2 | deg(v)≤2 | deg(v)=2

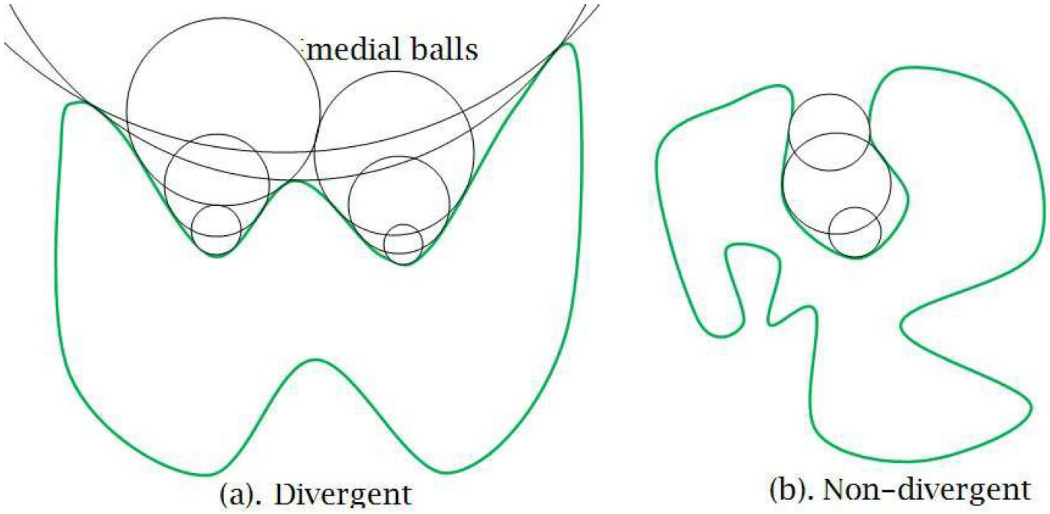
r-regular Shapes [Att97]

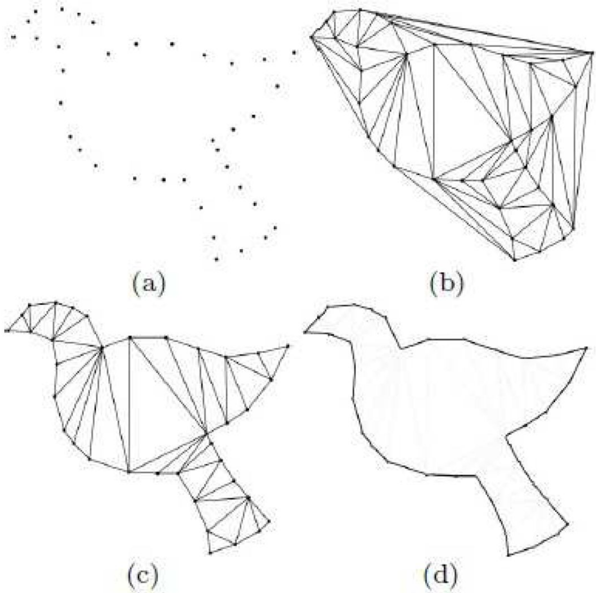
An r-regular shape has curvature $\geq r$ everywhere

Requires uniform sampling of boundary

Boundary consists of edges shared by Delaunay circumcircles with property of angle<threshold depending on uniform sampling density and curvature r

Shape-Hull Graph [PM15b]

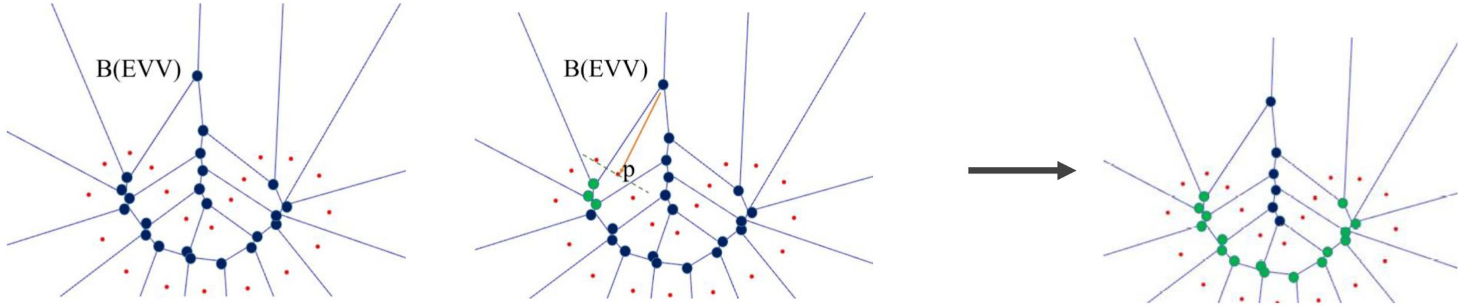




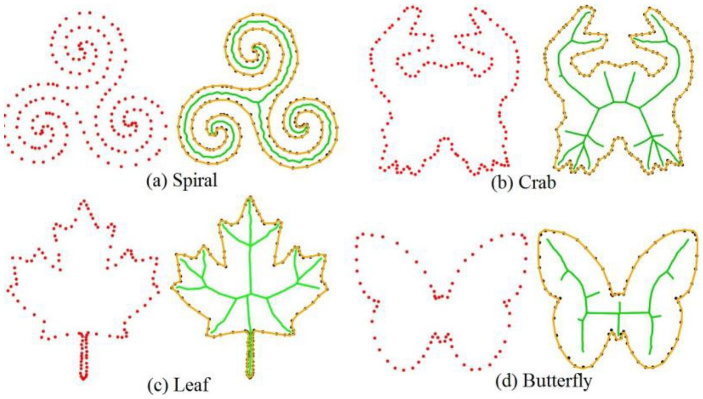
Reconstructs smooth curves with divergent concavity

Eliminates Delaunay triangles with circumcenter outside boundary

Voronoi Labeling [PPT*19]



Incrementally labels orientation from estimated normals via Voronoi poles

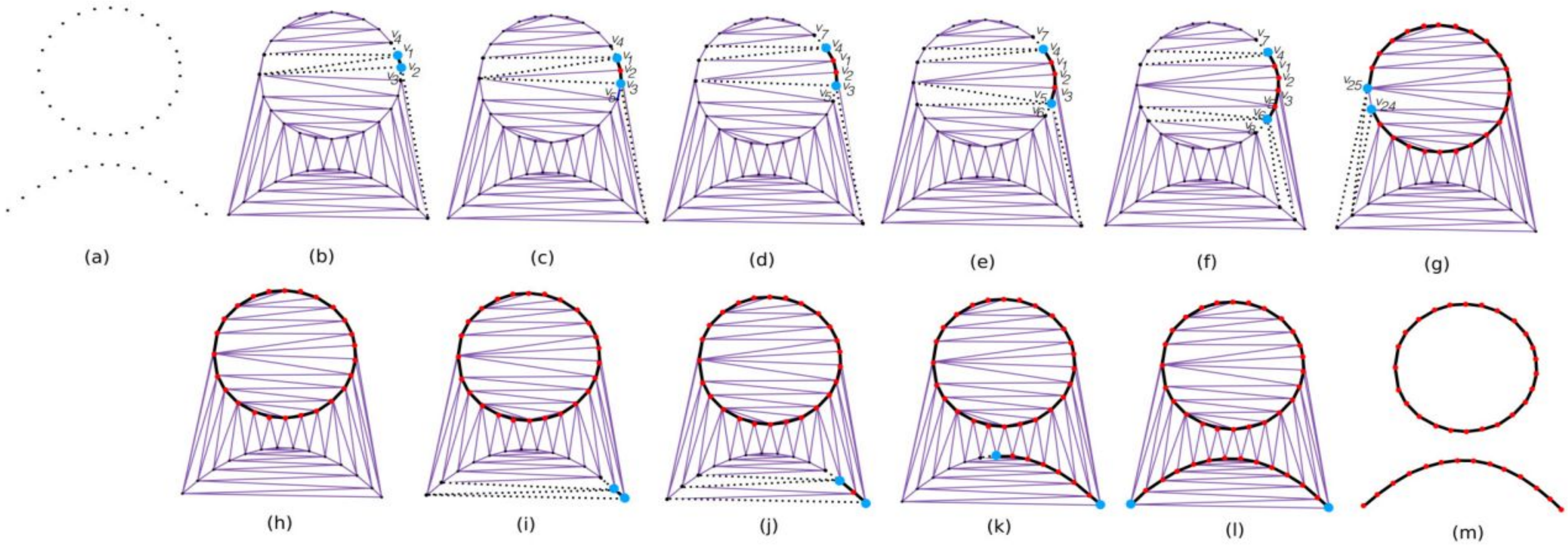


Guaranteed ε-sampling as well as

bi-tangent neighborhood convergence

Also computes medial axis

Crawl Thru Neighbors [PM16]



Connects neighbors greedily, heuristic decides curve closed/open

Parameter-free: handles open+multiple curves, holes and outliers

Algorithms Based on Graphs - Conclusion

α-shapes [EKS83], Ball-pivoting [BB97], β-skeleton [KR85], γ-neighborhood [Vel92], Sculpting [Boi84a], EMST [FMG94], edge exchange [OM11], inflating [OM13], *r*-regular shape [Att97], Shape-hull graph [PM15b], Voronoi labeling [PPT*19], Crawl thru neighbors [PM16]

They often require a global parameter

Good results mostly for uniformly sampled point density

Delaunay graph is not guaranteed to contain the reconstruction

Reconstruction is often slow or trapped in local minima

Algorithms Based on Feature Size - Overview

Crust [ABE98]

Anti-Crust [Gol99]

NN-Crust [DK99]

Conservative Crust [DMR99]

Lenz [Len06]

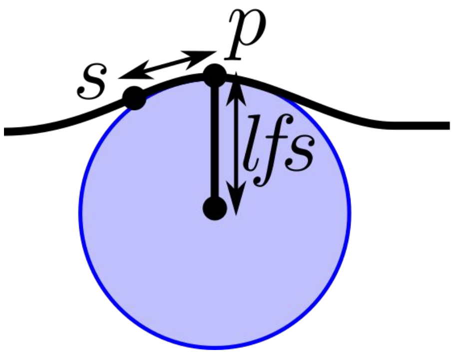
Hiyoshi [Hiy09]

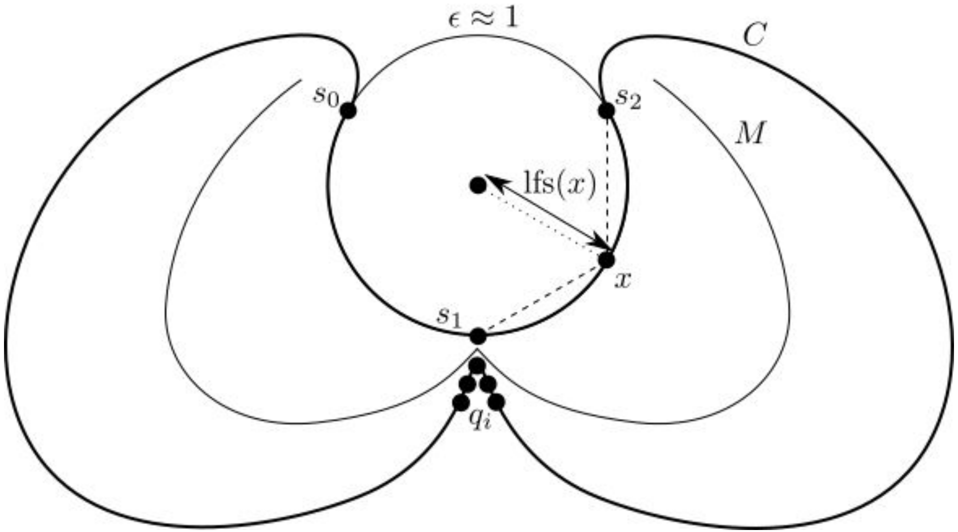
HNN-Crust [OMW16]

SIGDT [MOW22]

Crust [ABE98]

Seminal paper: feature sized reconstruction - no more uniform sampling required





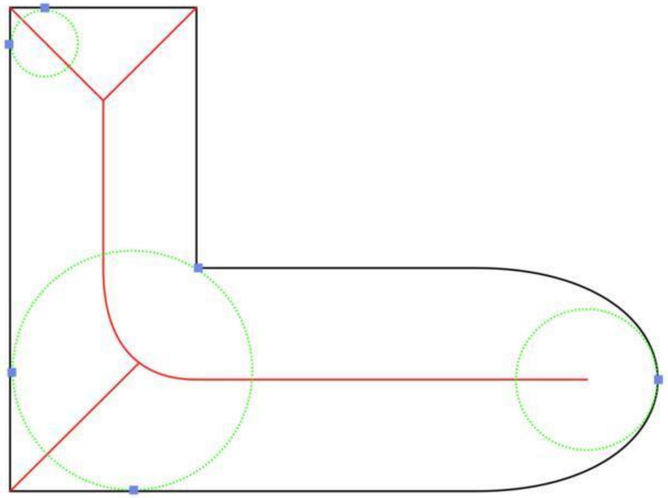
ε-Sampling [ABE98]:
∀p ∈ C, ∃s ∈ S :
||p, s|| < ε lfs(p)

extracts DG and Voronoi graph

Proof: ε < 0.252 ≙ α > 151°

Anti-Crust [Gol99]

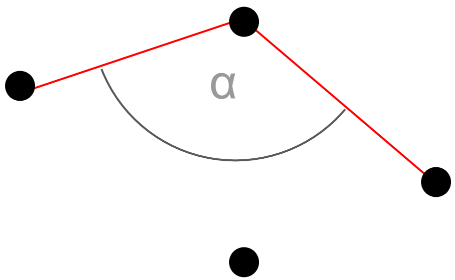
Extracts the Crust in a single step from the Delaunay graph



Also extracts the medial axis skeleton

NN-Crust [DK99]

Simple and elegant improvement of Crust:

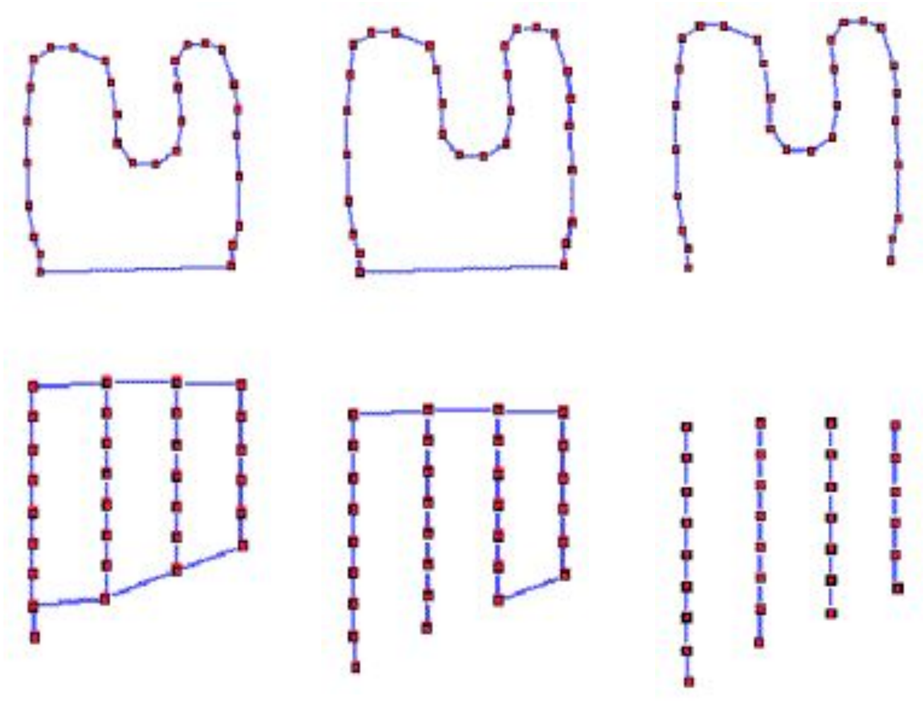


First, connects point to nearest neighbor

Then to nearest neighbor in half-space s.t. angle > 90°

Proof: $\varepsilon < 1/3$, corresponding to $\alpha > 141°$

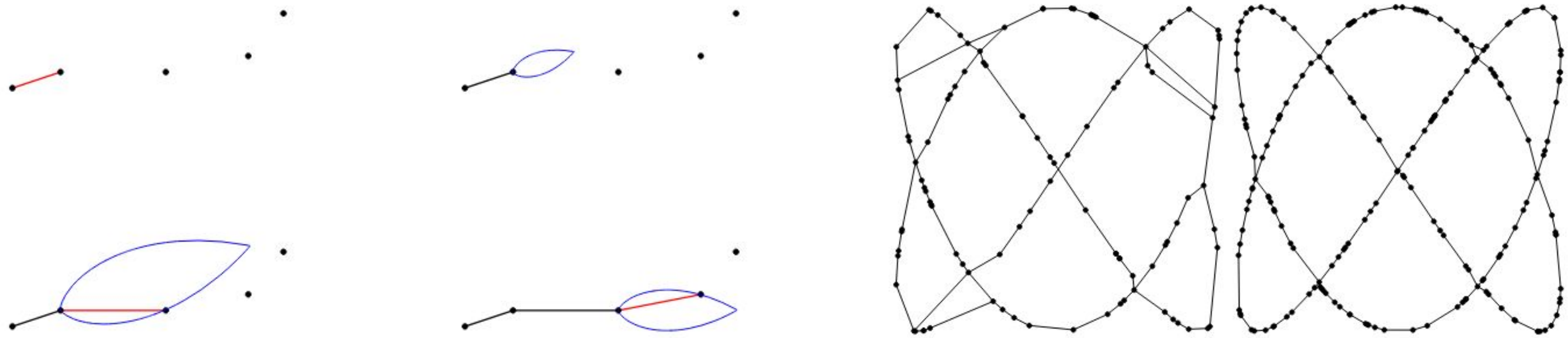
Conservative Crust [DMR99]



Crust NN-Crust Conservative Crust

- Filters edges from Gabriel graph
- Robust to outliers
- Collections of open/closed curves
- But requires a parameter
- Misses some sharp corners

Lenz: Probe Reconstruction [Len06]



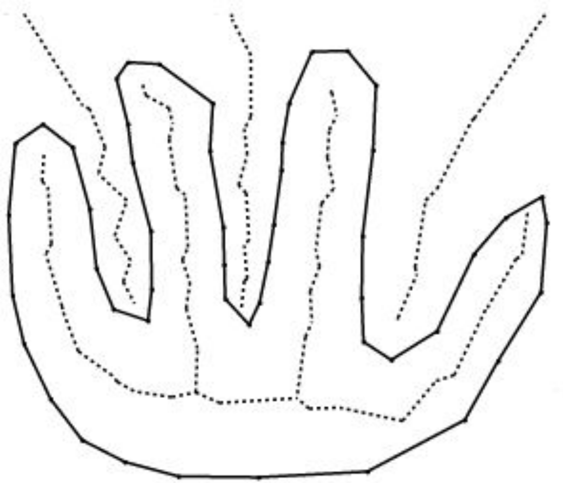
Starts with a seed edge and connects edges with a probe shape

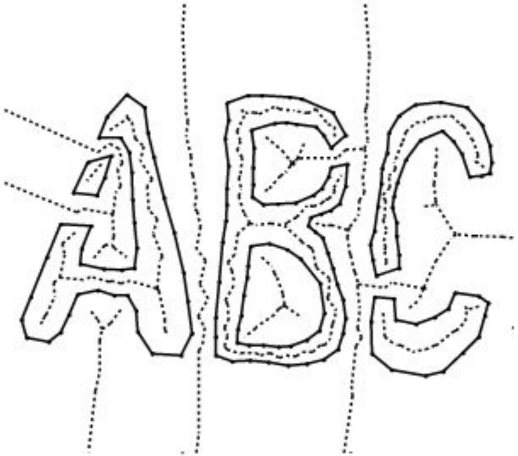
Requires an angle parameter

Permits self-intersections

Claims ε < 0.48 but no proof

Hiyoshi: TSP [Hiyo9]





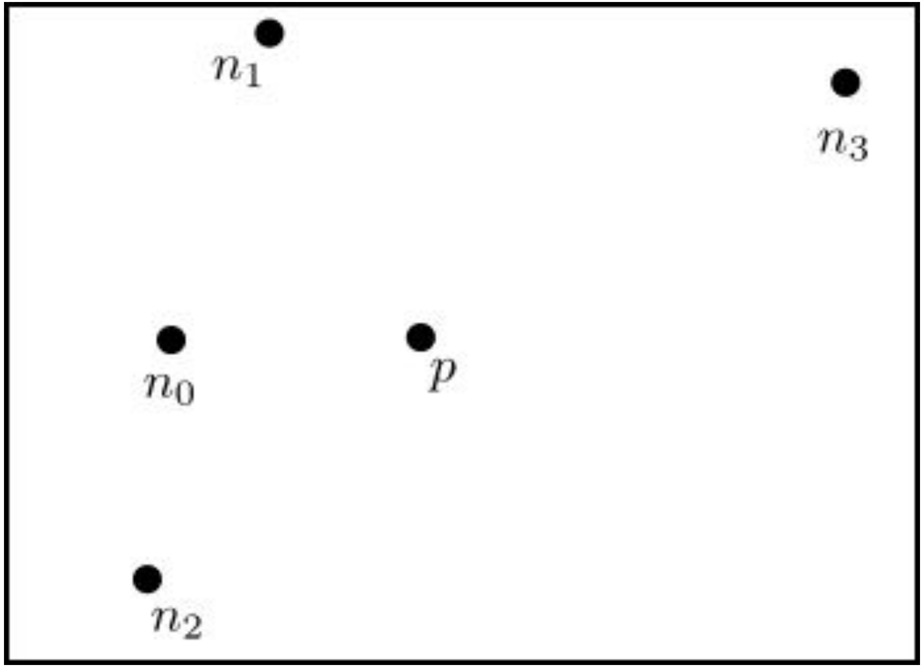
Adapts Traveling Salesman Problem to multiple connected curves

Transforms it into maximum-weight 2-factor problem (solvable in P time)

Proof for: $\varepsilon < ⅓$, $u < 1.46$ (relative uniformity of adjacent edge lengths)

HNN-Crust [OMW16]

Simple variant of NN-Crust, reducing angle from 90° to 60°:



First, connect nearest neighbor

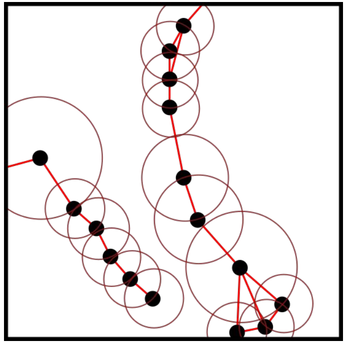
Construct half-space

Connect to nearest neighbor

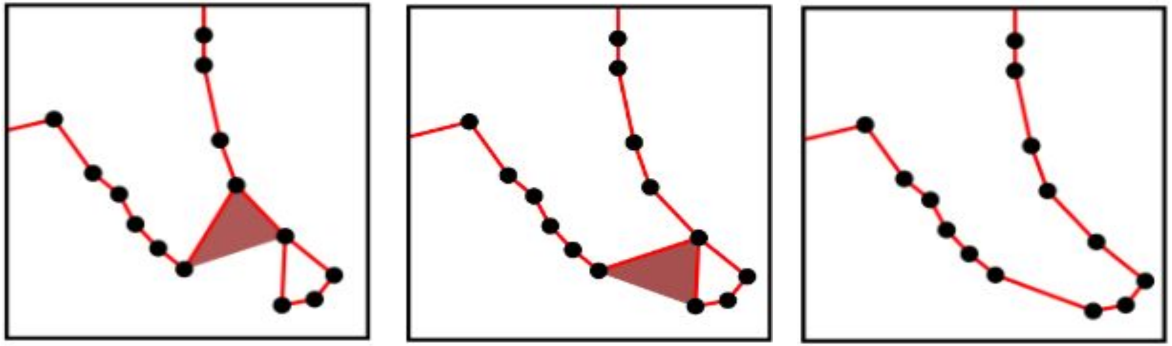
in opposite half-space

$\varepsilon < 0.47$

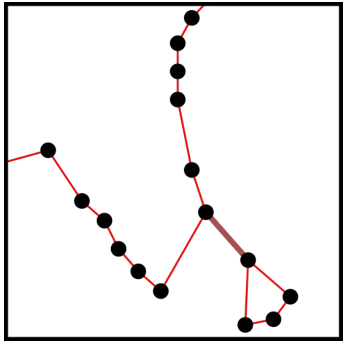
SIGDT [MOW22]



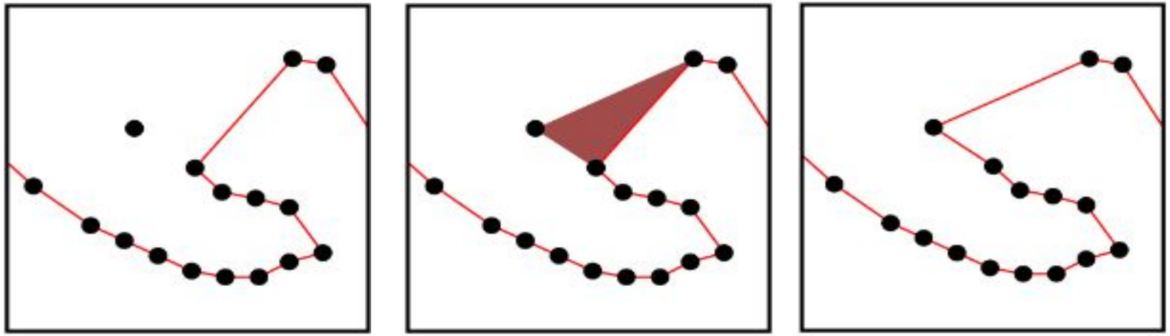
1) SIGDT=SIG ∩ DT



3) Inflating creates a manifold boundary

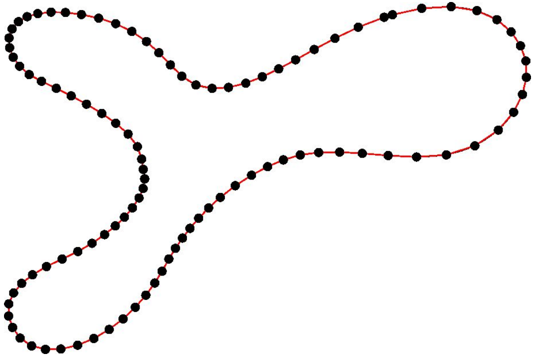


2) Enforce d ≥ 2



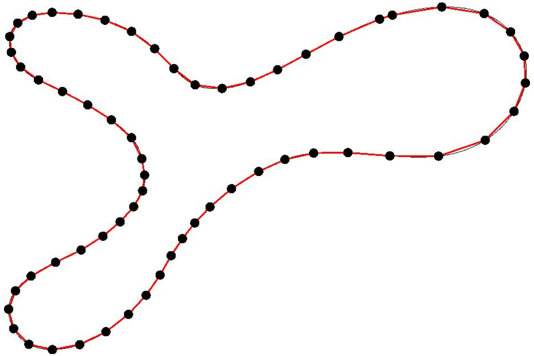
4) Sculpting interpolates interior vertices: ε<0.5, u<2

Sampling Conditions of Crust Algorithms



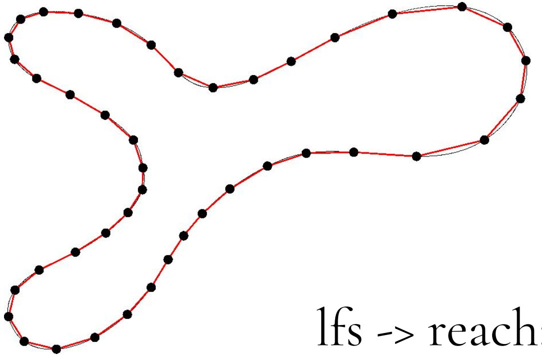
Crust:

$\varepsilon < 0.252$



NN-Crust:

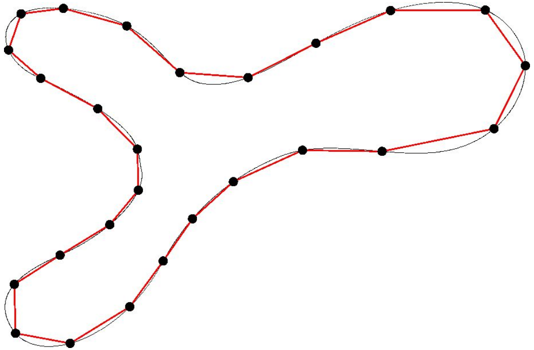
$\varepsilon < 1/3$



HNN-Crust:

$\varepsilon < 0.47$

lfs -> reach: $\rho = \varepsilon/(1-\varepsilon)$



HNN-Crust:

$\rho < 0.9$

Algorithms Based on Feature Size - Conclusion

Crust [ABE98], Anti-Crust [Gol99], NN-Crust [DK99], Conservative Crust [DMR99]

Lenz [Len06], Hiyoshi [Hiy09], HNN-Crust [OMW16], SIGDT [MOW22]

Guarantees on sampling condition

Work well for non-noisy point sets

Outline

Topic: Benchmark and Demo

To be precise:

- What all our benchmark has?
- How to use our benchmark?
- What all we evaluated?
- What are our conclusions?

Presenter:

Amal Dev PARAKKAT

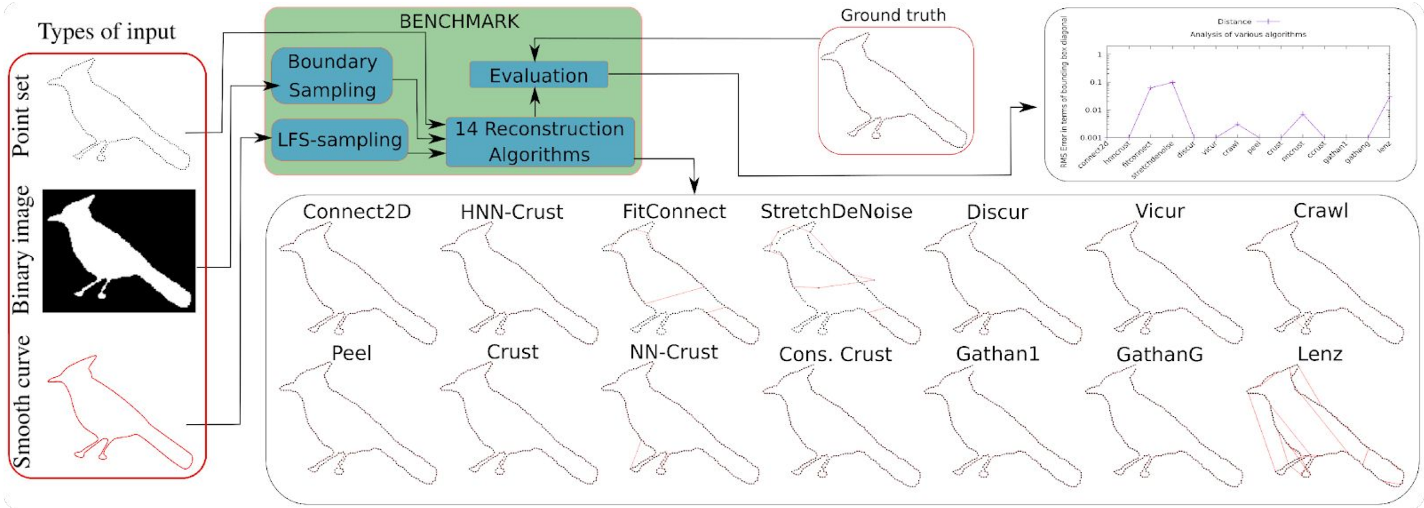
Assistant Professor

Institut Polytechnique de Paris

amal.parakkat@telecom-paris.fr

The Benchmark

Our benchmark contains: Algorithms, Dataset, Sampling tools, Evaluation criterias, and Test scripts



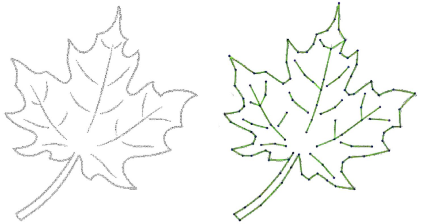
The Benchmark - Algorithms

We included 15 publicly available algorithms

Contains algorithms from late 90s (Crust family) to 2018

- Crust, NNCrust, CCrust, Gathan, GathanG, Lenz, Discur, Vicur, OptimalTransport, Connect2D, Crawl, HNNCrust, FitConnect, StretchDenoise, Peel

We removed OptimalTransport from experiments since it simplifies curves



The Benchmark - Dataset

Our dataset contains more than 2500 point sets:

- Classic data - Collected from various papers (using WebPlotDigitizer)
- Image data - Samples obtained from the silhouette images (taken from MPEG7 CE Shape-1, Edinburgh, 1070-shape image databases)
- Synthetic data - Analytical (shapes with sharp corners, & self-intersections) and ε-sampled points

Classic Data

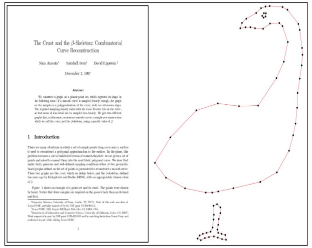
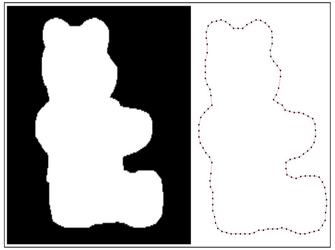
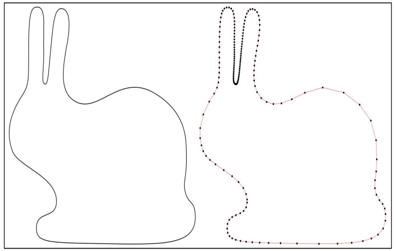


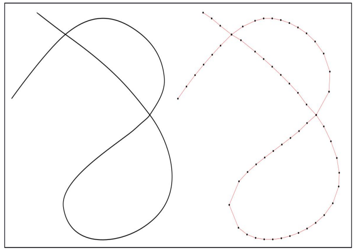
Image Data



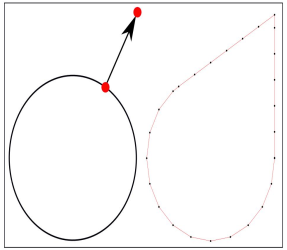
LFS sampling



Non-manifold



Sharp corners

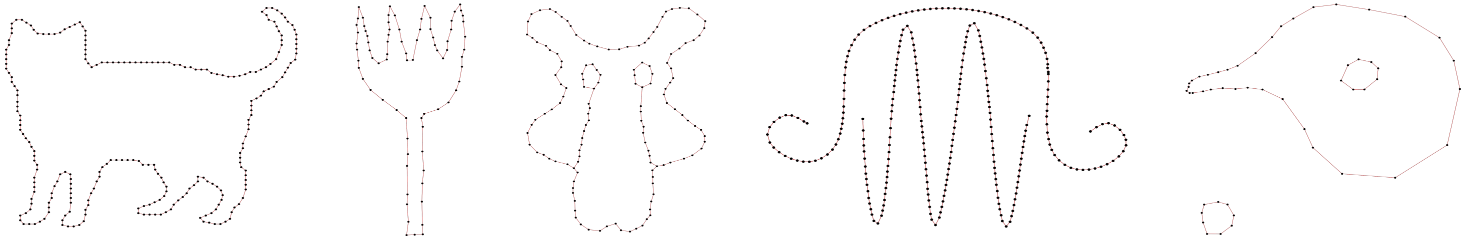


The Benchmark - Dataset

We also provide ground truths (linear approximation) as:

- Ordered vertices: A loop of vertices - for simple closed curves
- Edge list: List of edges - for complex curves

Grouped under the following categories:



Manifold   Sharp   Non-manifold   Open curves   Multiple curves

Moreover, we provide an interactive ground truth generation tool

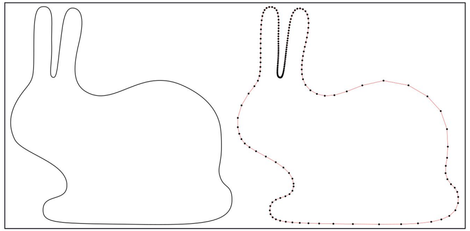
The Benchmark - Sampling tools

LFS-sampling tool:

- Samples are made from input Bezier curve representation
- Maximal empty disks are computed to create a medial axis approximation
- Estimate LFS at each sample and use it to pick a set of samples satisfying the ε-sampling condition

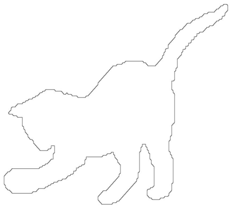
Contour sampling tool:

- Binary image contour is extracted to generate a set of samples
- Starting from a random sample, iteratively remove all samples within a user defined distance r

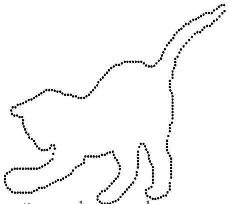


A Bezier curve and its LFS-based sampling

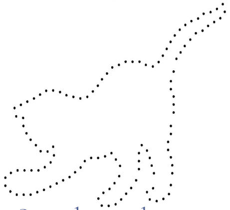
A binary image



Extracted contour



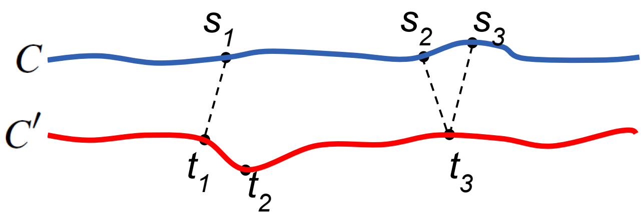
Samples with r=20



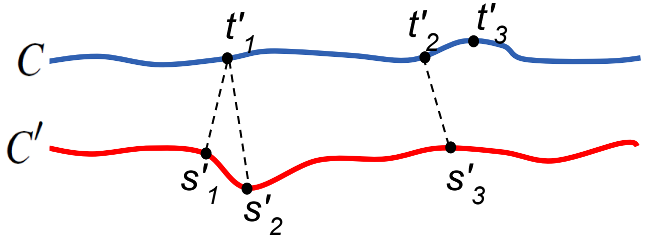
Samples with r=50

The Benchmark - Evaluation criteria

Let closest point correspondences be D and D' of two curves C and C'



$$D = (s,t)|s \in C', t = M(s)$$



$$D' = (s',t')|s' \in C, t' = M'(s')$$

where M and M' be the respective non-bijective shortest distance maps

We use the following metrics to compare two curves:

$$H_D(C,C') = \max\left\{ \max_{(s,t)\in D} \|s-t\|, \max_{(s',t')\in D'} \|s'-t'\| \right\}$$

Hausdorff distance

$$RMS_D(C,C') = \sqrt{\frac{1}{N}\left(\sum_{(s,t)\in D} \|s-t\|^2 + \sum_{(s',t')\in D'} \|s'-t'\|^2 \right)}$$

Root mean squared distance $N = |D| + |D'|$

The Benchmark - Test scripts

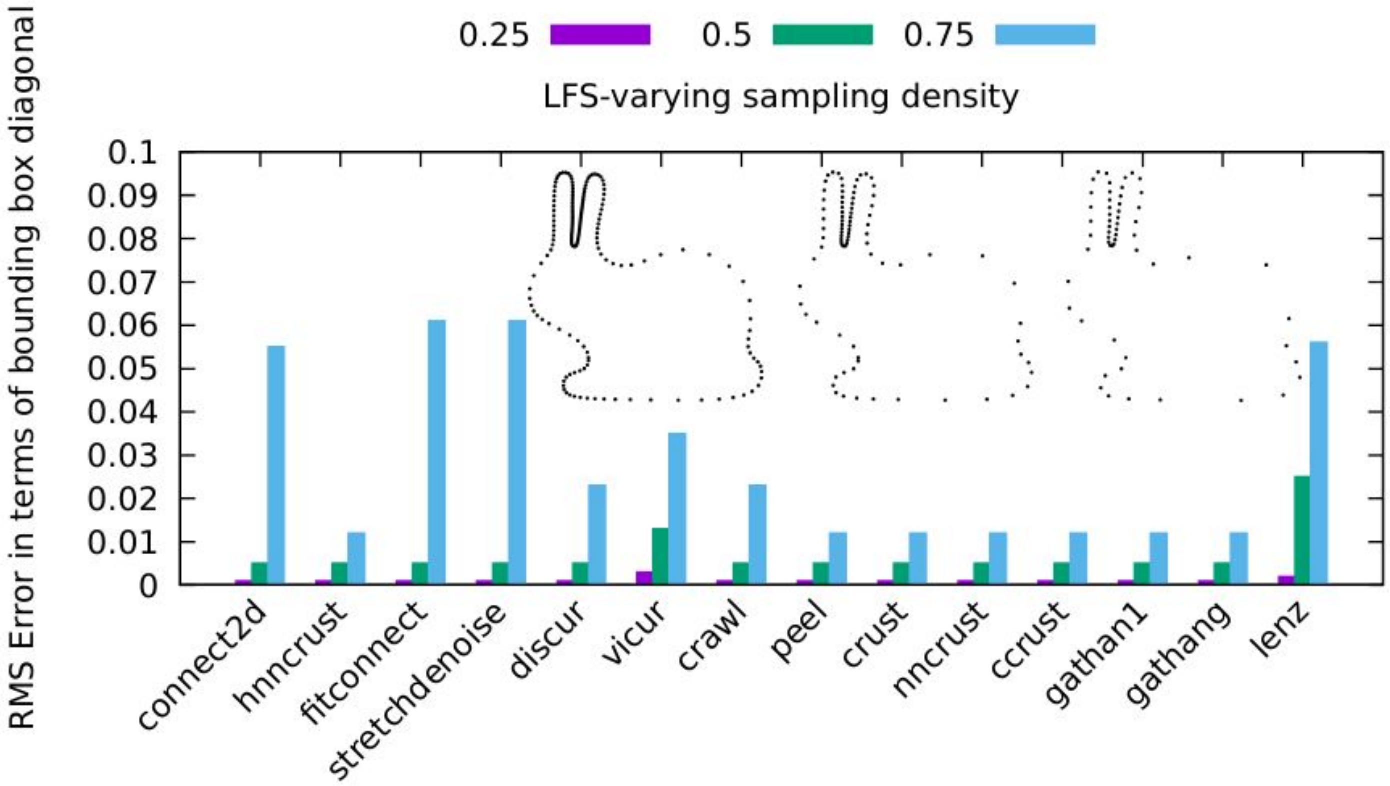
Driver program can be run with various arguments and options

A set of test scripts for quantitatively & qualitatively evaluate the algorithms

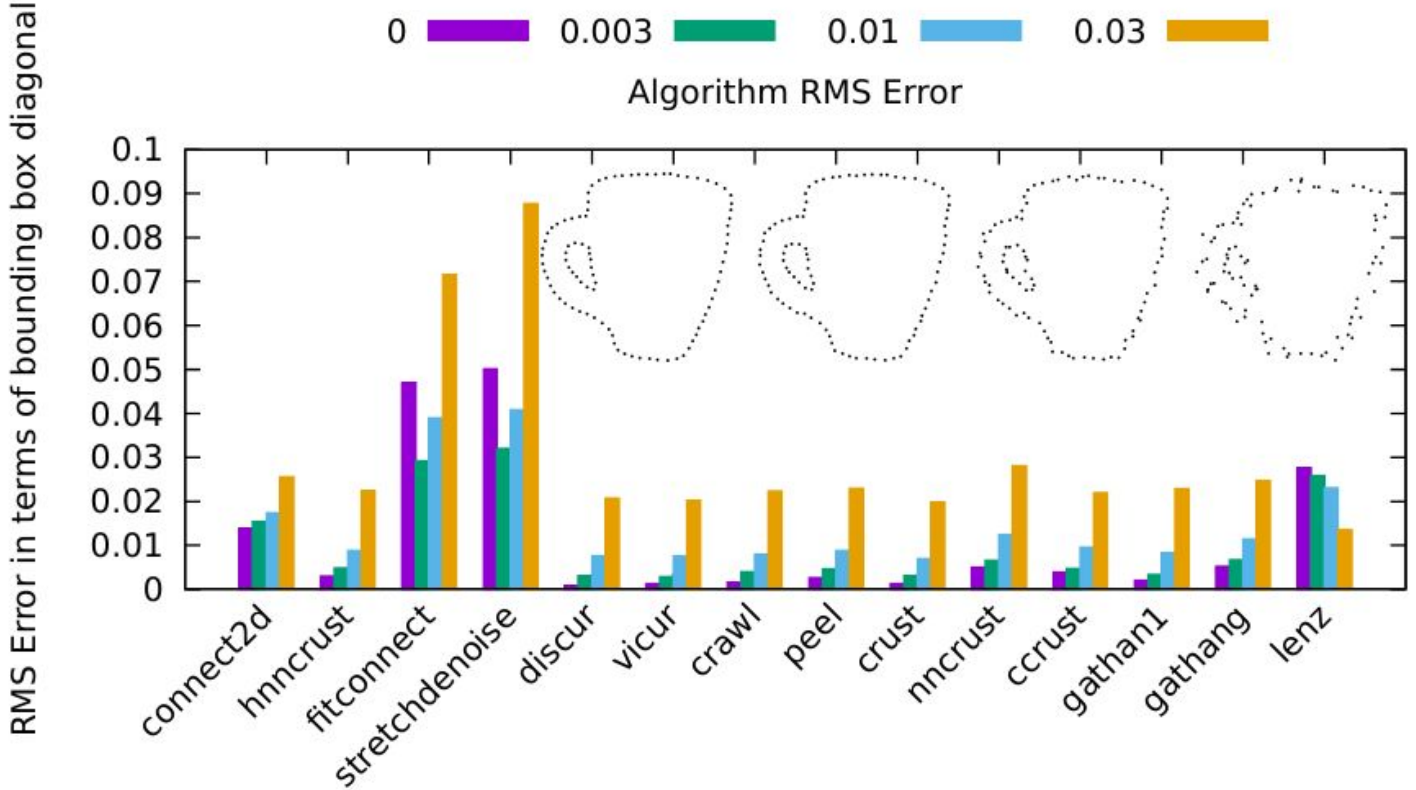
Each test script has a list of algorithms and test data, designed for the specific experiment:

- **run-sampling.sh:** ε-sampled [ABE98] test data
- **run-noisy.sh:** perturbed with uniform noise
- **run-lfsnoise.sh:** perturbed with lfs-based noise
- **run-outliers.sh:** added outlier points
- **run-manifold.sh:** whether reconstruction is a manifold
- **run-sharp-corners.sh:** sharp feature curves
- **run-open-curves.sh:** open curves
- **run-multiple-curves.sh:** multiply connected curves
- **run-intersecting.sh:** curves with intersections

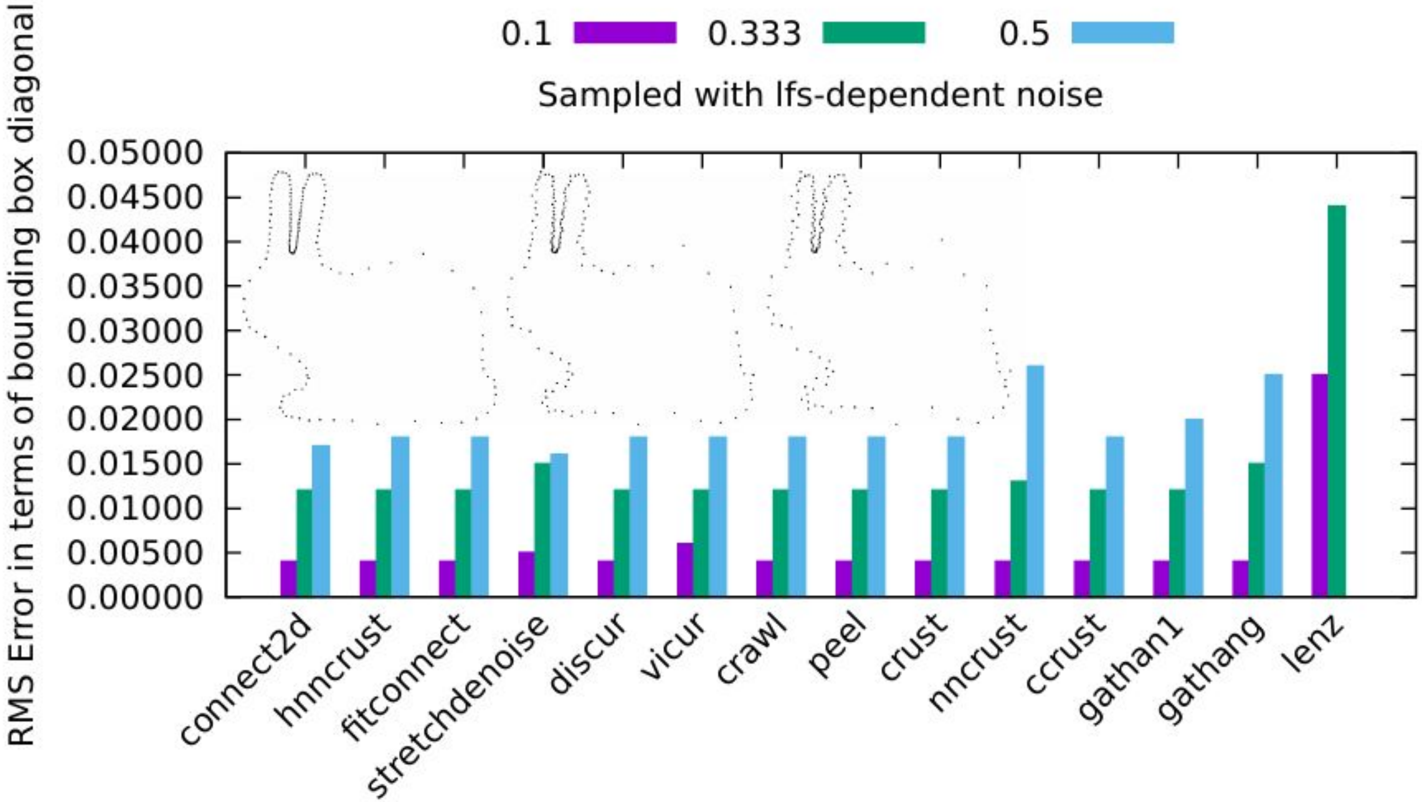
Evaluation - Sampling density as ε-sampling



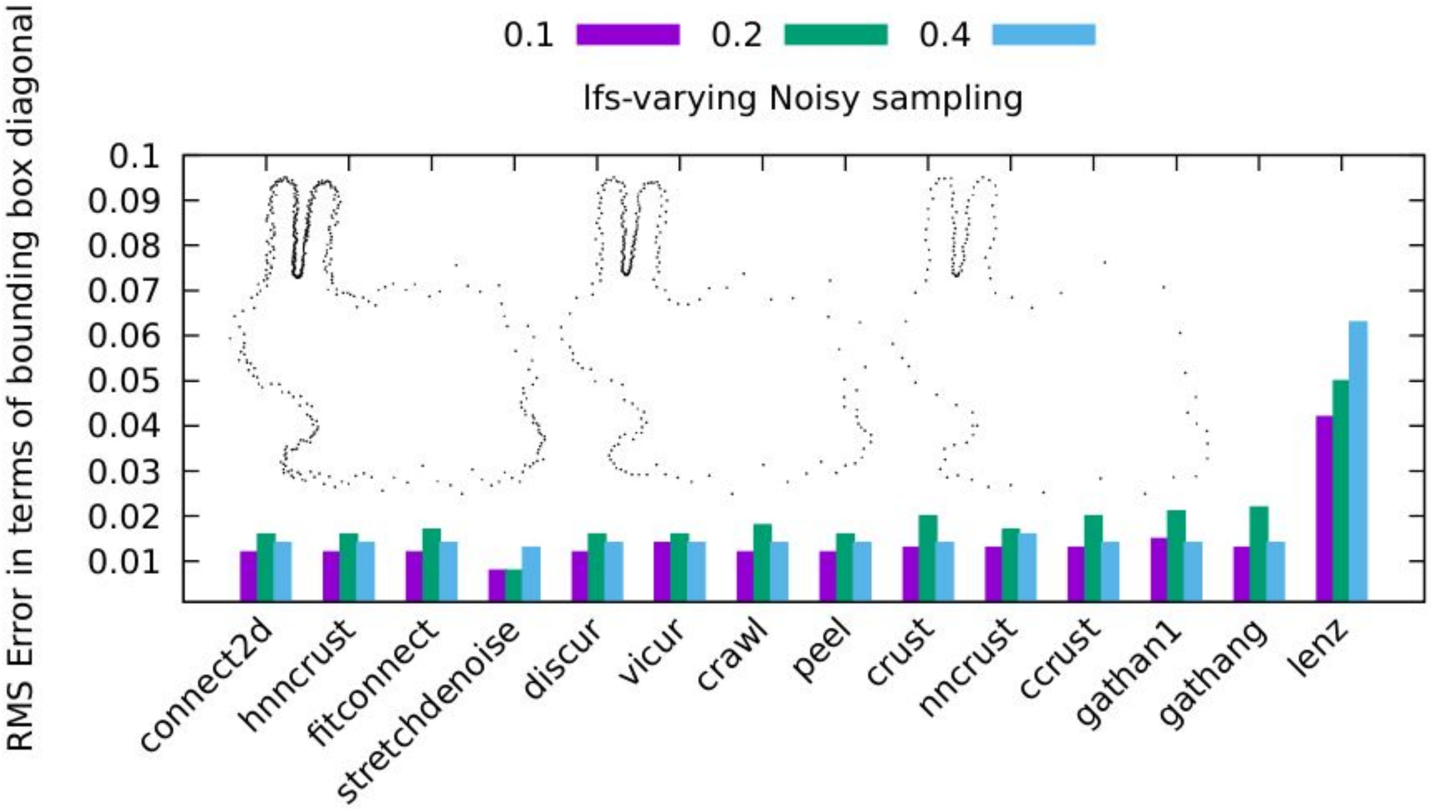
Evaluation - Noise robustness (BB Diagonal)



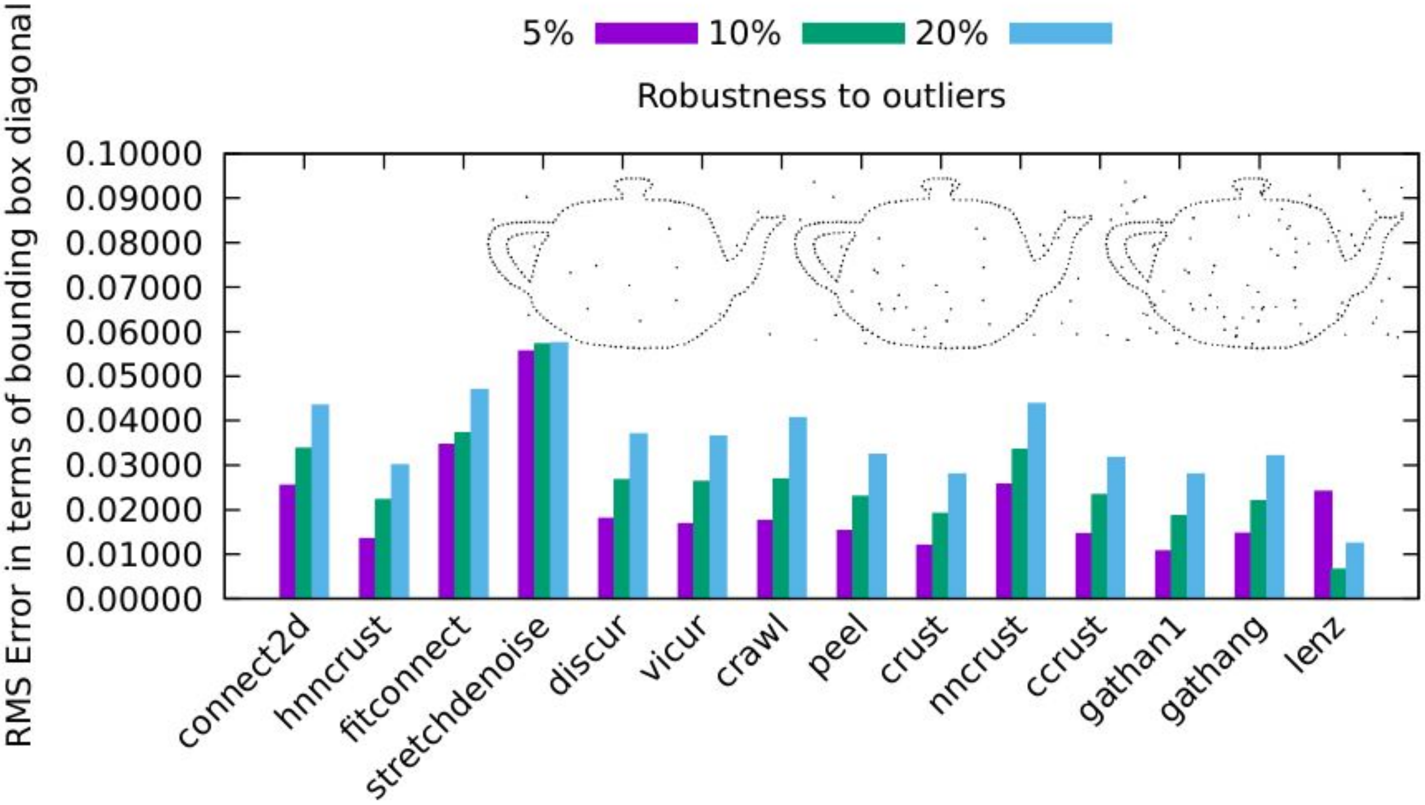
Evaluation - Noise robustness (LFS)



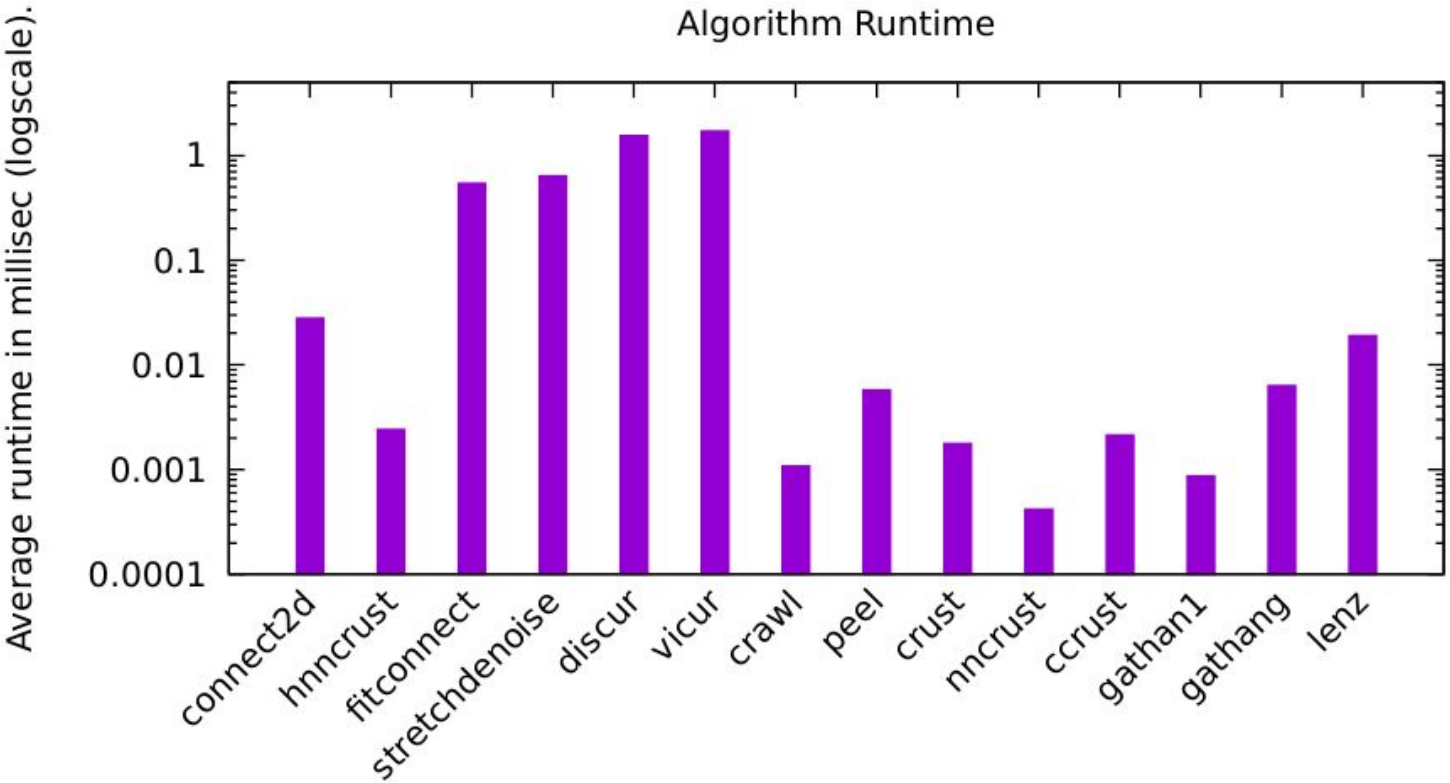
Evaluation - Noise robustness (ε-sampling + LFS)



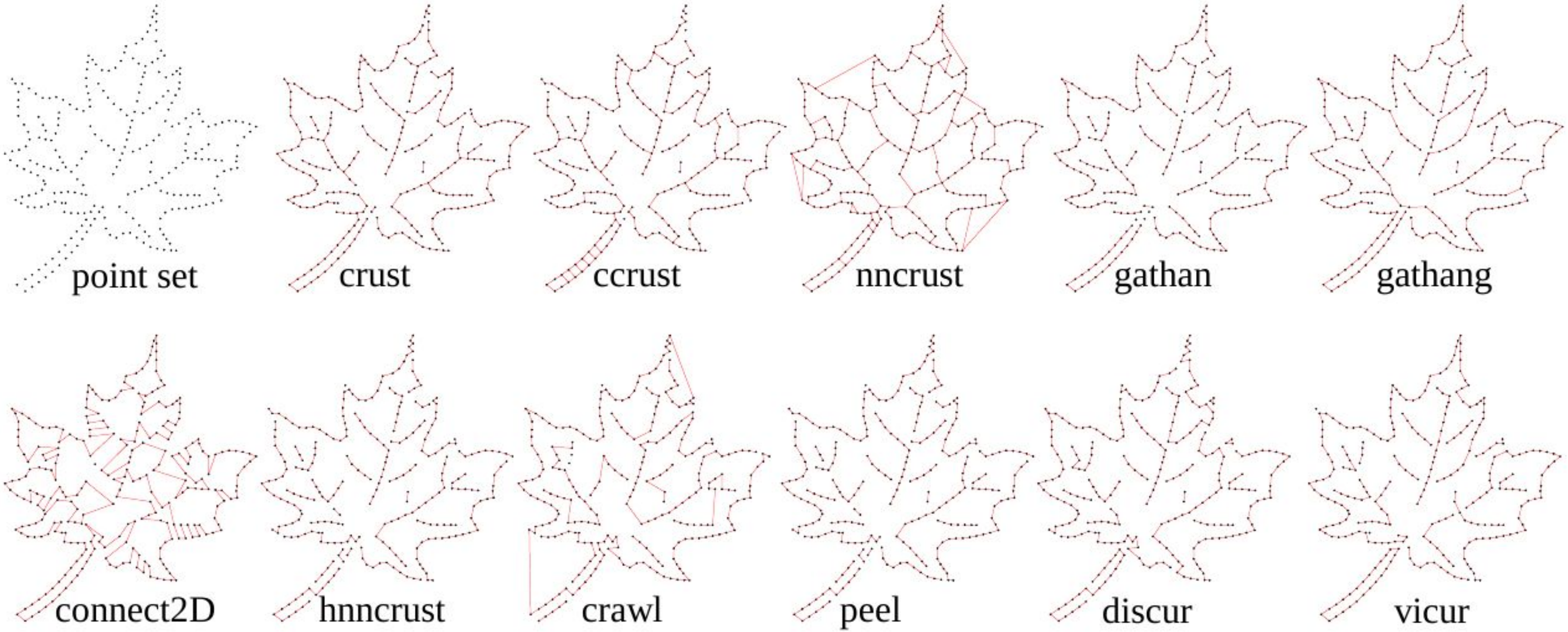
Evaluation - Outlier



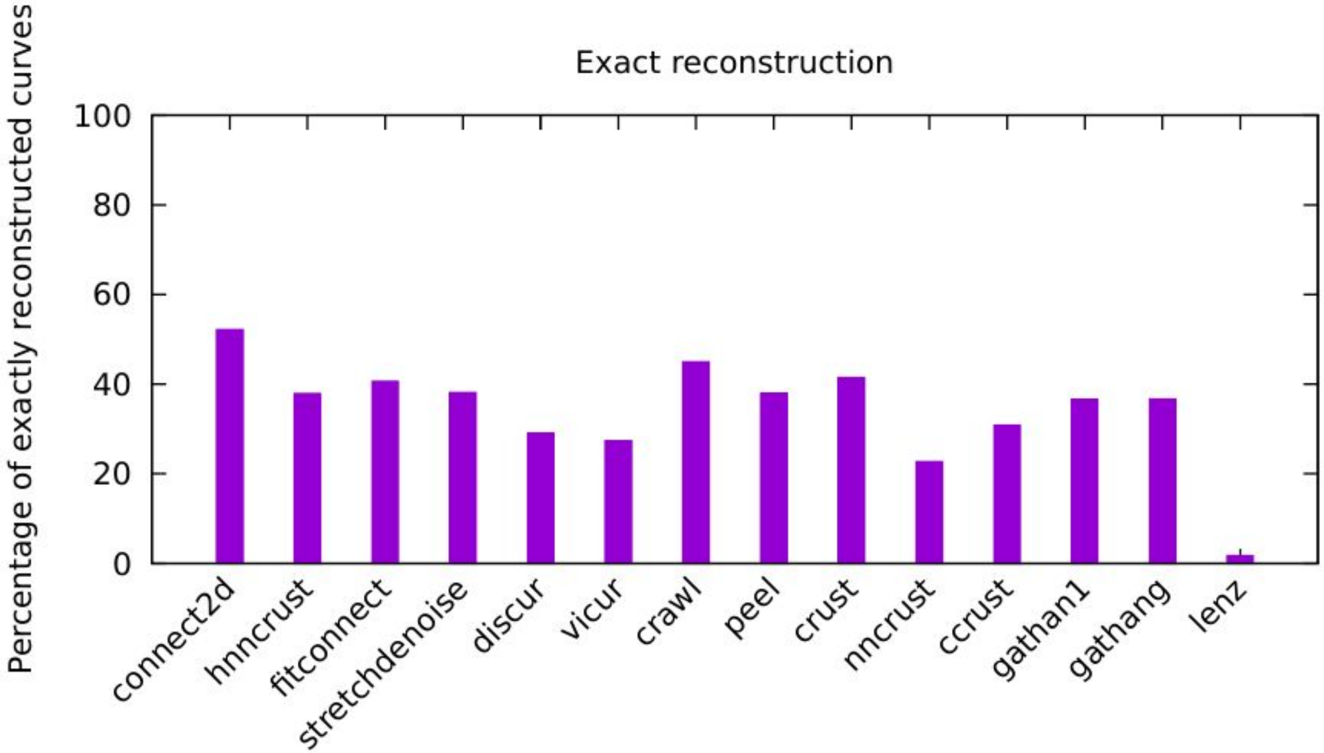
Evaluation - Running time



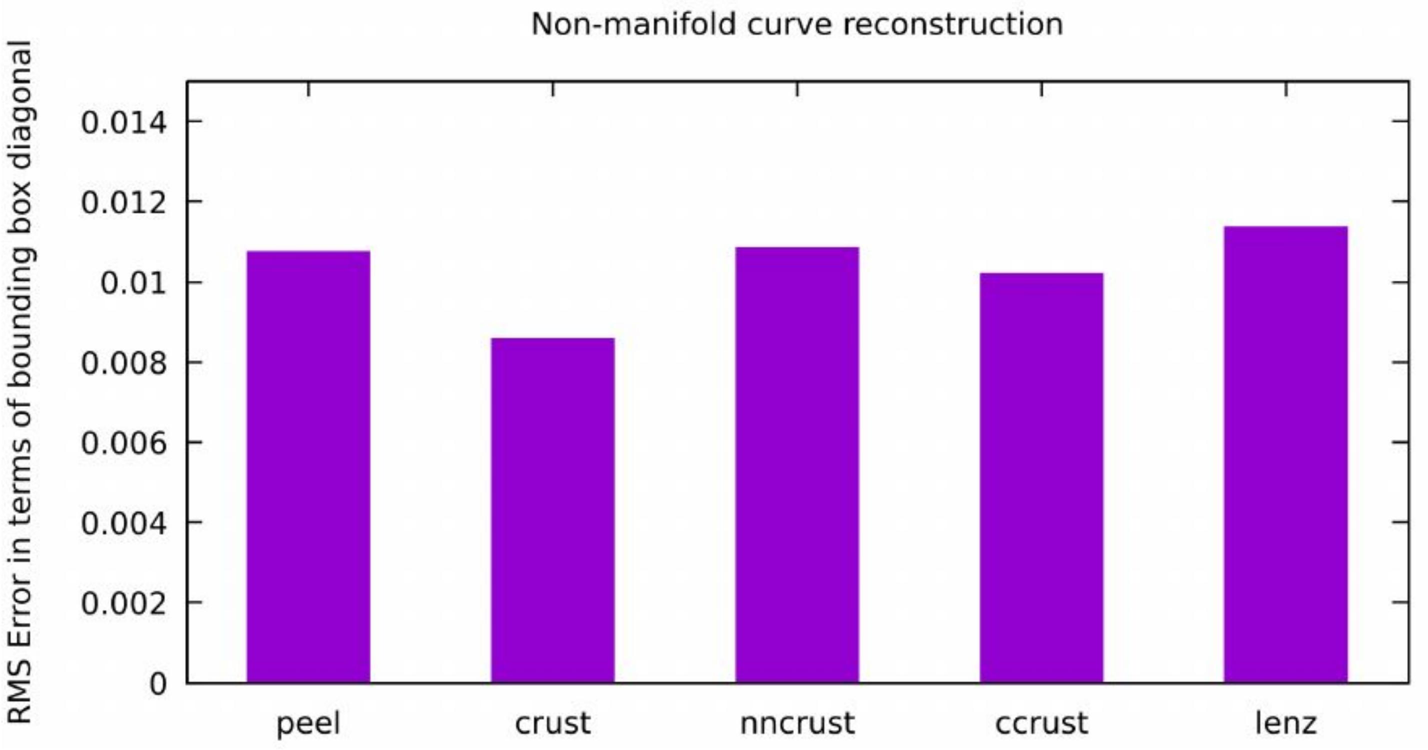
Evaluation - Qualitative Comparison



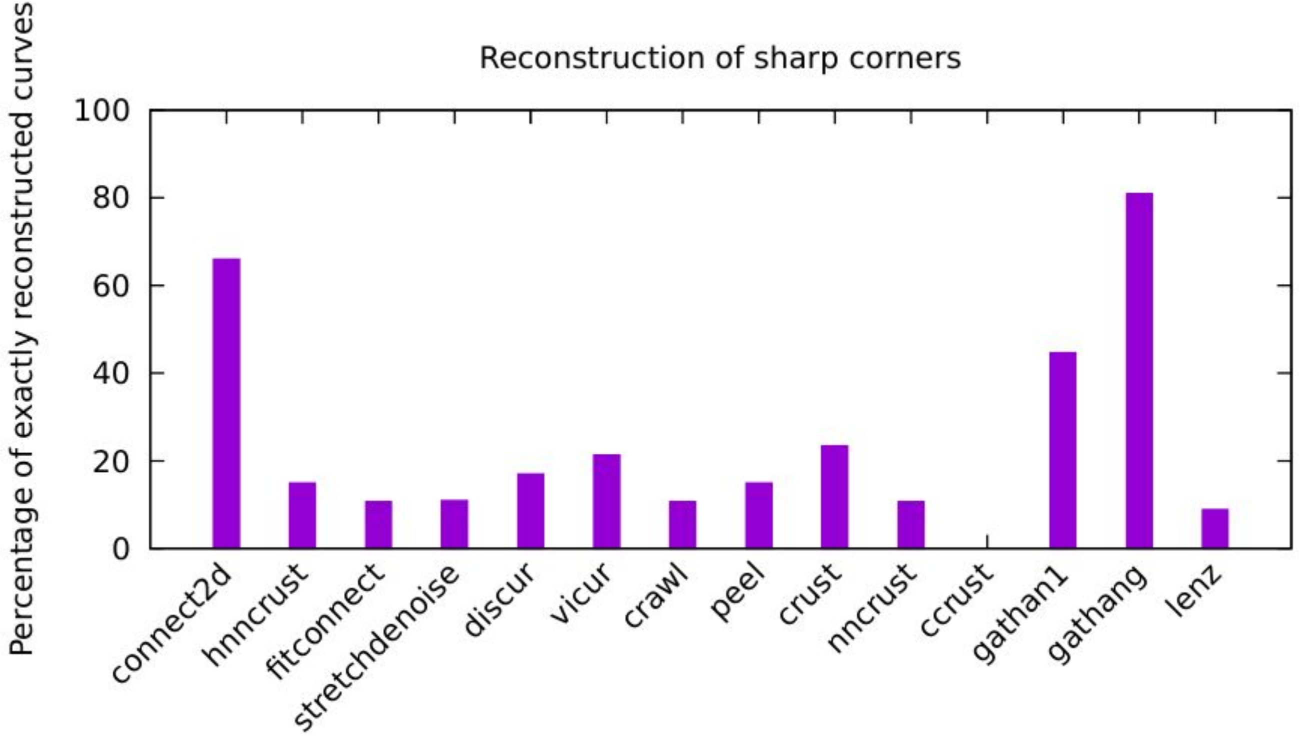
Evaluation - Qualitative Comparison



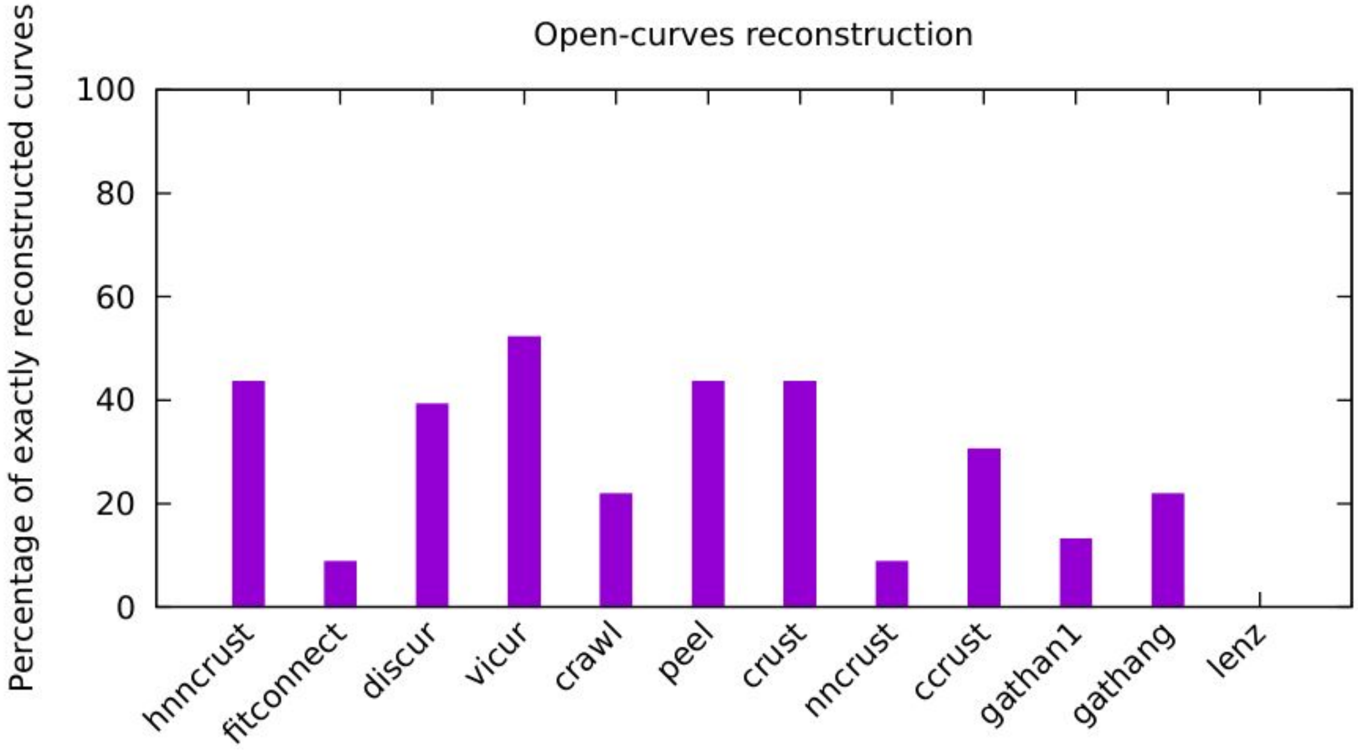
Evaluation - Qualitative Comparison



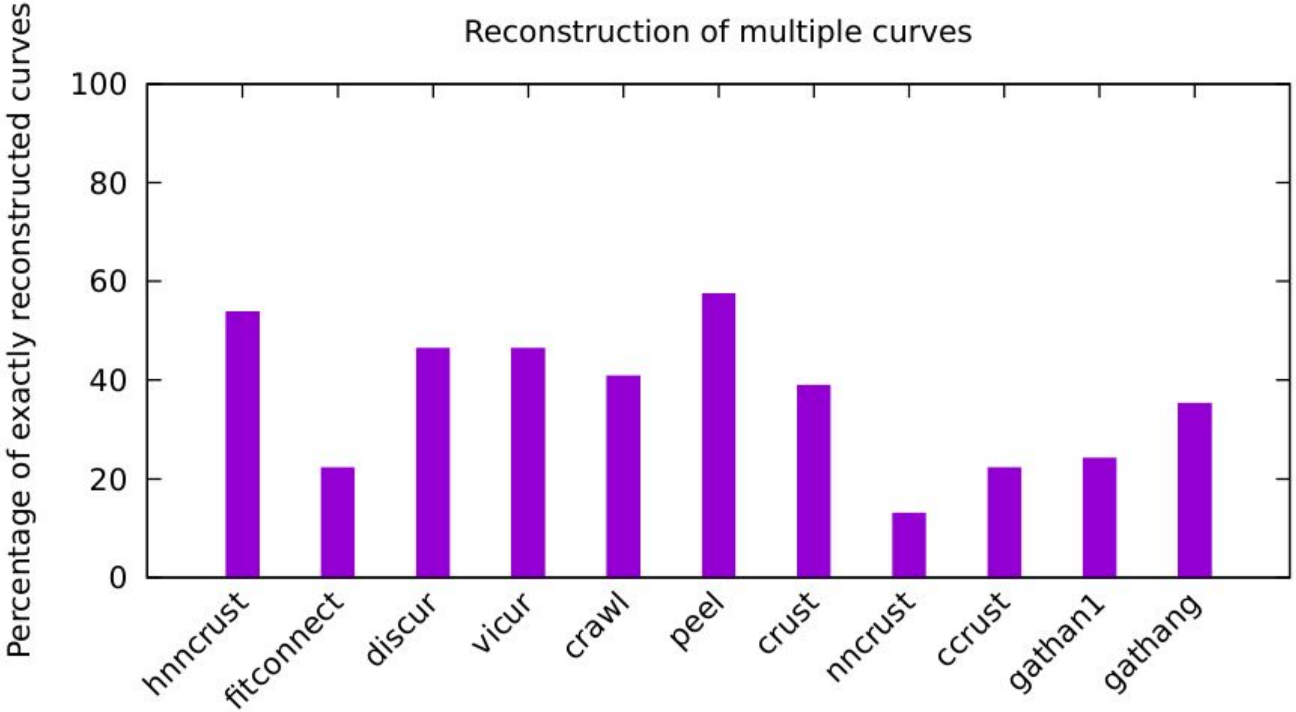
Evaluation - Qualitative Comparison



Evaluation - Qualitative Comparison



Evaluation - Qualitative Comparison



Evaluation - Overview

In short, we evaluate the robustness of various algorithms based on:

- Sampling density as ε-sampling
- Noise robustness as δ of bounding box diagonal
- Noise robustness as δ of lfs
- Noise+sampling density as ε-sampling and δ of lfs
- Outliers robustness in % of samples
- Average runtimes (in s)

Evaluation - Summary

Curve/Input feature	Best two algorithms in order
Uniform Noise	DISCUR, VICUR
Non-uniform Noise	STRETCHDENOISE, CONNECT2D
Outliers	HNN-CRUST, CRUST
Non-uniform sampling	HNN-CRUST, PEEL
Runtime	NN-CRUST, GATHAN1
Manifold curves	CONNECT2D, CRAWL
Non-manifold curves	CRUST, LENZ
Sharp features	GATHANG, CONNECT2D
Open curves	VICUR, HNN-CRUST
Multiple curves	PEEL, HNN-CRUST

Outline

Topic: Sketch Reconstruction

To be precise:

- Sketching
- Sketching and reconstruction
- Sketch completion
- Rough sketch simplification

Presenter:

Amal Dev PARAKKAT

Assistant Professor

Institut Polytechnique de Paris

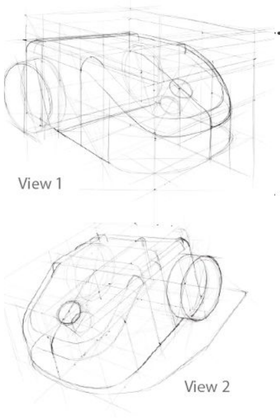
amal.parakkat@telecom-paris.fr

Sketching

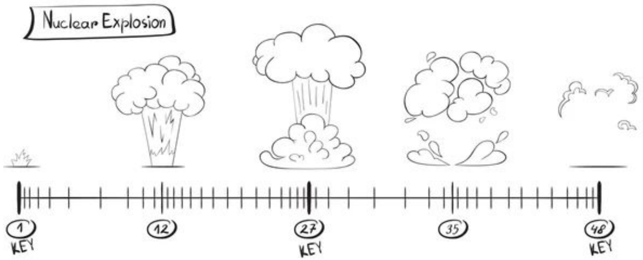
Sketching is an integral part of everyone's life
Sketching comes across us in various stages of our life:

- Kids playing with pencils and paints
- Intrinsic part of various academic curriculums
- Professional usage

MOREOVER, IT'S FUN!!!!!

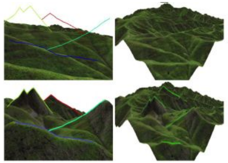


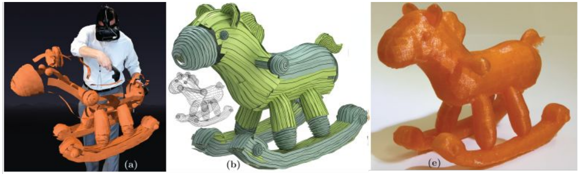
b. Concept sketch

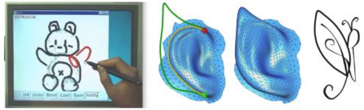


Relevance of sketch processing in Computer Graphics

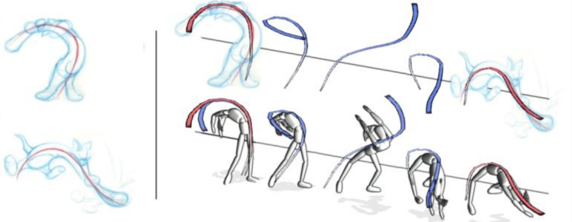

Wire art creation


Mountain creation

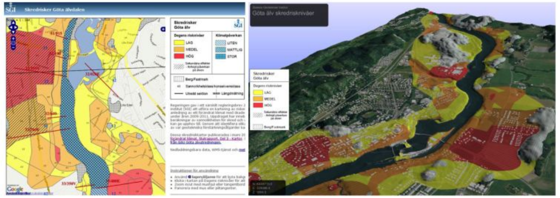

Printable models from VR drawings

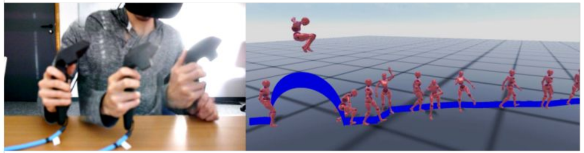

3d modeling & editing


Jewellery crafting


Expressive sketch-based animation


Animating sketches


Story telling


Sketching animation in VR

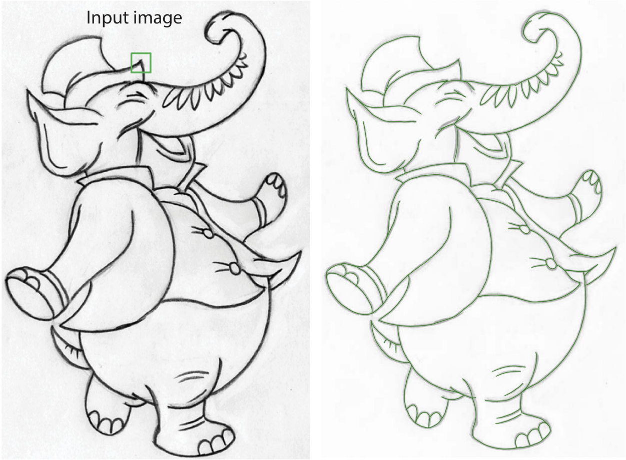

Sketch-based developable surfaces

Sketching and Reconstruction

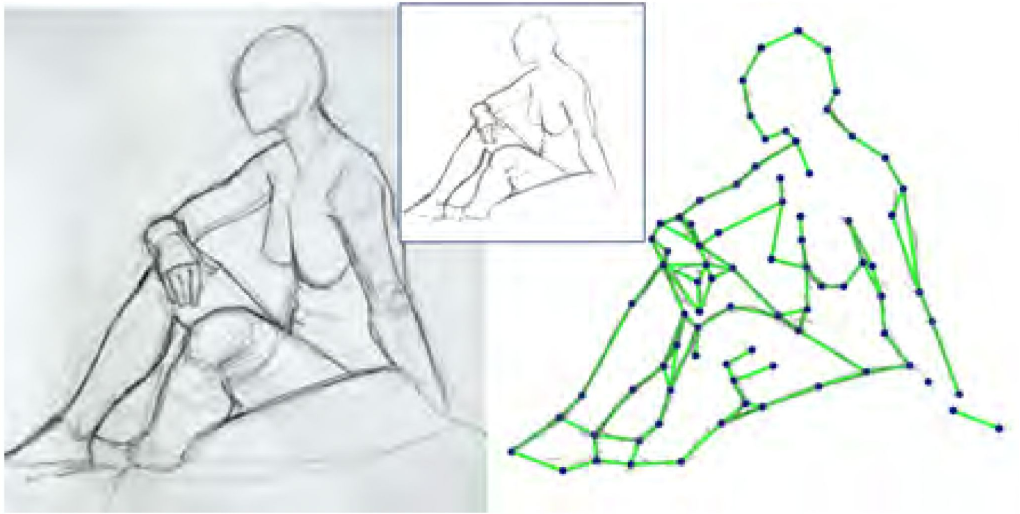
Sketching and reconstruction problem are closely related

Reconstructing from:

- Simple scanned sketch



© Bessmeltsev et al.



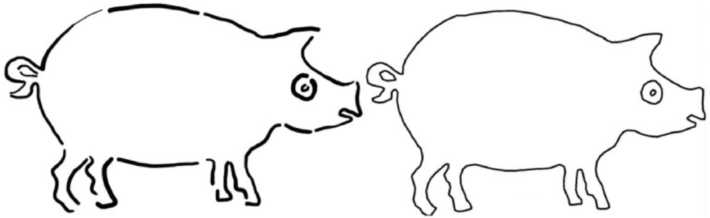
© de Goes et al.

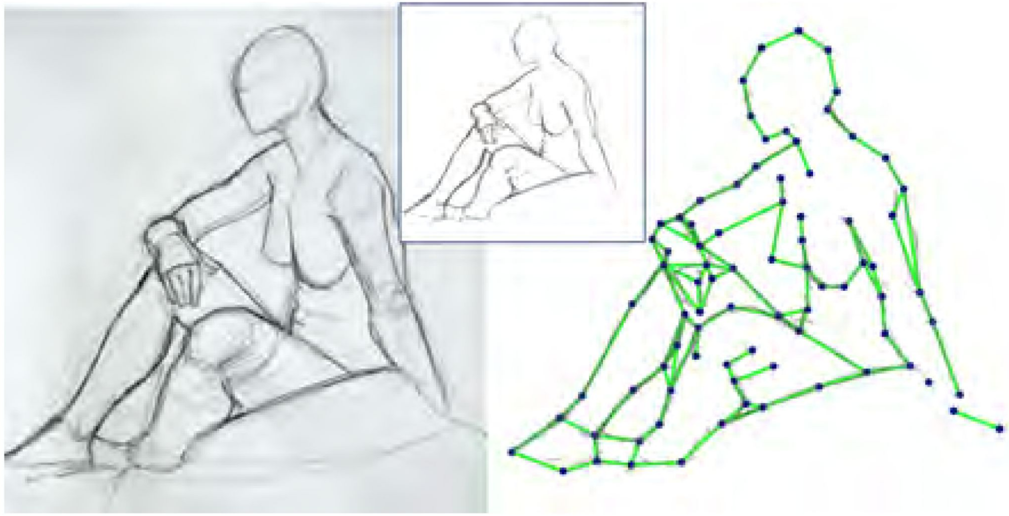
Sketching and Reconstruction

Sketching and reconstruction problem are closely related

Reconstructing from:

- Simple scanned sketch
- Missing strokes





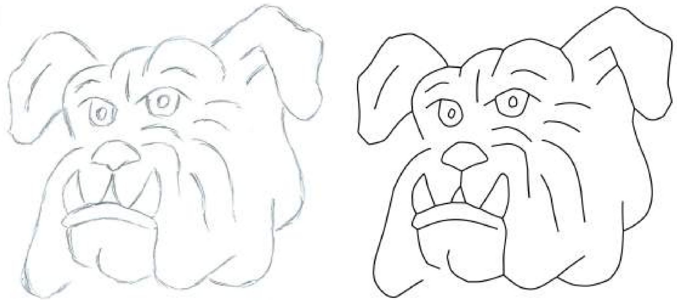
© de Goes et al.

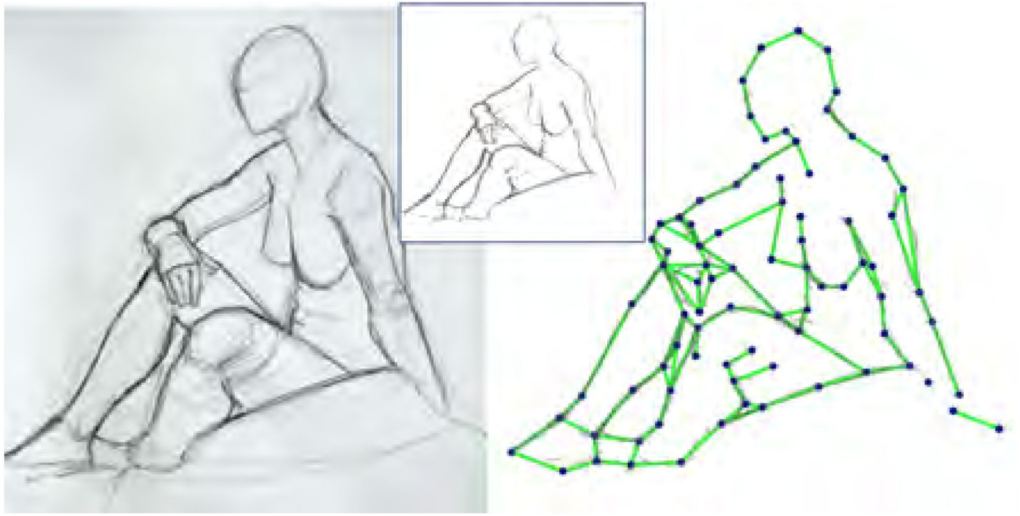
Sketching and Reconstruction

Sketching and reconstruction problem are closely related

Reconstructing from:

- Simple scanned sketch
- Missing strokes
- Noisy sketch





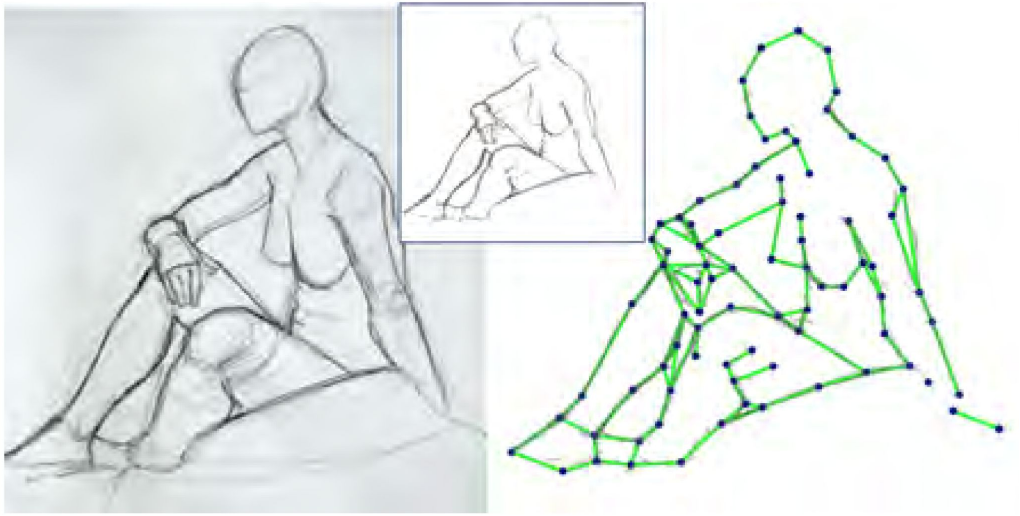
© de Goes et al.

Sketching and Reconstruction

Sketching and reconstruction problem are closely related

Reconstructing from:

- Simple scanned sketch
- Missing strokes
- Noisy sketch
- Dirty sketches



@ de Goes et al.

Sketch completion and Sketch simplification

We look at two important subproblems:

Sketch completion and sketch simplification

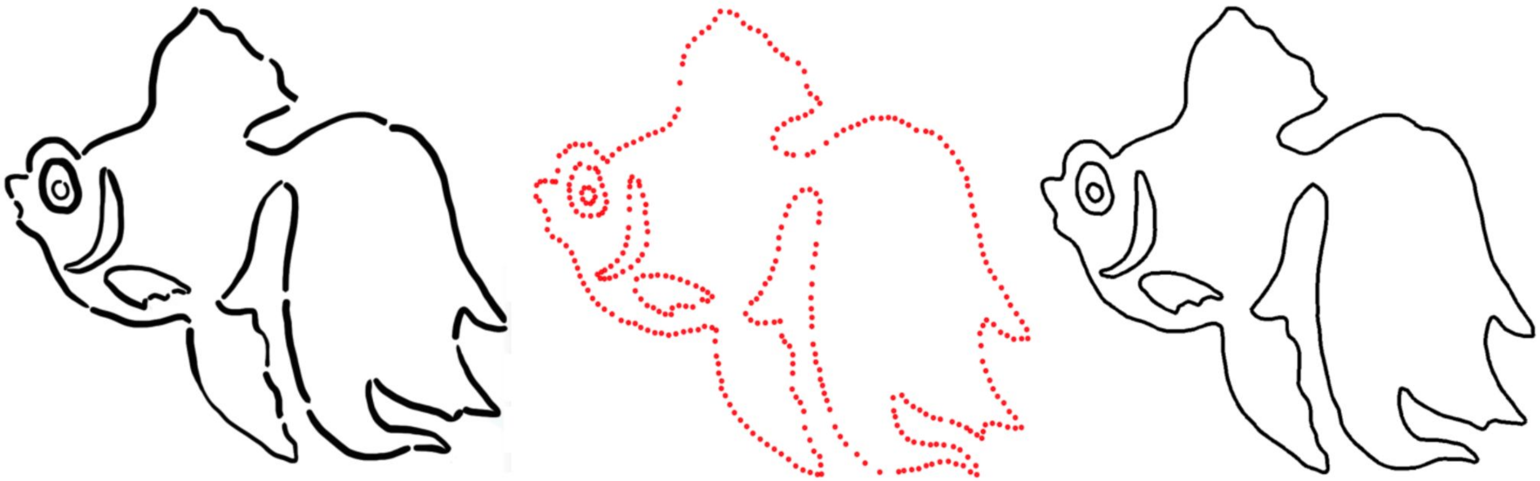
We won't have a detailed discussion, but a very quick and brief overview of:

- A.D. Parakkat, P. Memari, M.P. Cani, "Delaunay Painting: Perceptual Image Colouring from Raster Contours with Gaps" - Computer Graphics Forum 2022
- A.D. Parakkat, P. Madipally, H.H. Gowtham, M.P. Cani, "Interactive Flat Coloring of Minimalist Neat Sketches" - Eurographics 2020 (Short paper)

- A.D. Parakkat, M.P. Cani, K. Singh, "Color by Numbers: Interactive Structuring and Vectorization of Sketch Imagery" - ACM CHI 2021
- A.D. Parakkat, U.B. Pundarikaksha, R. Muthuganapathy, "A Delaunay triangulation based approach for cleaning rough sketches" - Computers & Graphics 2018

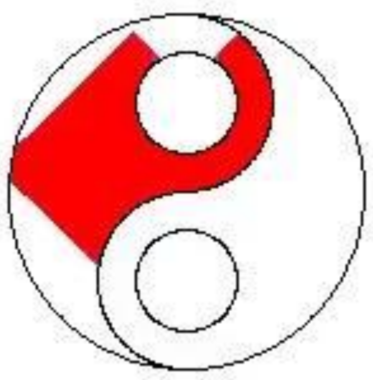
Sketch completion

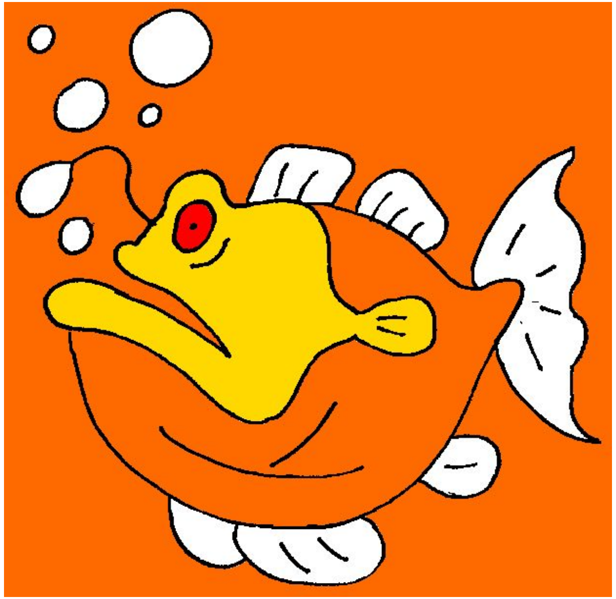
The input is a set of disconnected sketch strokes and is asked to appropriately connect them (in other words, filling the gaps in a line art)



Sketch completion - Sketch coloring

Using Flood-fill algorithm

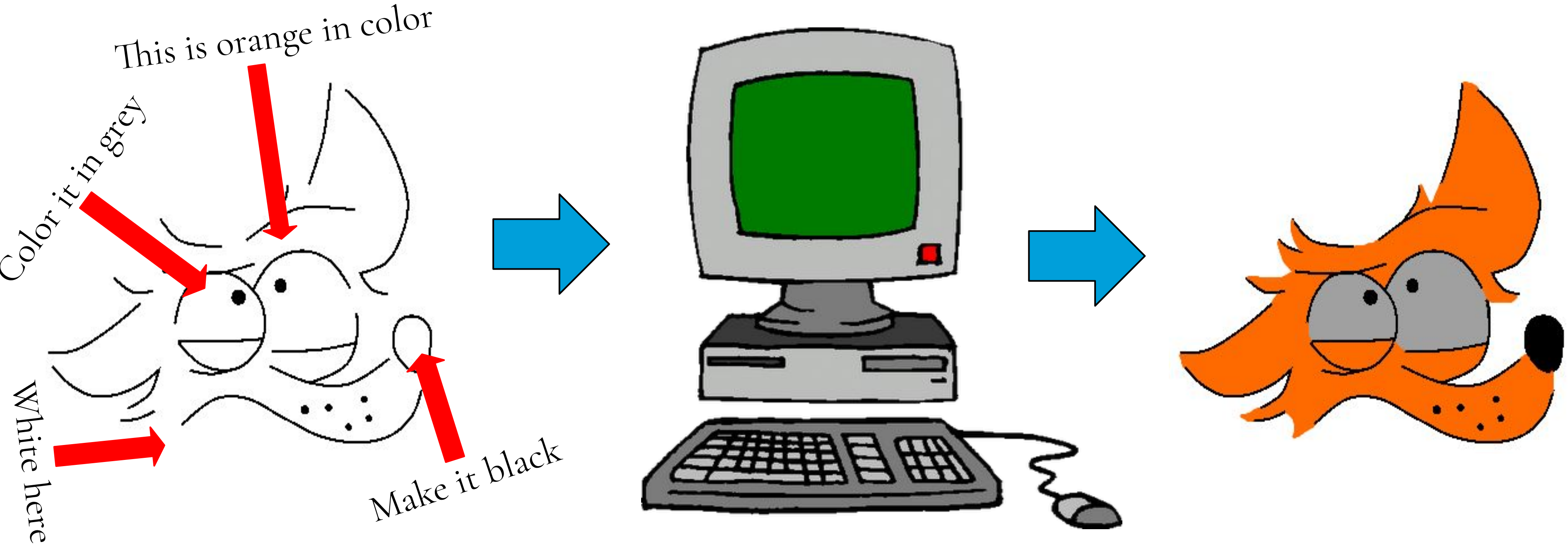




Distinct artistic feeling:

- Le Grand Méchant Renard et autres contes...
- Ernest & Celestine

Sketch completion - Sketch Coloring



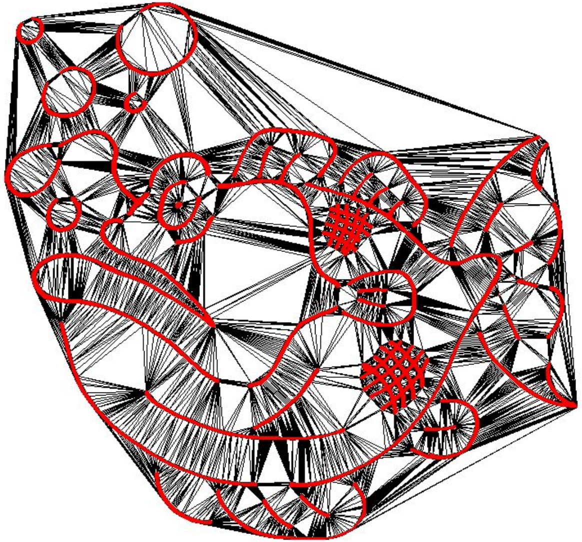
COLOR HINTS

DESIRED COLORING

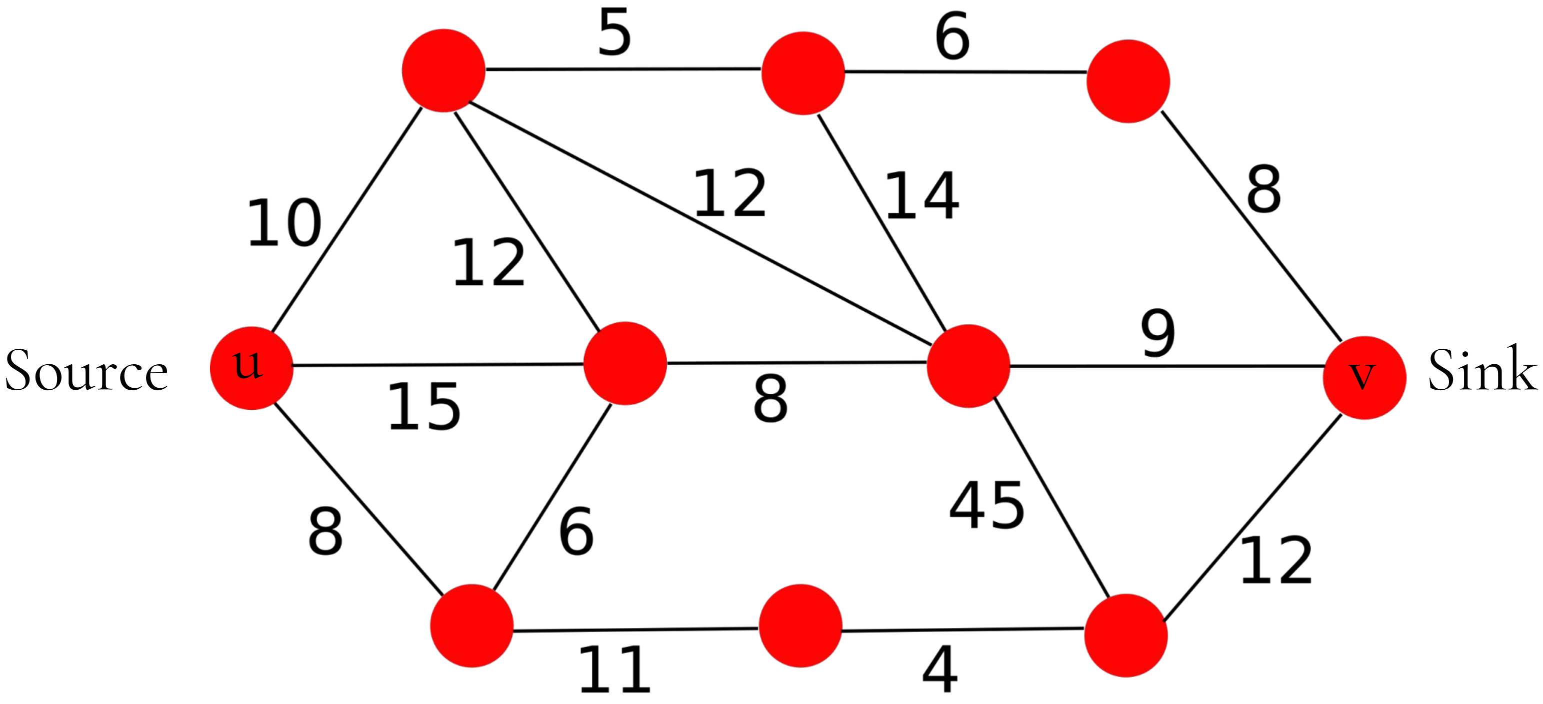
Infer the unknown boundary!!!

Sketch completion - Delaunay Painting

Assumption - Required boundary is present in the Delaunay Triangulation (Delaunay confirming)
Problem boils down to connecting appropriate points in Delaunay Triangulation



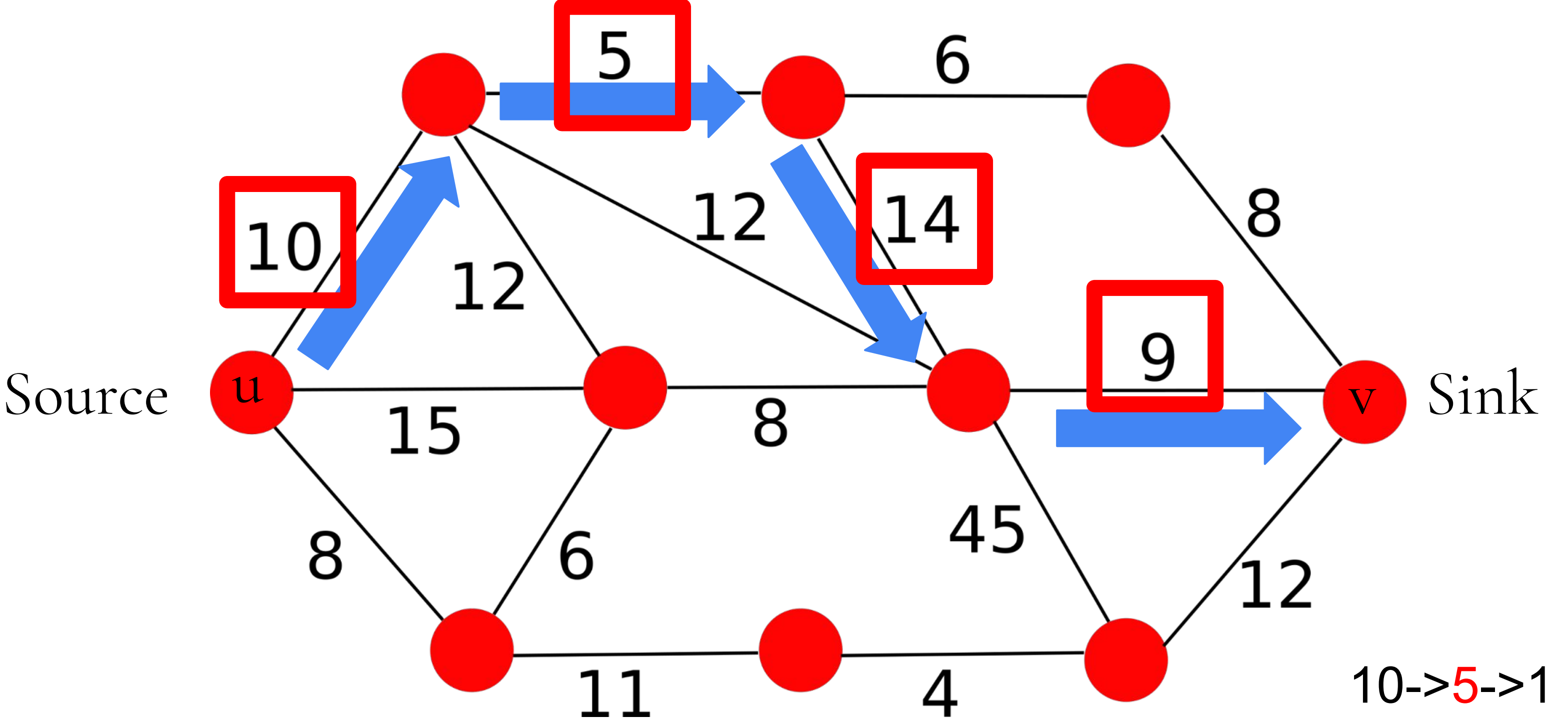
Delaunay Painting - Defining flow



$$Flow(u,v) = max(f(X) : \forall\, paths\, X\ \ from\ u\ to\ v)$$

$$f(X) = min(Weight(u,v) : \forall (u,v) \in X)$$

Delaunay Painting - Defining flow

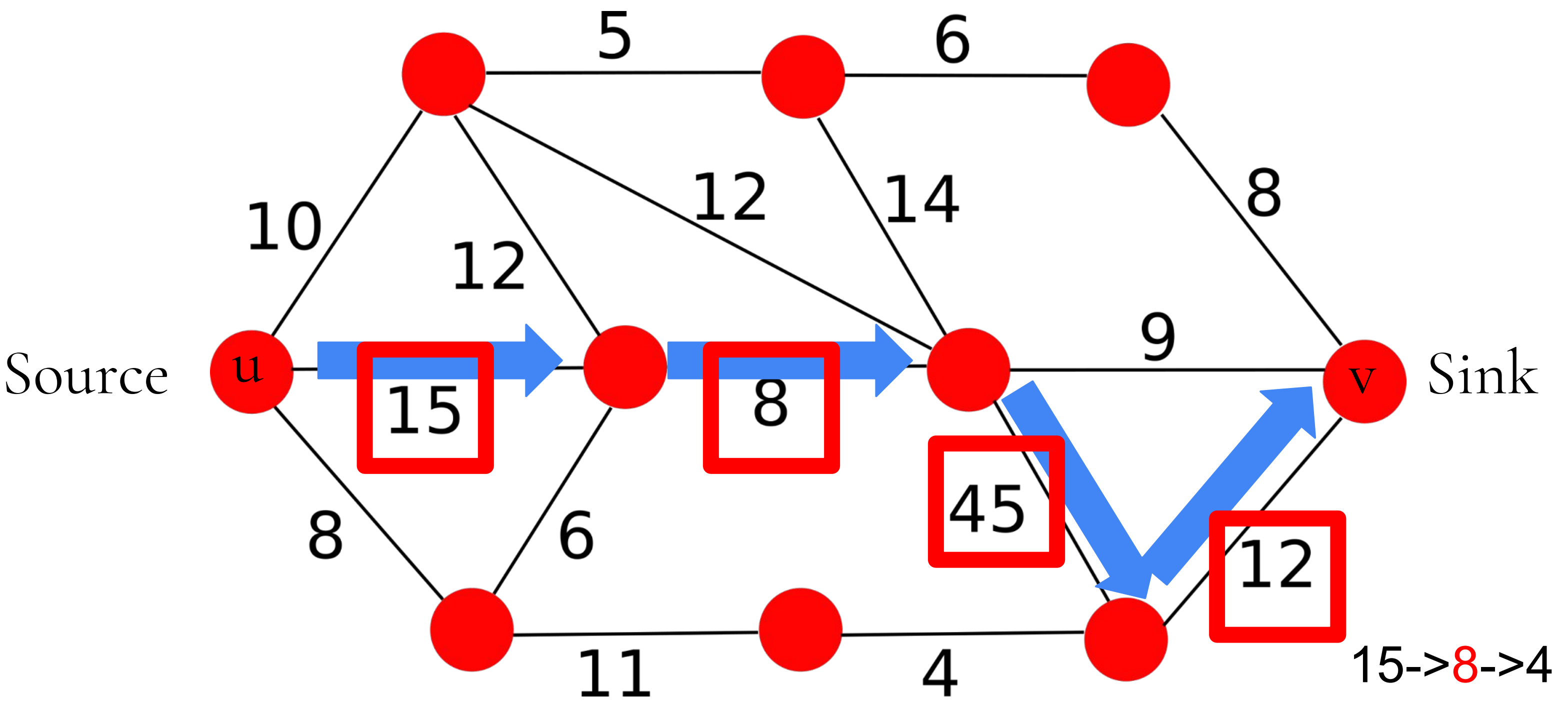


10->5->14->9

$$Flow(u,v) = max(f(X) : \forall\, paths\, X\ from\ u\ to\ v)$$

$$f(X) = min(Weight(u,v) : \forall (u,v) \in X)$$

Delaunay Painting - Defining flow

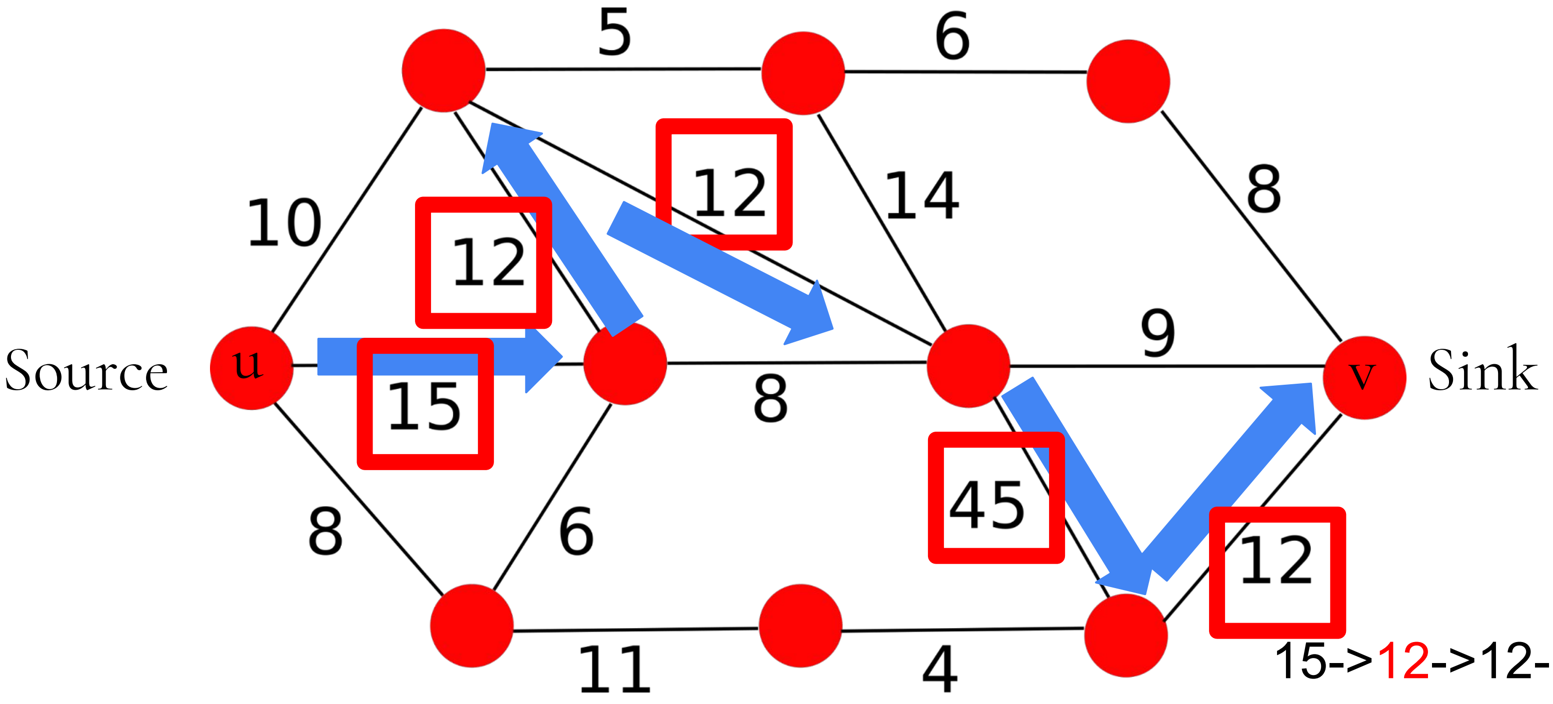


15->8->45->12

$$Flow(u,v) = max(f(X) : \forall \, paths \, X \;\, from \; u \; to \; v)$$

$$f(X) = min(Weight(u,v) : \forall (u,v) \in X)$$

Delaunay Painting - Defining flow

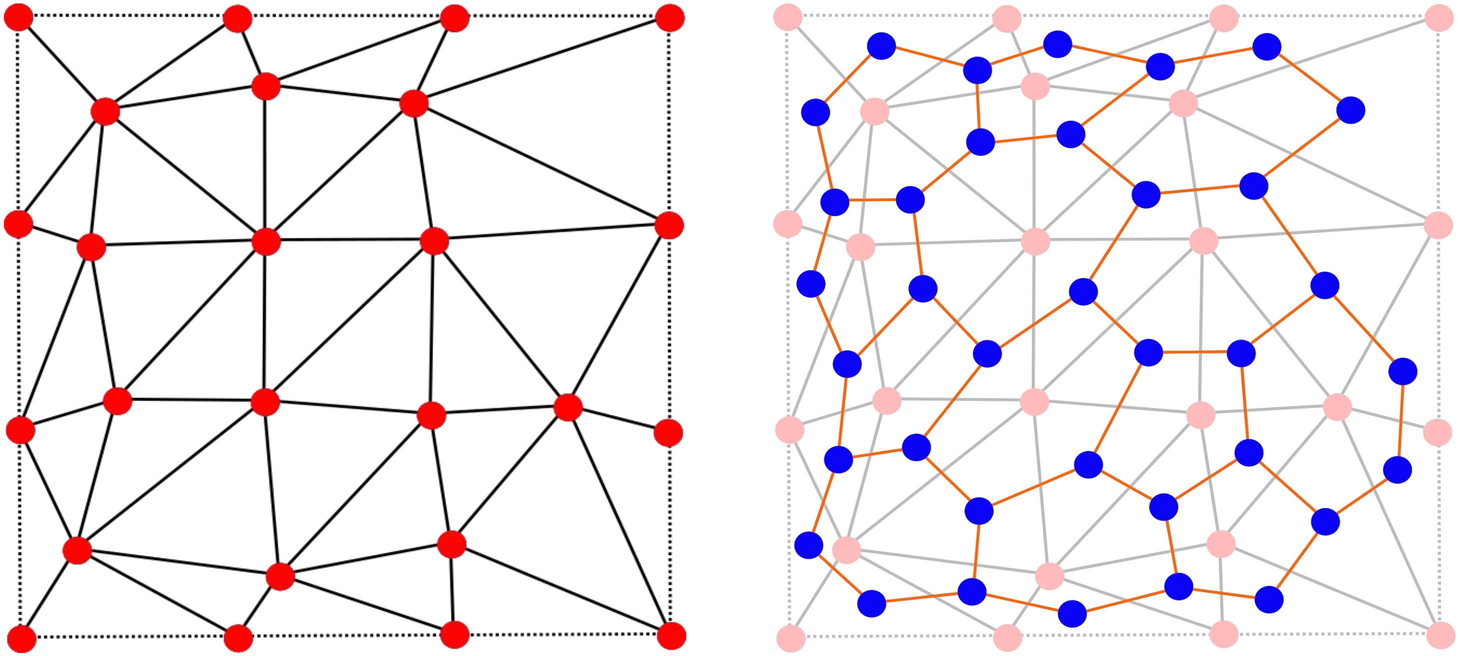


15->12->12->45->12

$$Flow(u,v) = max(f(X) : \forall\, paths\, X\ from\ u\ to\ v)$$

$$f(X) = min(Weight(u,v) : \forall (u,v) \in X)$$

Delaunay Painting - Creating a graph

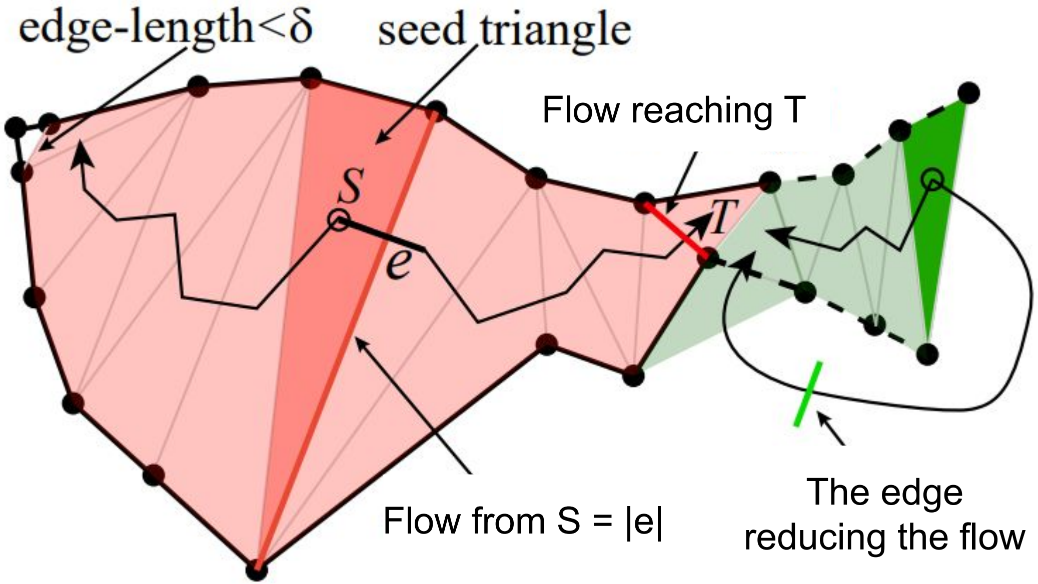


Weighted dual graph

Create a color_strength for all vertices and initialize it as ‘0’

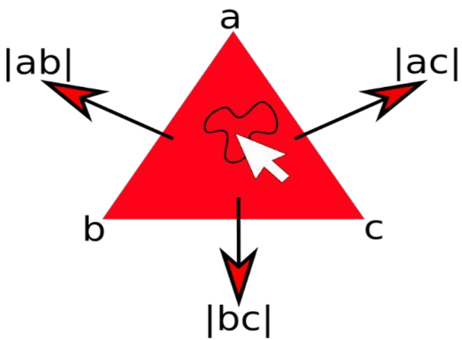
Delaunay Painting - Logic

Recursive color spreading
Color strength updation
Priority queue based exploration



Delaunay Painting - Updation

Color starting from a triangle ‘v’

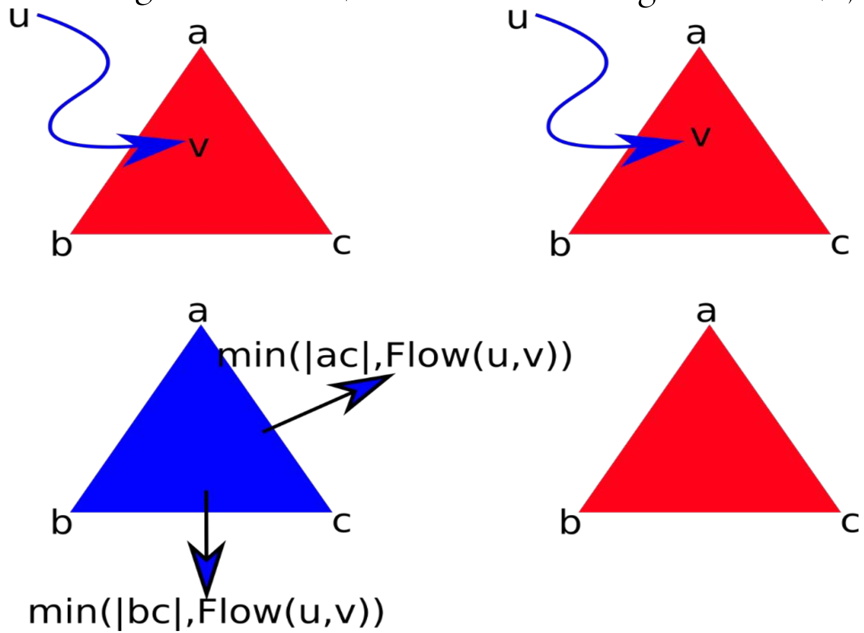


Color_strength = ∞
Spread color to all the neighbors

Color coming from ‘u’ to the triangle ‘v’

Color_strength < Flow(u,v)

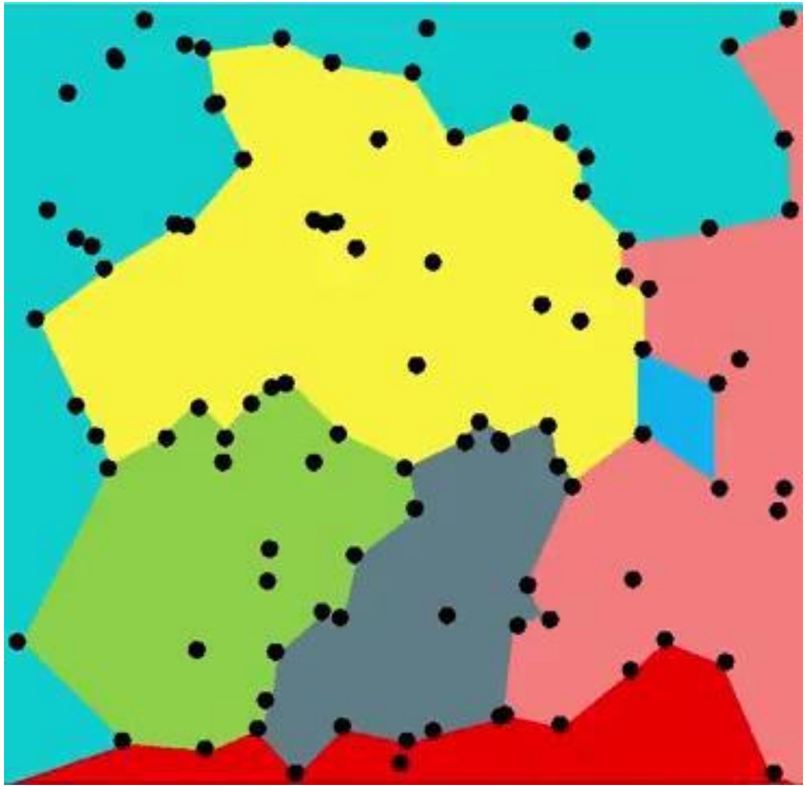
Color_strength > Flow(u,v)



Color_strength = Flow(u,v)
Spread to the other 2 neighbors

Do nothing

Delaunay Painting - Demo

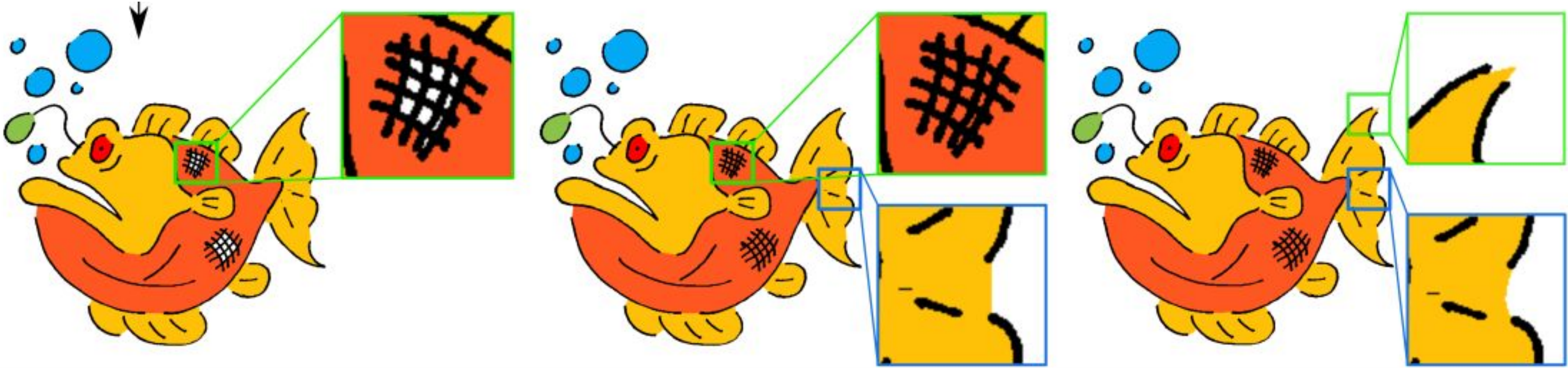


Delaunay Painting - Demo

Delaunay Painting - We can do a lot more

Are we missing something?
Too much work if the sketch has additional information like shading
Not aesthetically appealing

Delaunay Painting - Additional functionalities



How to address them?

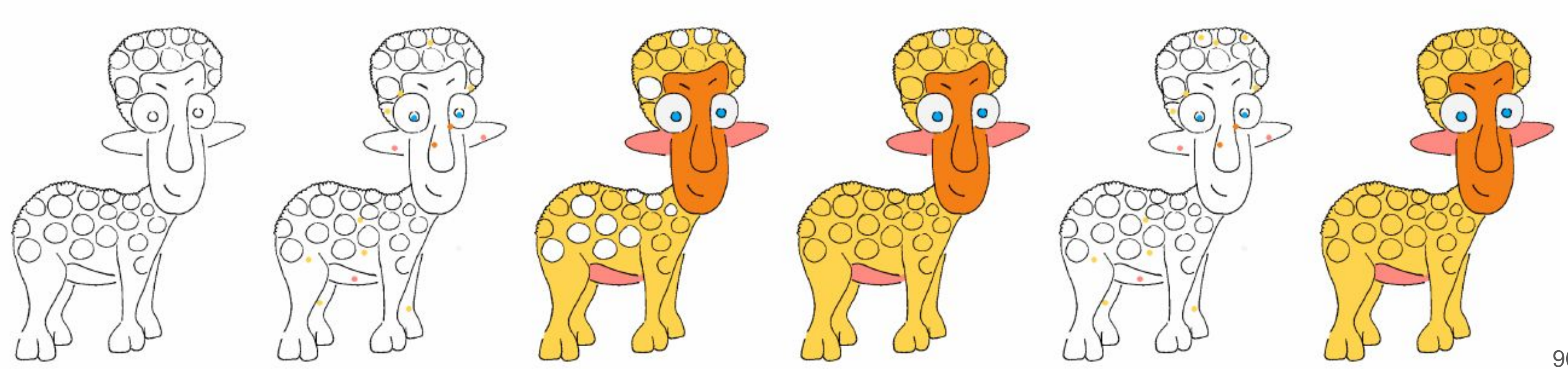
Too much work if the sketch has additional information like shading - Diffuse colors

Not aesthetically appealing - Give finishing using an aesthetic curve completion

Delaunay Painting - Color Diffusion

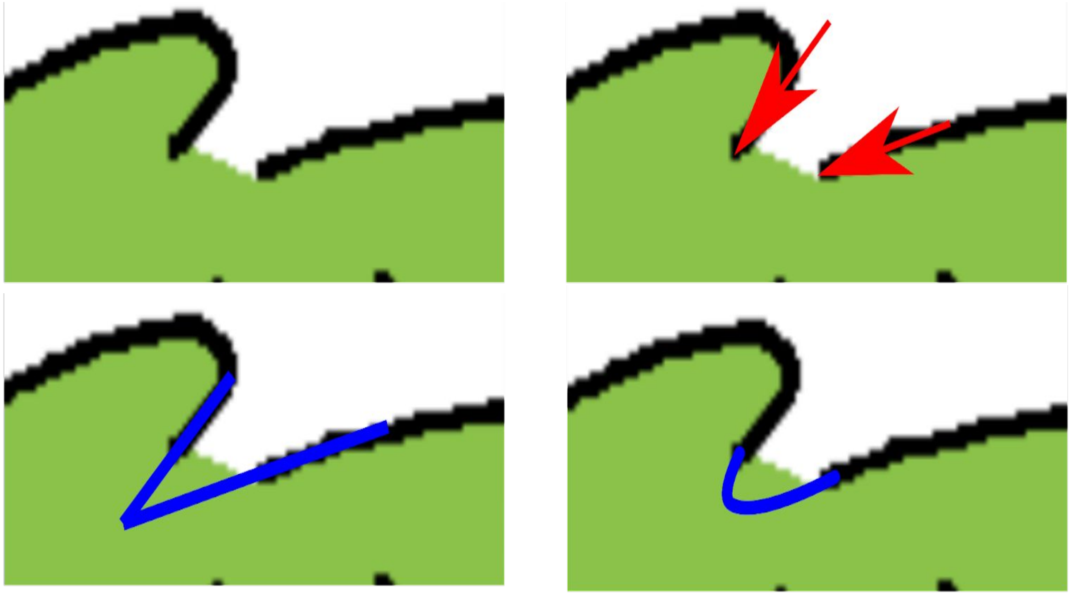
Presence of shading information (hatching) makes the coloring process time consuming
Artists usually use a comparatively smaller brush size for shading/hatching

Bipartite the regions into two - colored and uncolored
Recursively update the bipartition



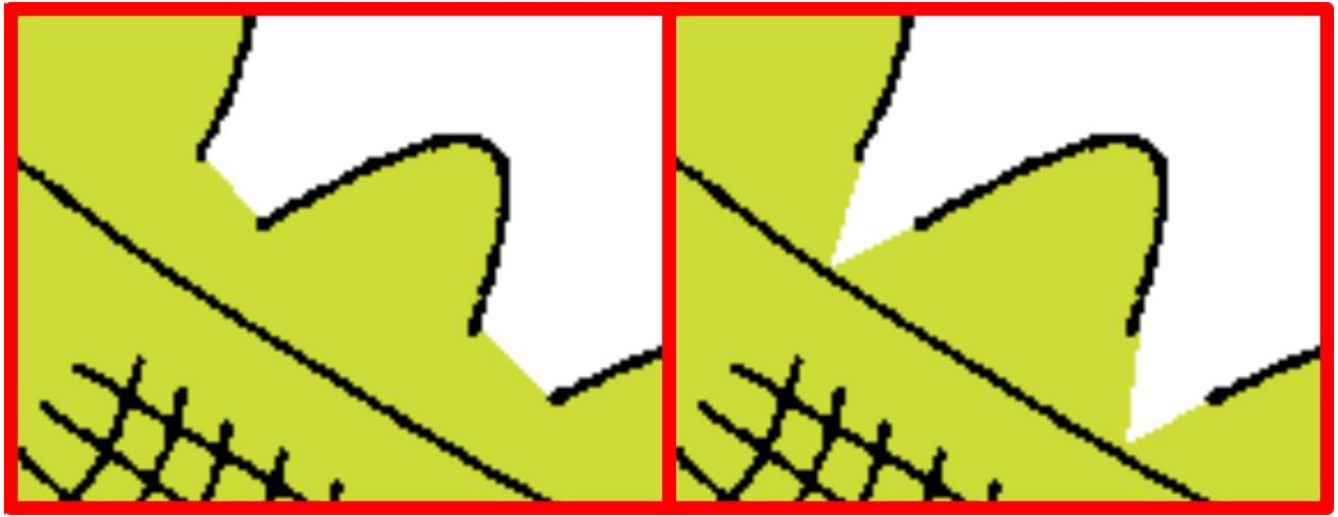
Delaunay Painting - Aesthetic curve completion

Splitting points - Edges having different colors on its associated triangles
Sharp corner or smooth curve?
Decision based on tangent approximation



Delaunay Painting - Sharp corner heuristics

Angle constraint - angle between tangents less than $\pi/3$
Perpendicular constraint - intersection to edge distance is less than 2*||edge||
Linearity constraint - pixels near the endpoints are linearly arranged



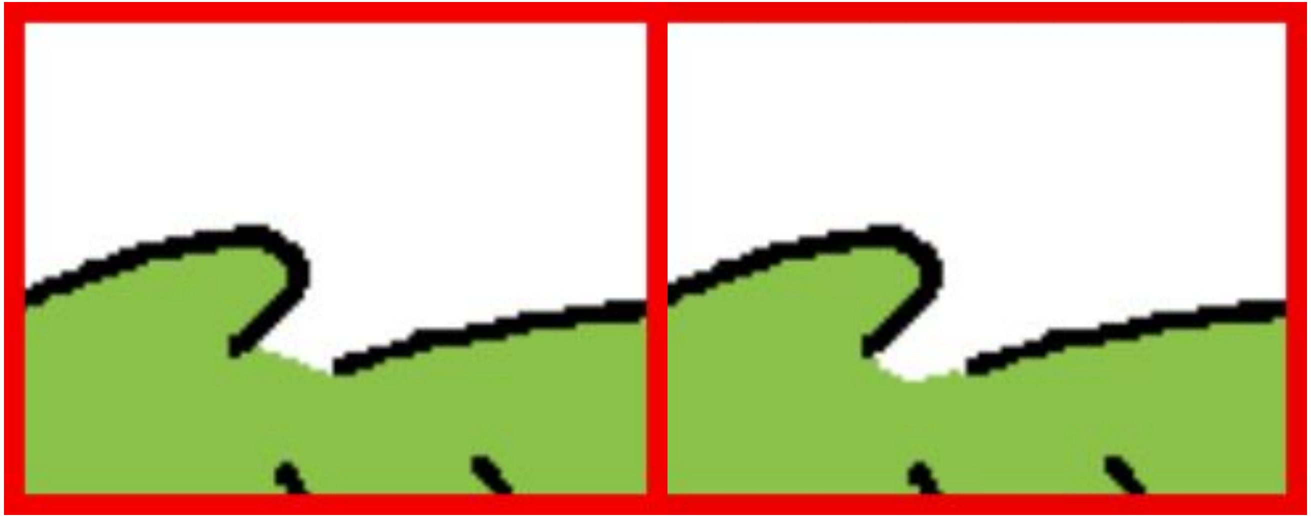
All three constraints are qualified -> sharp corner, else -> smooth curve

Delaunay Painting - SIMVC curves

Perceptually pleasing contour

Scale Invariant Minimum Variation Curve - to form more circular arcs

$$E_{SIMVC-Entem} = \frac{(\int ds)^5}{\|B-A\|^2} \int \left(\frac{d\kappa(s)}{ds}\right)^2 ds$$



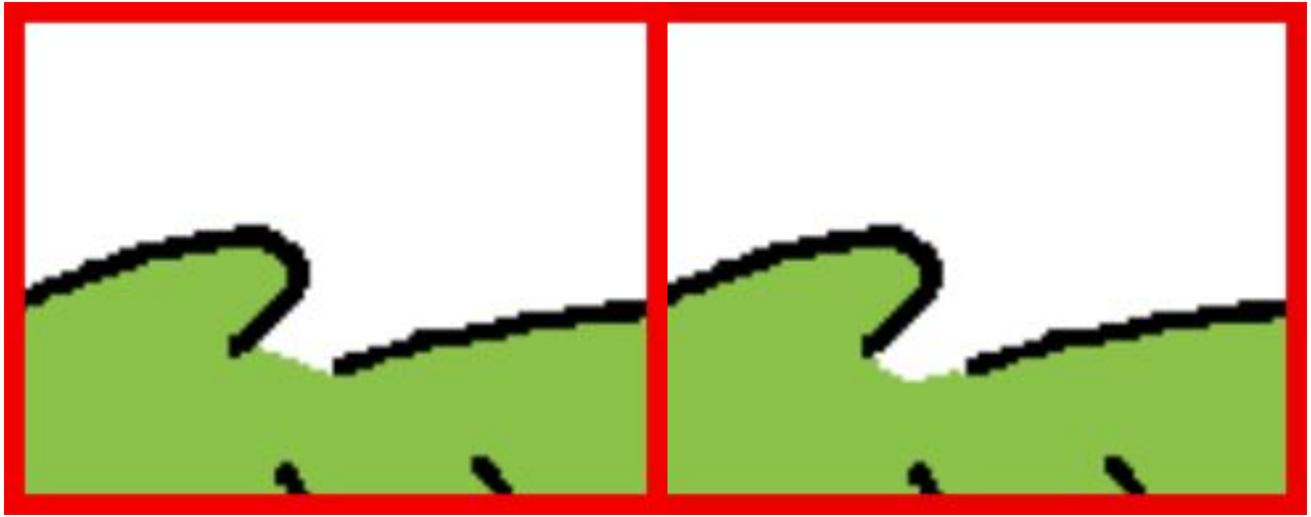
Delaunay Painting - SIMVC curves

Perceptually pleasing contour

Scale Invariant Minimum Variation Curve - to form more circular arcs

$$E_{SIMVC-Entem} = \frac{(\int ds)^5}{||B-A||^2} \int \left(\frac{d\kappa(s)}{ds}\right)^2 ds$$

How small the curve is?



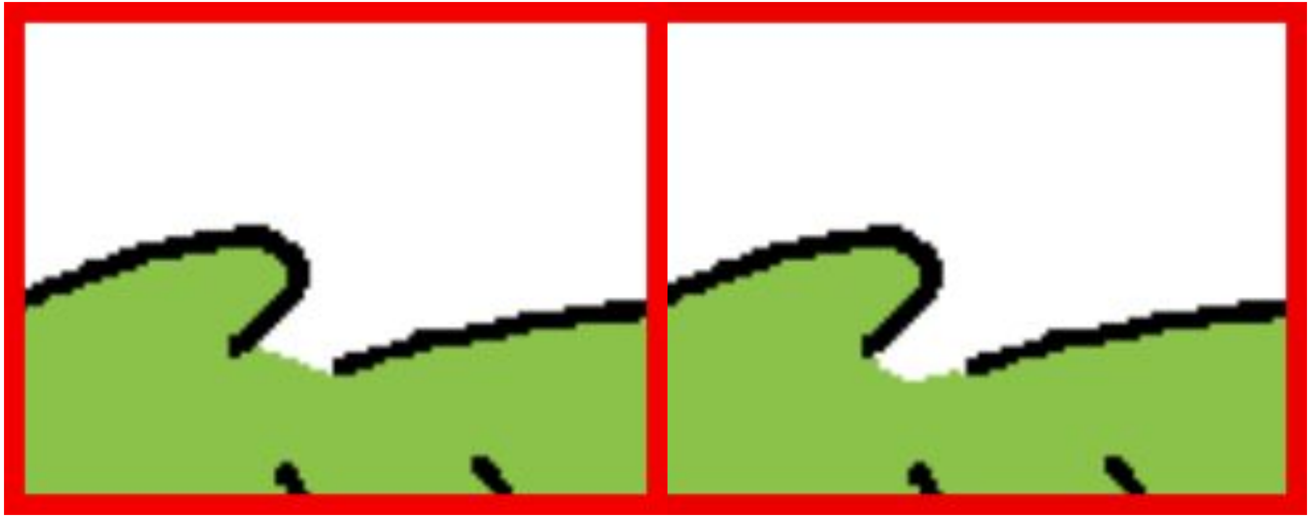
Delaunay Painting - SIMVC curves

Perceptually pleasing contour

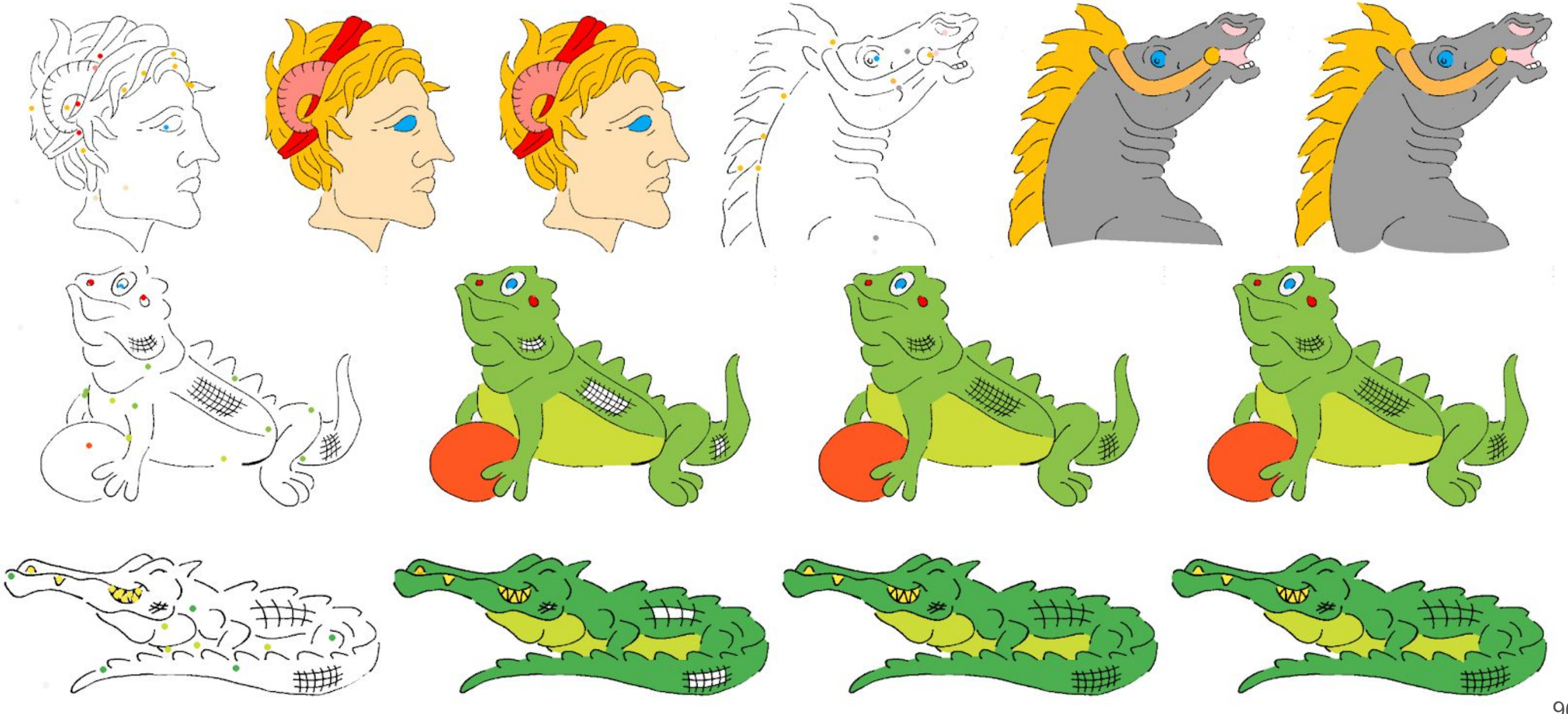
Scale Invariant Minimum Variation Curve - to form more circular arcs

$$E_{SIMVC-Entem} = \frac{(\int ds)^5}{\|B-A\|^2} \int \left(\frac{d\kappa(s)}{ds}\right)^2 ds$$

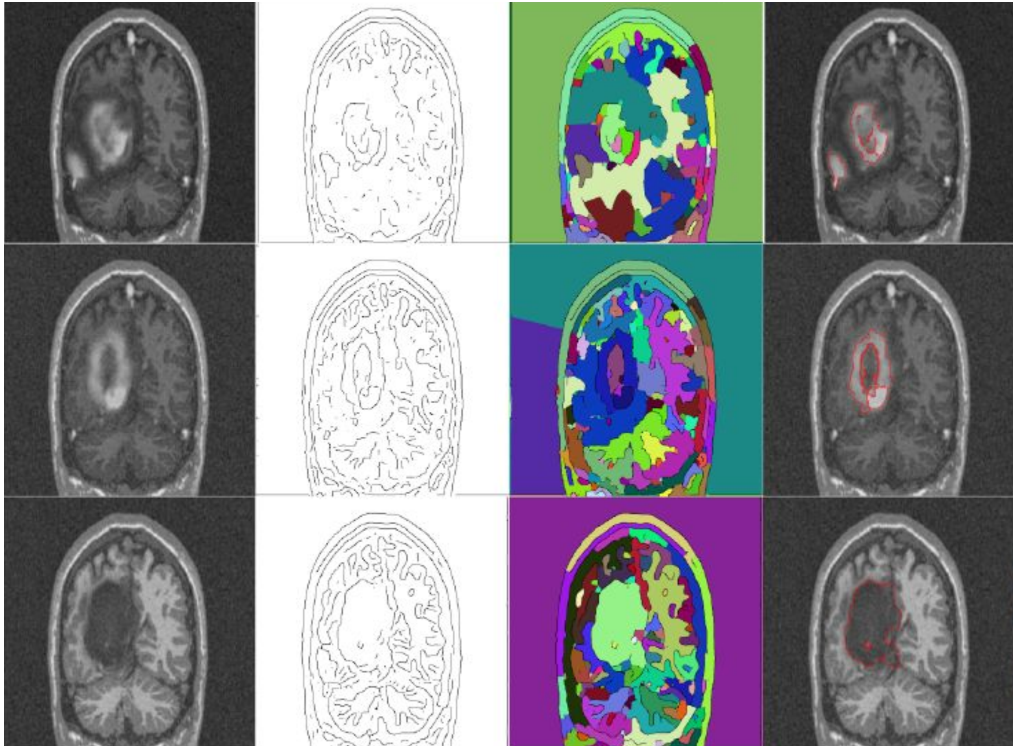
How much curved?



Delaunay Painting - More Results

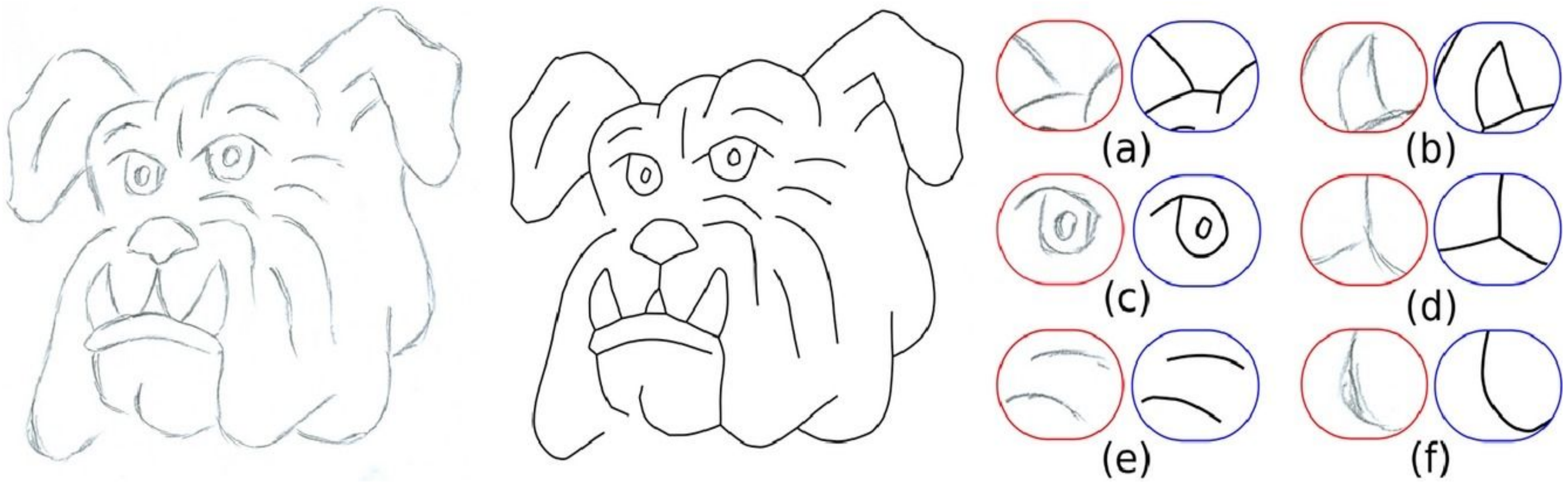


Delaunay Painting - Labelling medical images

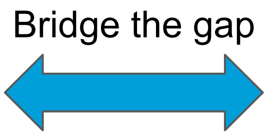


MRI Images @ Prastawa M et al.

Sketch Simplification



Rough Sketches (Artists) — Bridge the gap ⟷ — **Vector Sketches (Developers)**

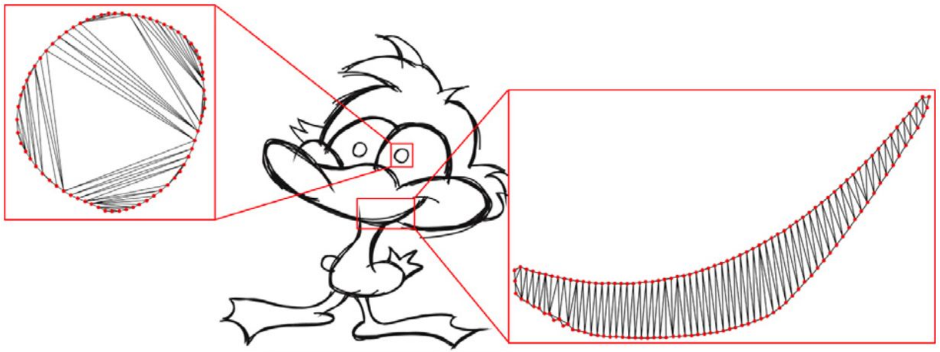


Rough Sketches (Artists)	Vector Sketches (Developers)
Easy to draw	Easy to edit/manipulate
Artistic freedom to fix mistakes	Small and crisp
Shape refinement	Scalability

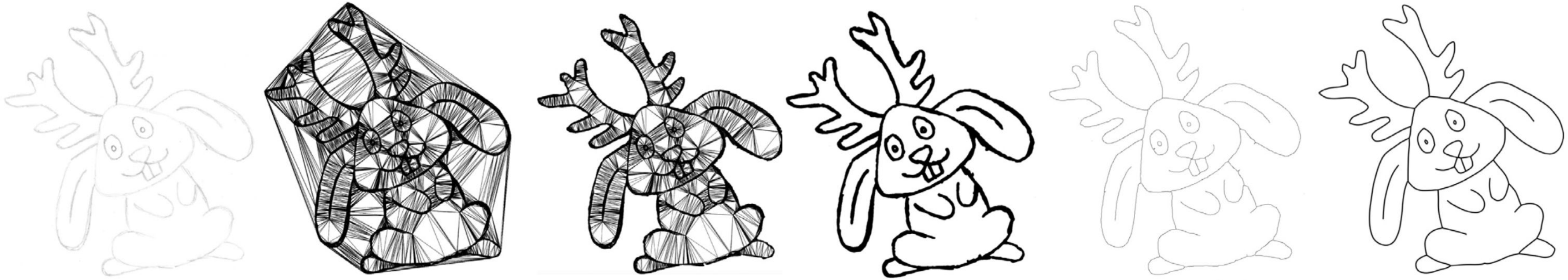
Sketch Simplification - Automatic triangle growing

Delaunay triangles inside regions will have a “fat triangle”

Delaunay triangles inside adjacent strokes will have only “thin triangles”



Overview:

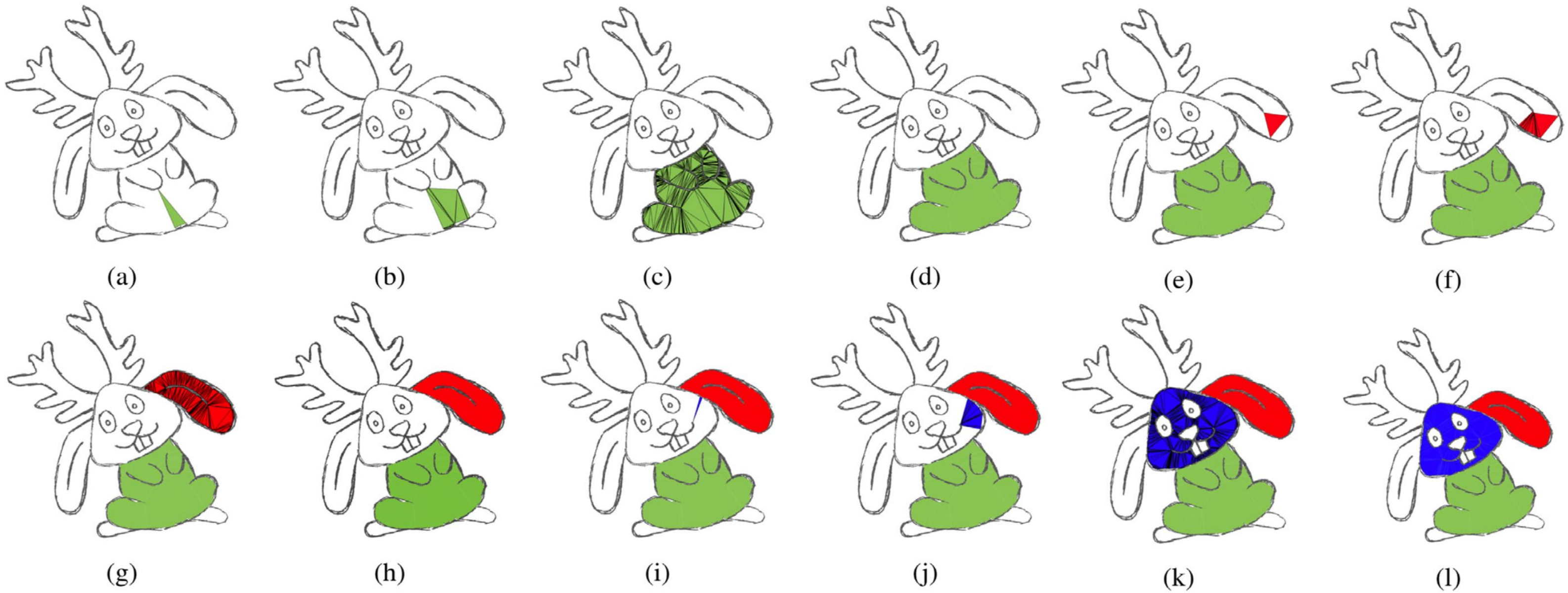


Automatic triangle growing

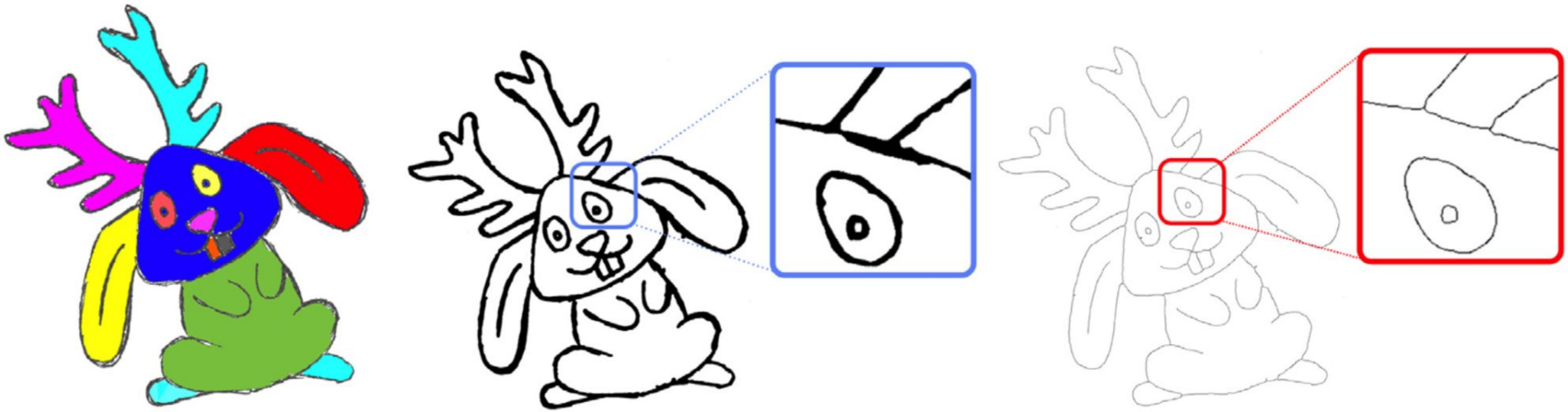
Algorithm: Start from the largest "valid" ungrouped triangle

Recursively group neighboring triangles until a "condition" is satisfied

Restart the procedure



Automatic triangle growing



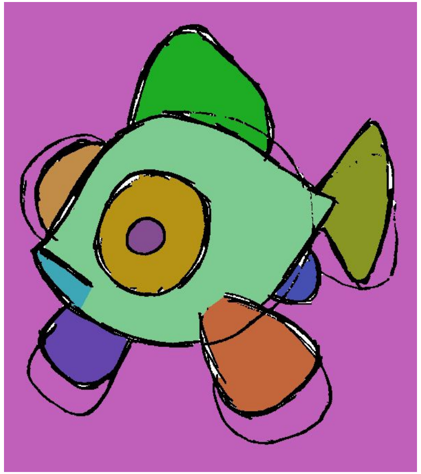
Stop the procedure when there are no more “valid” ungrouped triangles
Pick all the ungrouped triangles (lies inside adjacent strokes) - group and color them
Compute the skeleton of this colored group, and fit cubic Bezier curves

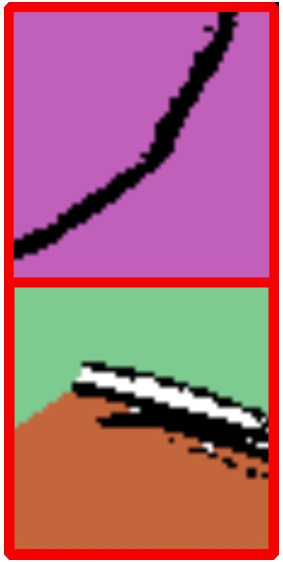
Sketch Simplification - Color by numbers

Perception plays an important role in simplification - Not available in Automatic triangle grouping
Design sketches usually have construction lines
Idea: Make users annotate the parts that should be grouped

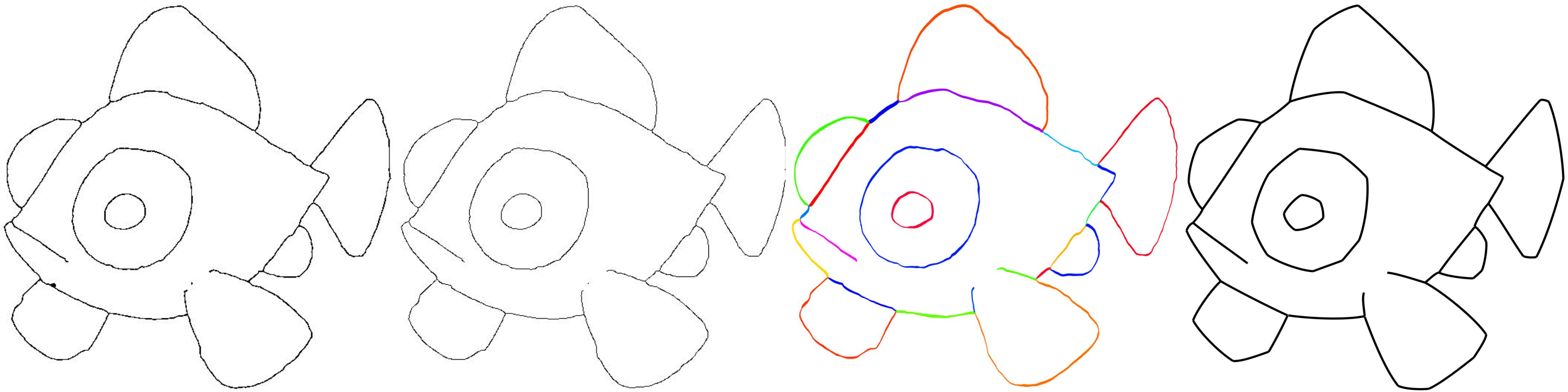
Playful interface: Ask the user to give same color on the opposite sides of a required stroke

And we already know how to do it!!! - Delaunay painting



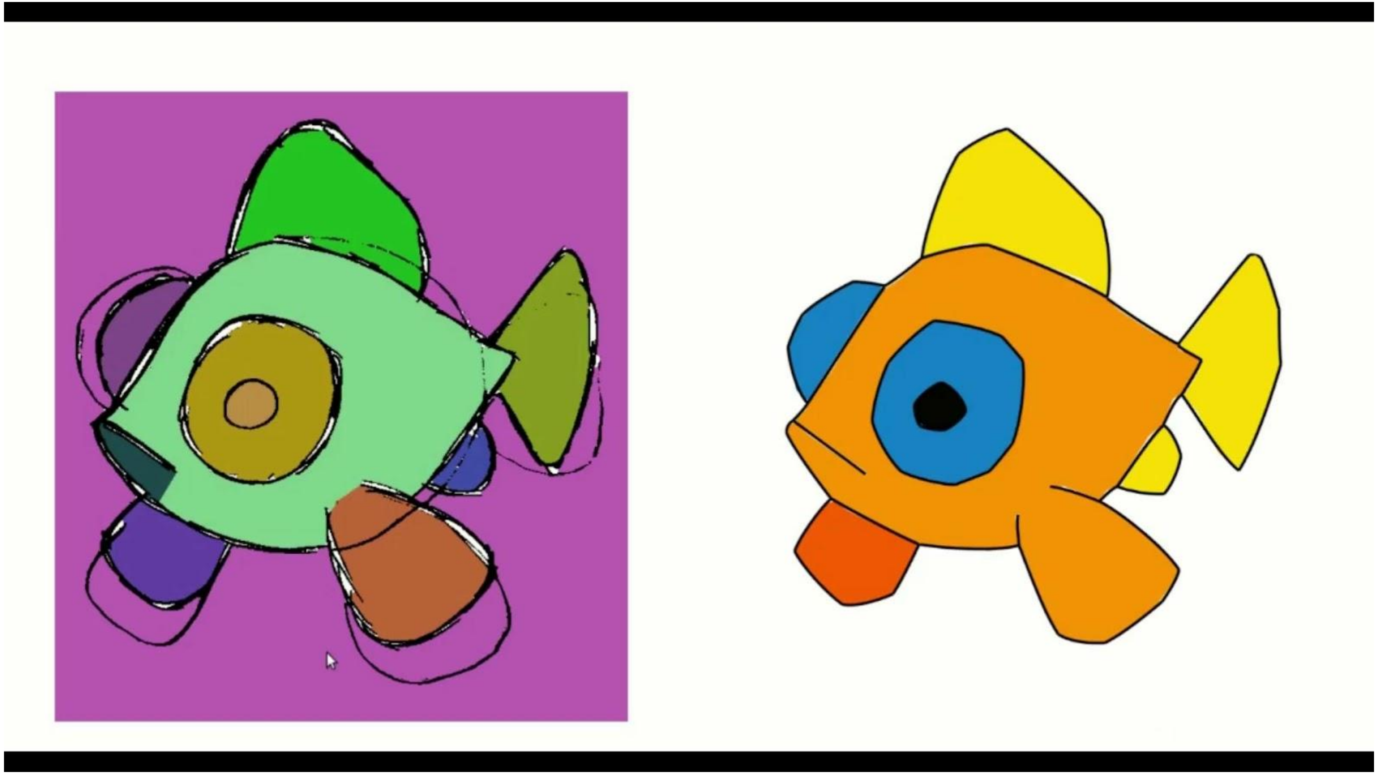


Color by numbers - Complete procedure

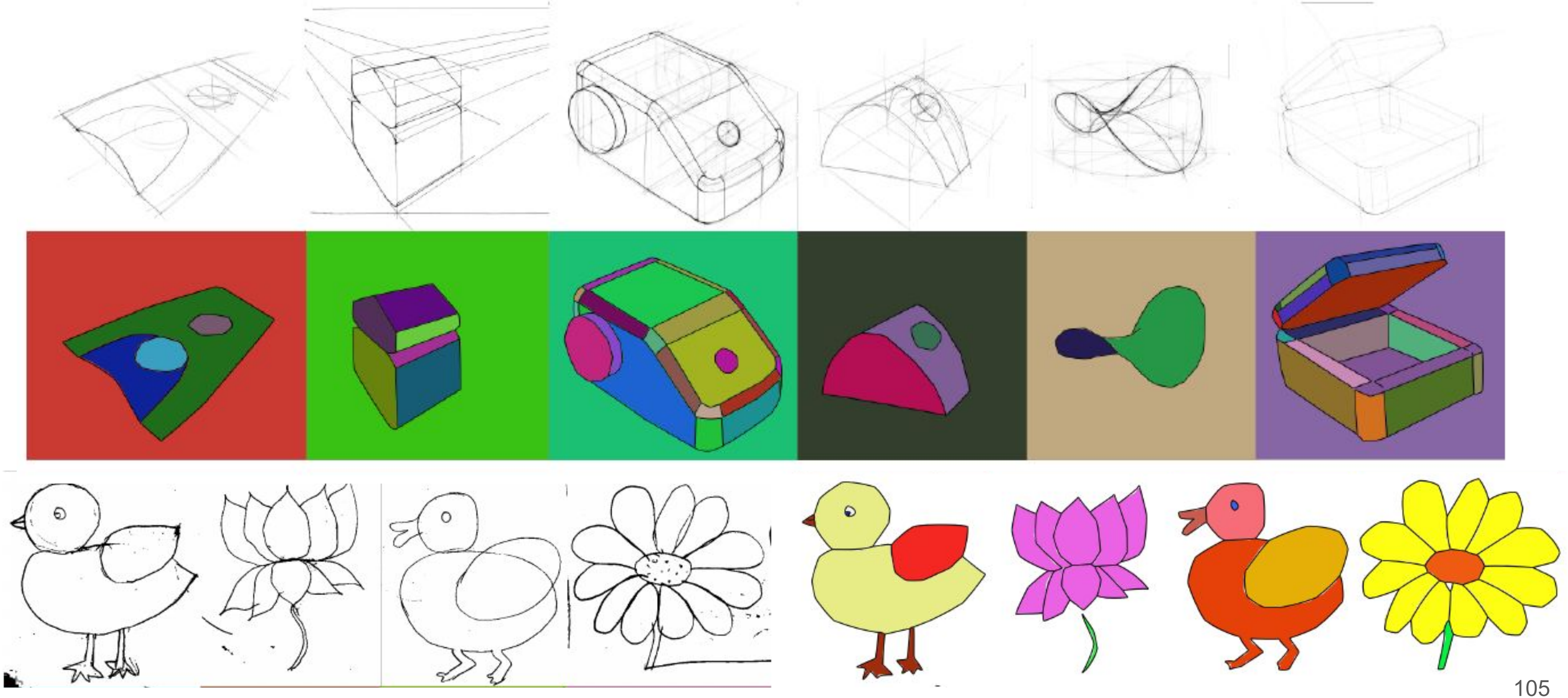


Grouping the strokes | Finding the skeleton | Constructing a curve network | Bezier curve fitting

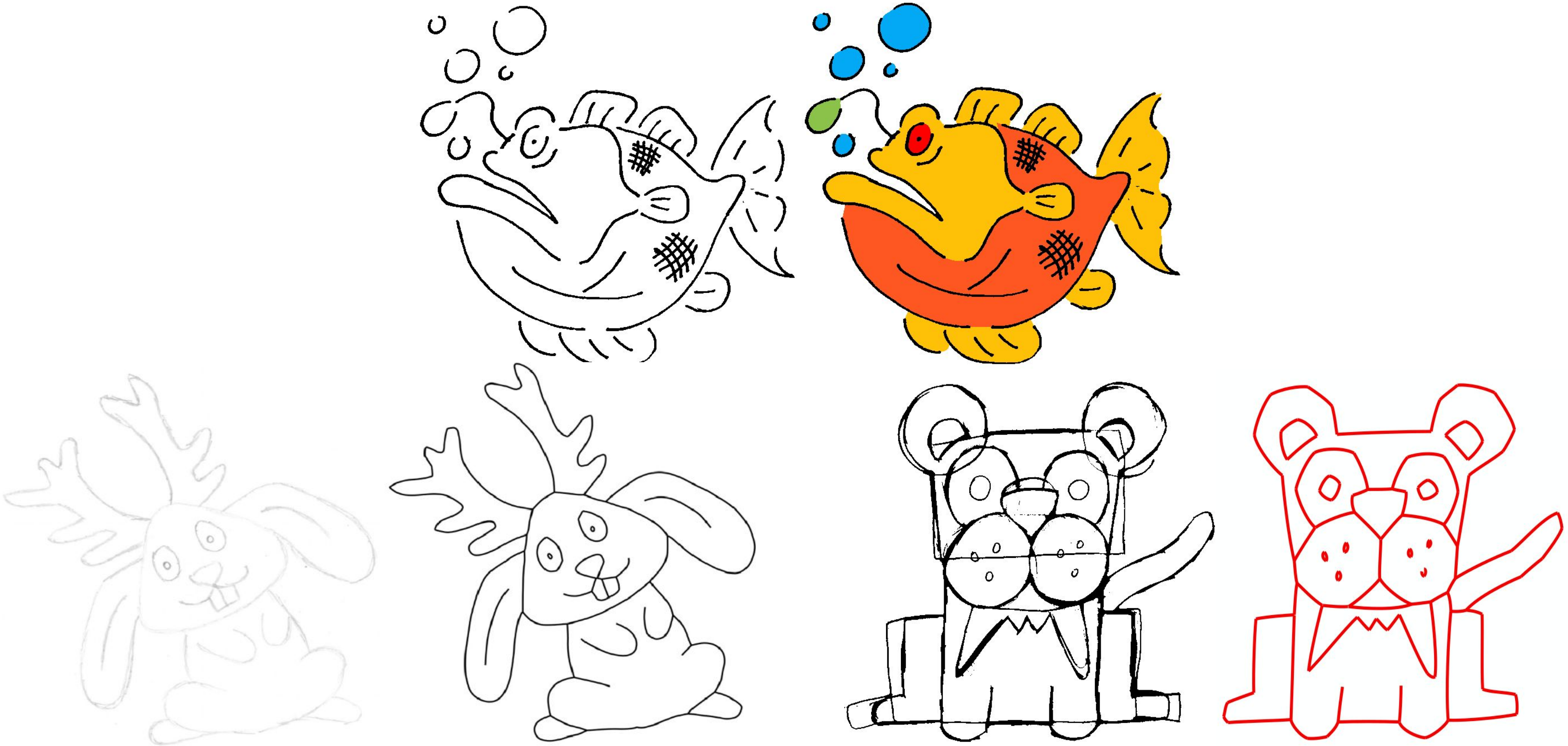
Color by numbers - Demo



Color by numbers - Results



Reconstruction from Sketches



Outline

Topic: Shape Characterization

- Shape/Region Reconstruction
- HVS based Algorithms
- Delaunay based Algorithms
- Sampling Models
- Evaluation Practices
- Future Directions

Presenter:

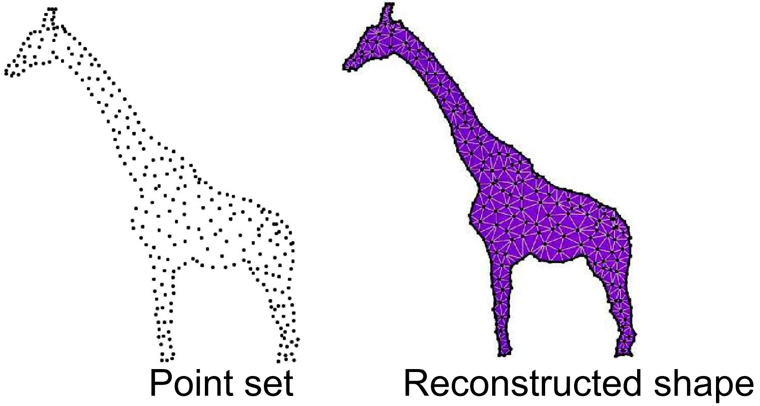
Jiju Peethambaran

Assistant Professor

Saint Mary's University, Halifax

Shape Characterization or Region Reconstruction

- Given a finite set of points sampled from a planar object or region, construct a polygonal boundary that best approximates the object or region



- **Inputs** are known as dot pattern/area samples/region samples
- **Outputs**: graphs or polygons
- Compared to curve reconstruction, more signals (or samples) about the shape is available

Applications

- Computer graphics- geometric modeling[1]

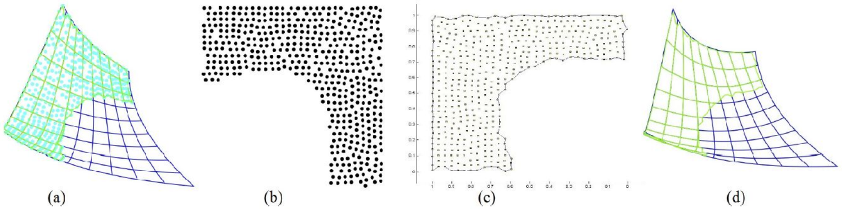
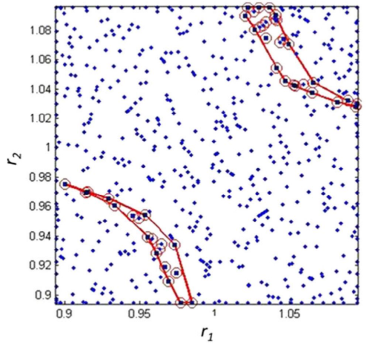


Figure : (a) Points on surface with constraints, (b) Points in parametric 2D space, (c) Reconstruction, (d) Trimmed patch

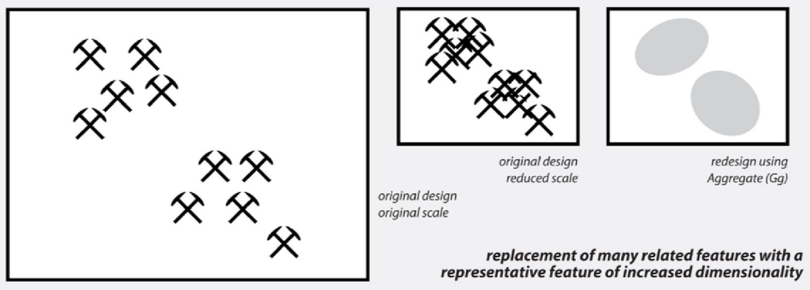
- Identification of island failure regions in the design space of reliability-based crash optimization[2]



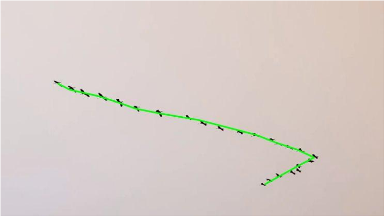
[1] Sundar et al. 2014, “Foot point distance as a measure of distance computation between curves and surfaces”, *Computers & Graphics*
[2] Ganapathy et al. 2015, Alpha shape-based design space decomposition for island failure regions in reliability-based design”, *Struct. Multidisc. Opt.*

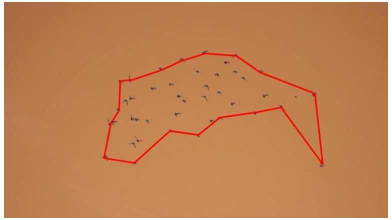
Applications

- Map generalization- e.g., aggregation of buildings to form single polygon[3]



- Outline of trees or flock of birds[4]

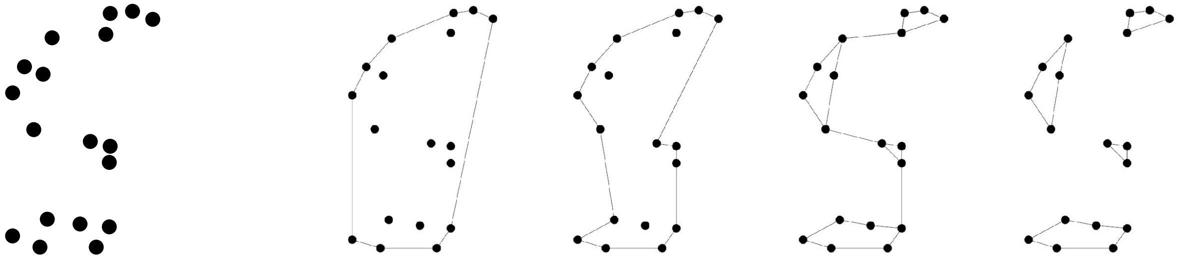




[3] Roth et al. 2014, “a typology of operators for maintaining legible map designs at multiple scales”, *Cartographic Perspectives*
[4] Pandey et al. 2021, “Towards Video based Collective Motion Analysis through Shape Tracking and Matching”, *IET Electronic Letters*

Challenges

- **Ill-posed/Vague problem**[5] – a precise mathematical definition for ‘shape’ is almost impossible
    - Rich variety of shapes and forms
    - Heterogeneity of point set sampling (density and distribution)
- **Different interpretations** for ‘shape’ based on human cognition, visual perception and application demands



[5] Edelsbrunner 1998, “Shape Reconstruction with Delaunay Complex”, *LATIN*

Region Reconstruction Criteria [6]

- Should every member fall within the region or outliers permitted?

- Should any points fall on the boundary, or they must fall in the interior?

[6] Galton et al. 2006, "What is the Region Occupied by a Set of Points?", *GIScience*

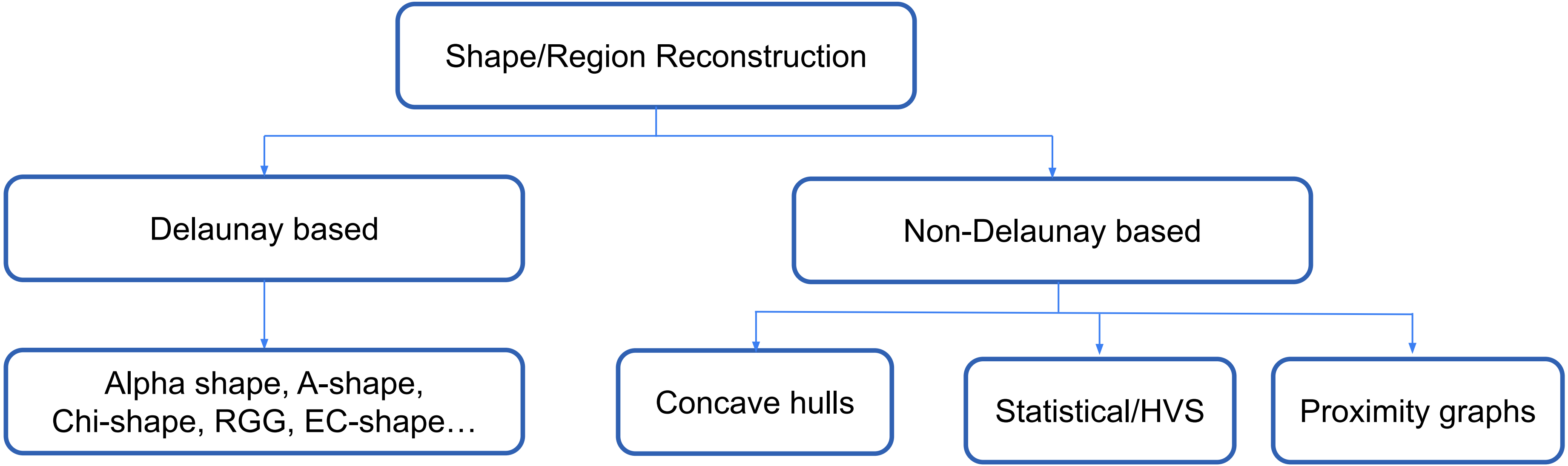
Region Reconstruction Criteria [6]

- Should the region boundary be polygonal, or can it be smooth and curved?

- Should the region boundary be a simple polygon?

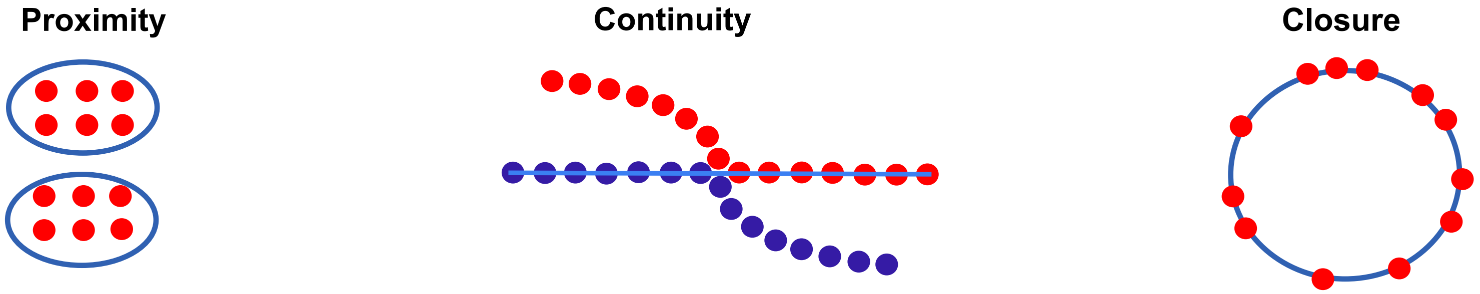
[6] Galton et al. 2006, “What is the Region Occupied by a Set of Points?”, *GIScience*

Classification



Human Visual Perception

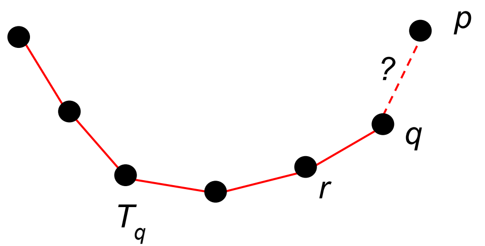
- Gestalt Laws of visual perception



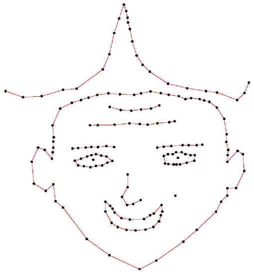
DISCUR[7]

- A vision function that encodes p's relation to T_q and the edge (r, q)

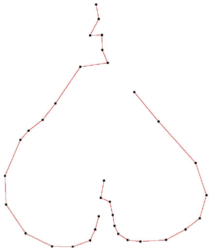
$$E[p, T_q] = h_d \frac{h}{s} \left(1 + \frac{h_d}{\sigma_d}\right)^{\frac{\sigma_d}{h_d}}$$



- If $d(p, q) < E[p, T_q]$, connect p to q
- Parameter free algorithm
- Open/closed curves, multiple curves, sharp corners
- Dense sampling at sharp corners



DISCUR result



A failure case

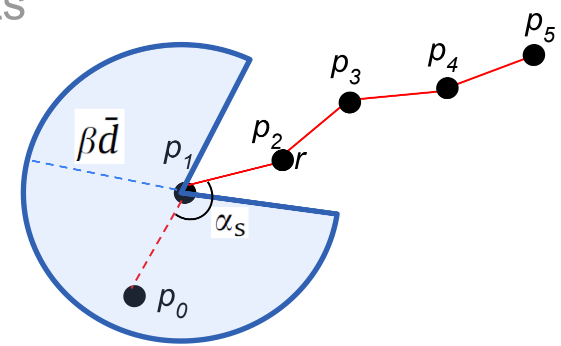
[7] Zeng et al. 2008, “A distance-based parameter free algorithm for curve reconstruction”, *Computer Aided Design*

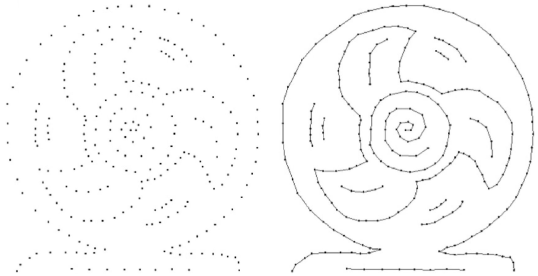
VICUR[8]

- DISCUR limitation: arbitrary selection of candidate points
- Vision function that encodes proximity and continuity

$$E[p, T_{p_1}] = \left[c\left(\frac{\alpha_s}{\bar{\alpha}} - 1\right)^2 + \left(\frac{1-c}{4}\right)\left(\frac{d_s}{\bar{d}+\sigma}\right)^2 + 1 \right]^{-1}$$

- Candidate point with highest E value is connected
- Sensitive to parameters, e.g., *c* balances the smoothness and nearness



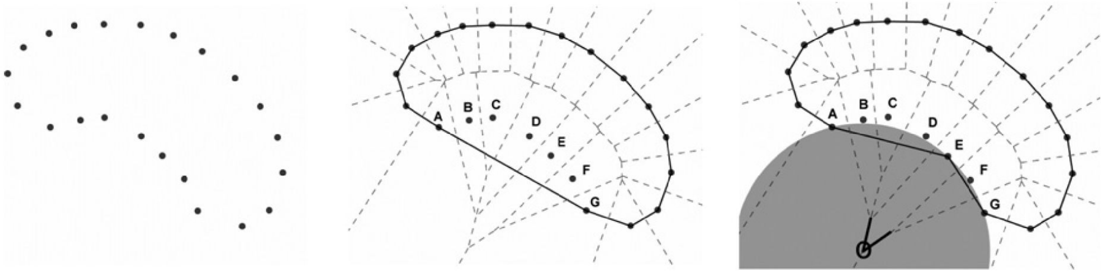


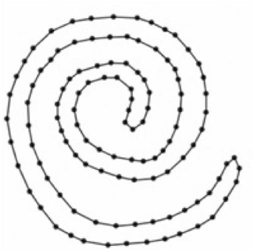
VICUR result [8]

[8] Ngyuen et al. 2008, “A human-vision-based algorithm for curve reconstruction”, *Robotics and Computer-Integrated Manufacturing*

Simple Shape[9]

- Shape according to HVS based on how the concavities are perceived
- Start from the convex hull and carve out the concavity by replacing outer edges by two new edges
- Edge selection is based on
  - Closeness criteria
  - Edge length criteria
  - Angular constraints (angle (EAG)- angle(EGA) must be minimum)



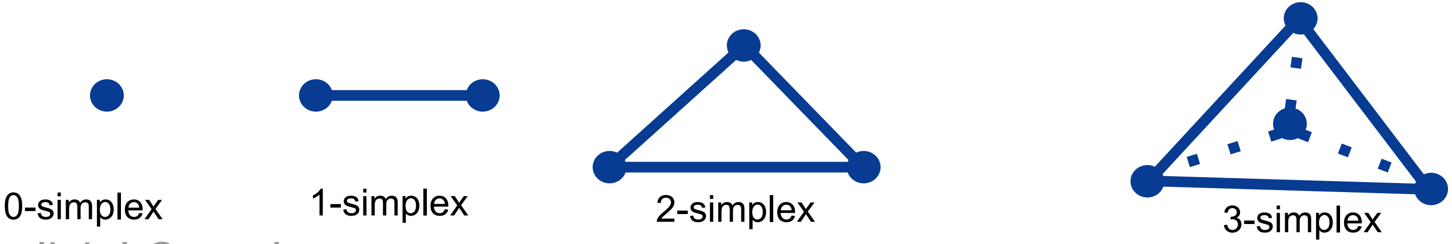


Spiral shape result

[9] Gheibi et al. 2011, "Polygonal shape reconstruction in the plane", *IET Computer Vision*

Simplicial Complex

- k-simplex(σ_k) : non-degenerate convex hull of k+1 geometrically distinct points in in R^d where k <= d.



- Simplicial Complex:

A simplicial complex, $\mathcal{K}$ is a set containing finitely many simplices that satisfies the following two restrictions:

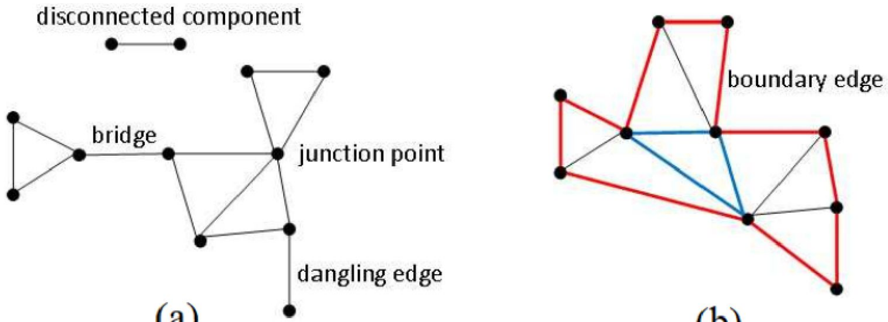
- $\mathcal{K}$ contains every face of every simplex in $\mathcal{K}$;
- For any two simplices, $\sigma, \tau \in \mathcal{K}$, their intersection $\sigma \cap \tau$ is either empty or a common face of σ and τ.

Regular Simplicial Complex

- Regular 2-simplicial complex:

A simplicial 2-complex $\mathcal{K}_2$ is said to be regular if it satisfies the following conditions:

- All the points in $\mathcal{K}_2$ are pairwise connected by a path on the edges.
- It does not contain any junction points, dangling edges or bridges.



(a) (b)

Delaunay Complex

- Given a finite set of points S in R^d, Delaunay complex is a simplicial complex DT(S) consisting only of:
  - all *d*-simplices whose circumspheres are empty of input points
  - all *k*-simplices which are faces of other simplices in DT(S)

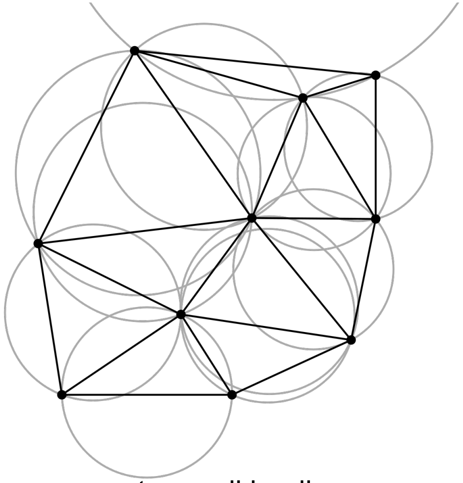


Image courtesy: wikipedia

Alpha Shape

- “Shape formed by a set of points”
- Ice Cream Carving Analogy[$]
  - Ice cream mass occupied in R^d and chocolate points
  - Sphere formed ice cream spoon
  - Carve out ice cream without bumping into the chocolate points
  - Carving spoon of small radius points
  - Carving spoon with huge radius convex hull

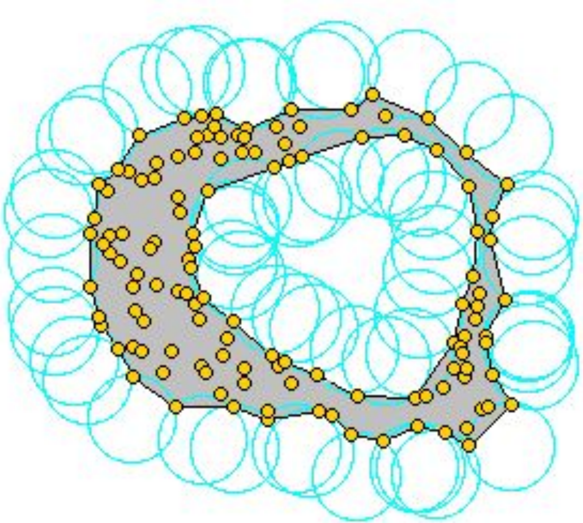
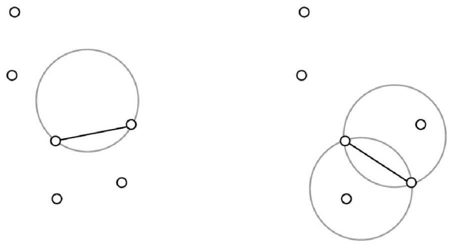


Image courtesy: CGAL Alpha shapes

[$] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graph.*, 13(1):43–72, January 1994.

Alpha Shape[10]

- α-exposed simplex: A k-simplex is α-exposed if there exists an empty α-ball b with $\sigma_k = \partial b \cap S$



α-exposed   not α-exposed

DEFINITION The boundary ∂S_α of the α-shape of the point set S consists of all k-simplices of S for $0 \le k < d$ which are α-exposed,

$$\partial S_\alpha = \{\sigma_k \mid k \le d, (v_0, v_1, \ldots, v_k) \subseteq S \; and \; \sigma_k \; are \; \alpha - exposed\}$$

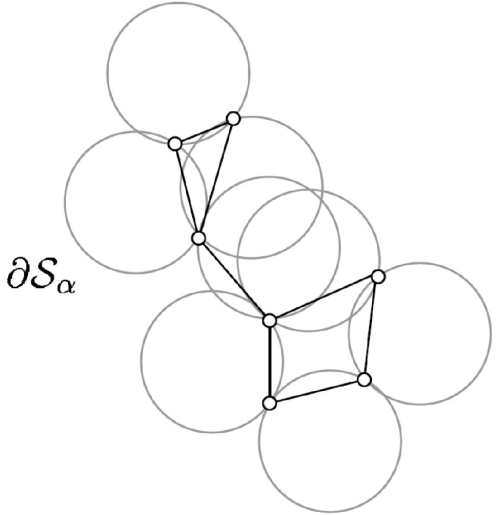
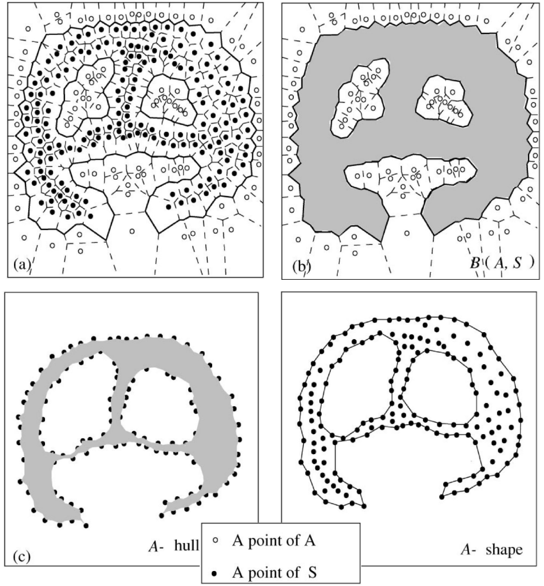


Image courtesy: [11]

[10] Edelsbrunner et al. 1983, “On the shape of a set of points in the plane”, *IEEE Transactions on Information Theory*
[11] Fischer K., “Introduction to Alpha Shapes”, *Technical Report, Stanford University*

A-Shape[12]

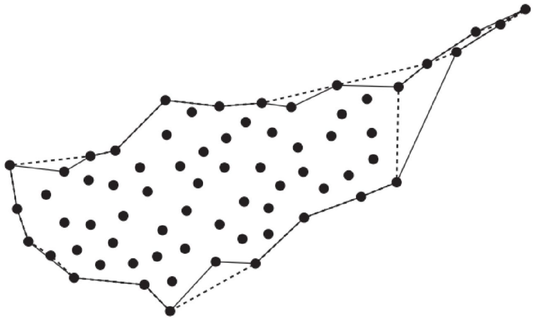
- Let $\mathcal{A} \subseteq R^2$ is a finite set of points, S
- $\mathcal{A}$-shape is generated by connecting $p, q \in S$ if there is an empty circle that passes through *p, q* and *a* $\in \mathcal{A}$
- Two parameter family of point sets $\mathcal{A} = \mathcal{A}(\alpha, t)$
  - $t \in [0, 1]$ is a local density measure
  - $\alpha \geq 0$ level of detail of the shape



[12] Melkemi et al. 2000, “Computing the shape of a planar points set.”, *Pattern Recognition*

Chi Shape[13]

- Simple polygon that characterizes the shape of point set, S.
- Start with the Delaunay Triangulation of S.
- Repeatedly remove longest boundary edges greater than a threshold *l* subjected to regularity constraints.
- Generates a regular polygon that contains S.

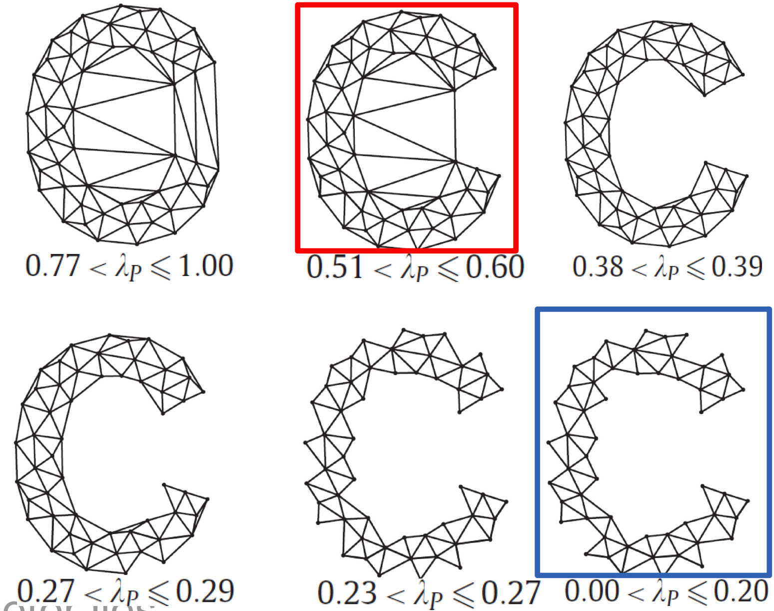


[13] Duckham et al. 2008, “Efficient generation of simple polygons for characterizing the shape of a set of points in the plane.”, *Pattern Recognition*

Chi Shape

- How to select *l*?

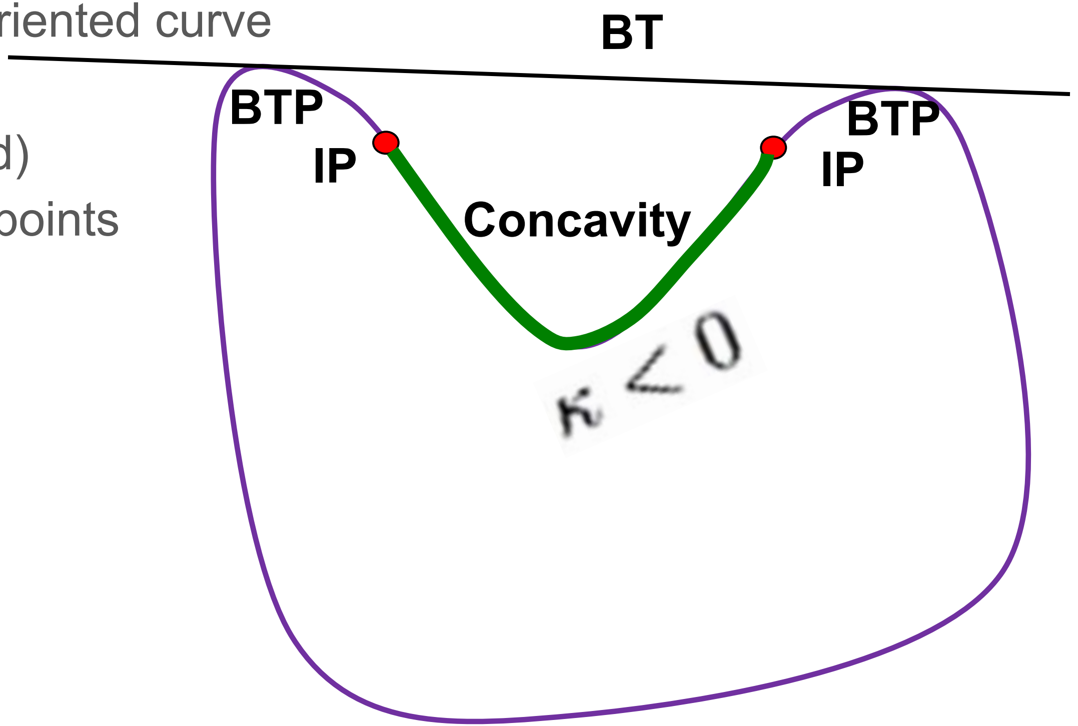
$$\lambda_P = \begin{cases} 1 & \text{if } l \geqslant \max_P \\ \dfrac{l - \min_P}{\max_P - \min_P} & \text{if } \min_P \leqslant l < \max_P \\ 0 & \text{if } l < \min_P \end{cases}$$



- Good characterization via normalized length parameter lies half-way between **max-MST** and **min-MAX**?

Characterization of object boundaries: Divergent Concavity[14]

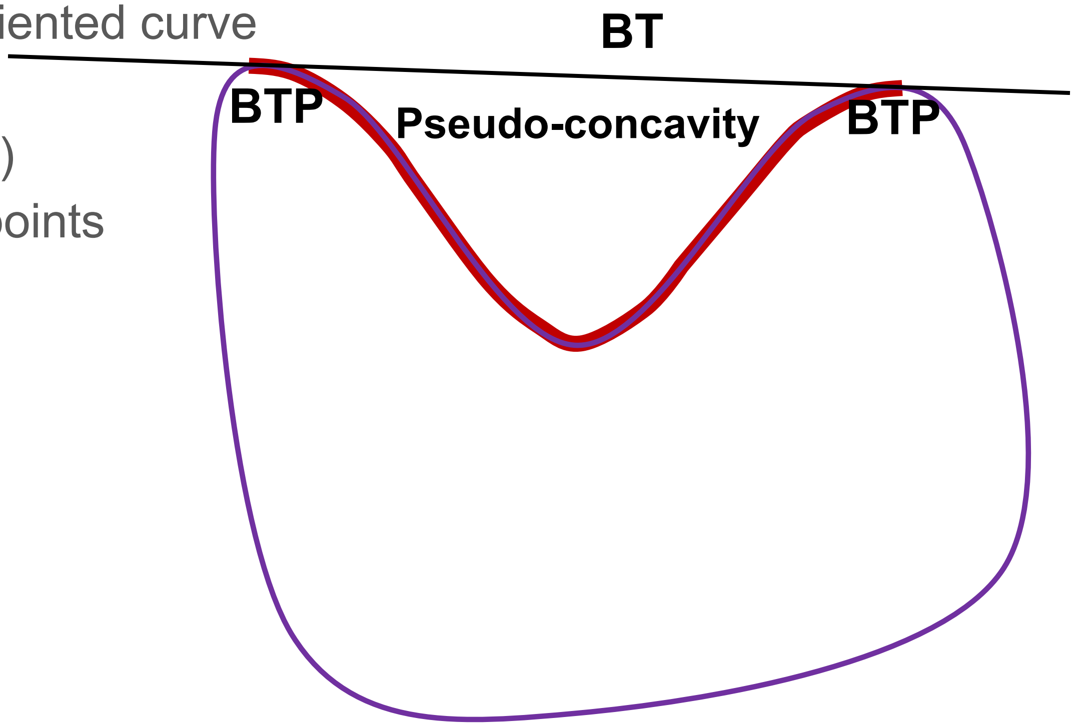
- Closed, planar and positively oriented curve
- Inflection points and curvature
- Concave portion (green colored)
- BT-bi-tangent, BTP-bi-tangent points



[14] Peethambaran J. 2015, “Reconstruction of Water-tight Surfaces through Delaunay Sculpting”, *Computer Aided Design*

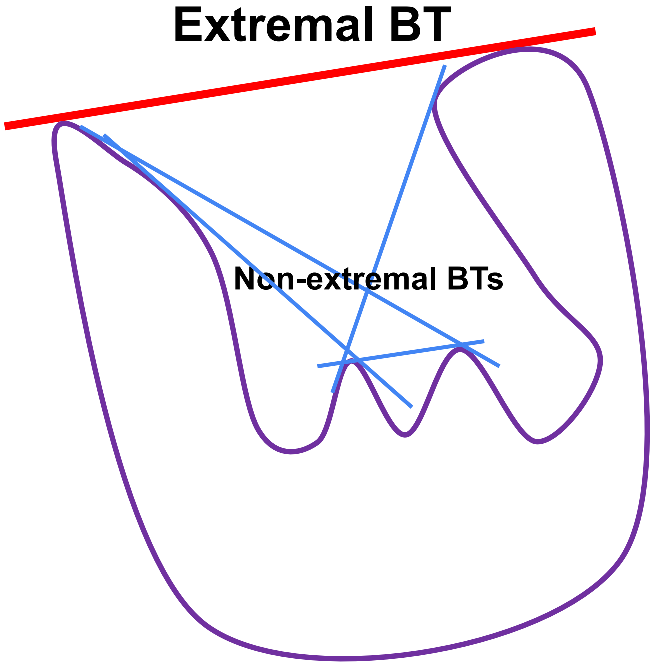
Characterization of object boundaries: Divergent Concavity

- Closed, planar and positively oriented curve
- Inflection points and curvature
- Concave portion (green colored)
- BT-bi-tangent, BTP-bi-tangent points
- Pseudo-concavity

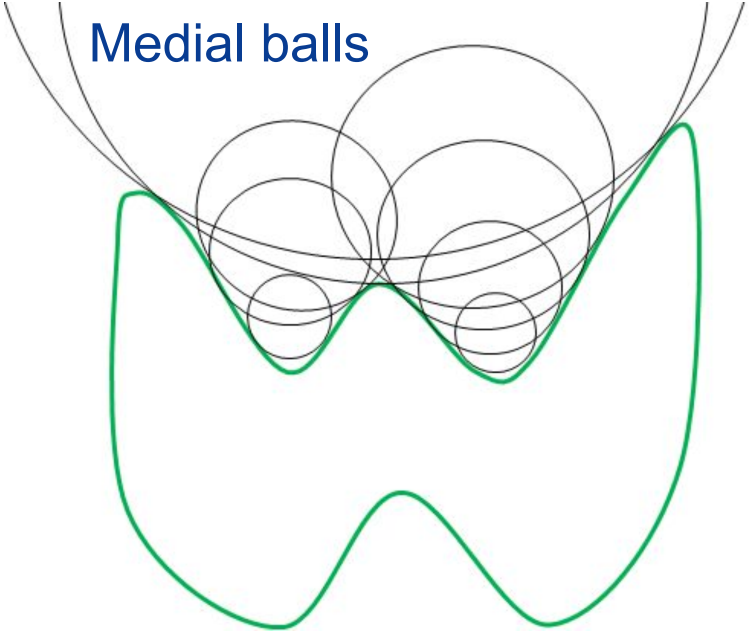


Characterization of object boundaries: Divergent Concavity

- Extremal Vs Non-extremal BT

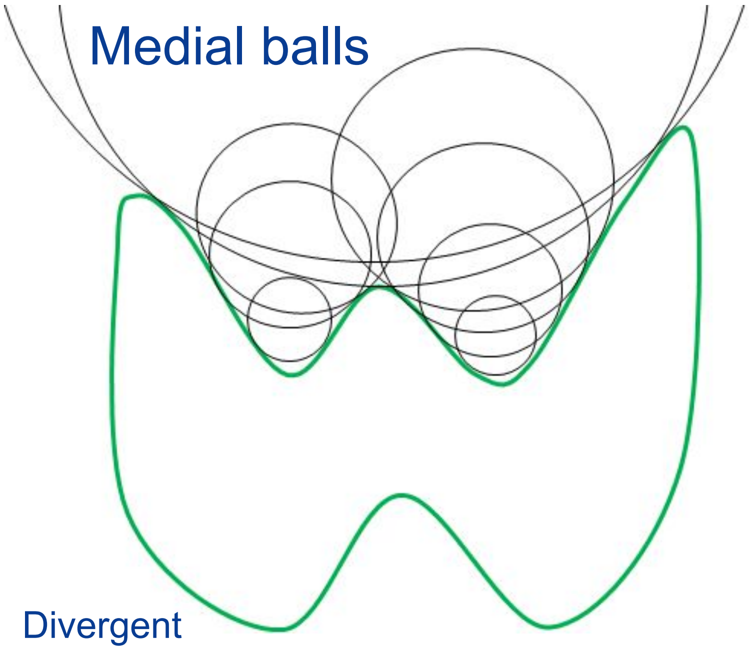


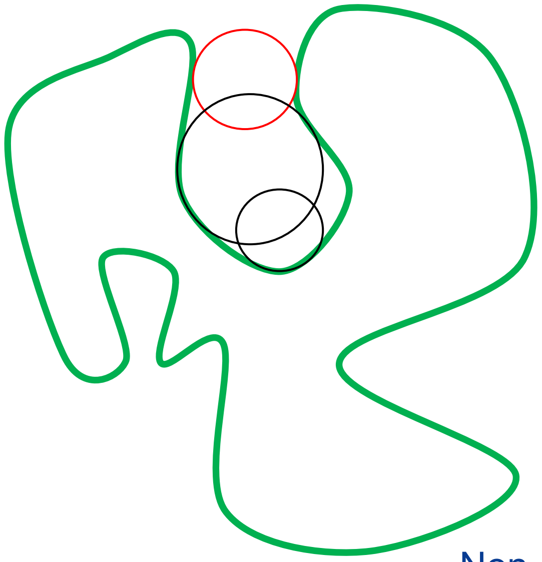
Characterization of object boundaries: Divergent Concavity



- Divergent pseudo-concavity

Characterization of object boundaries: Divergent Concavity



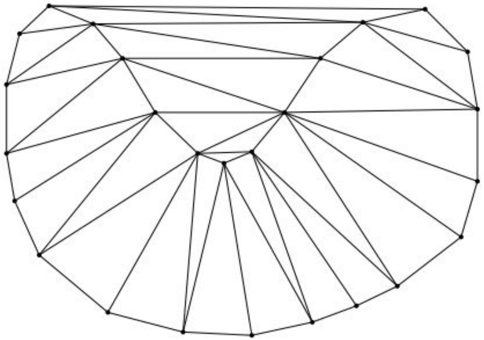


- If all the pseudo-concavities are divergent, then the curve is divergent

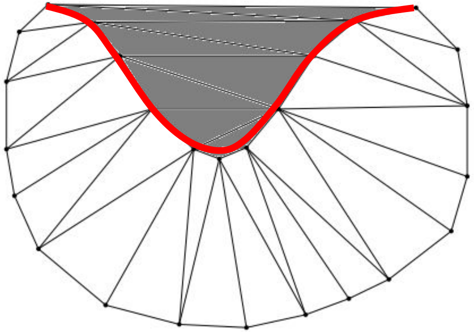
Divergent Concavity

- Implications[19]

Point set, S sampled from a divergent concave curve



DT(S)

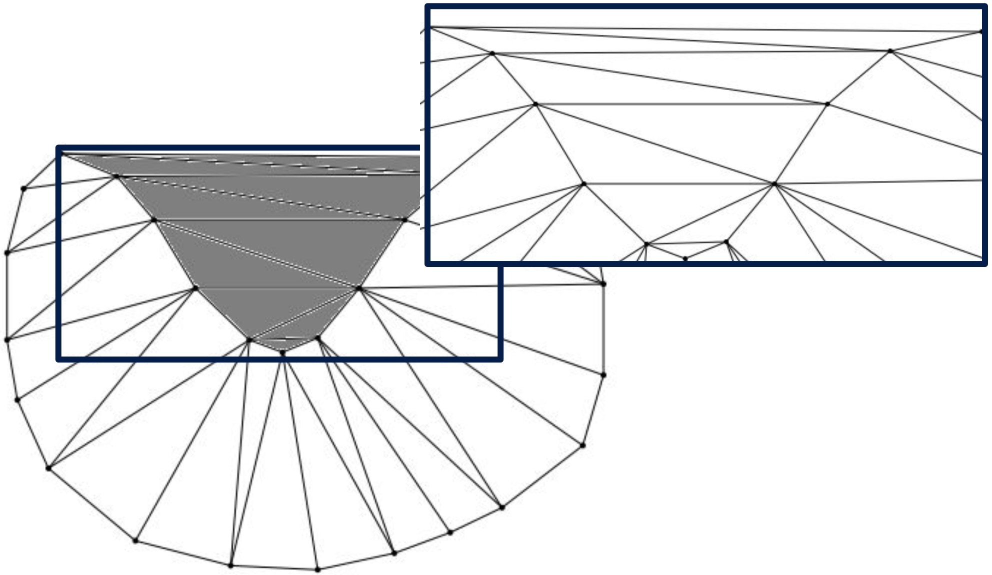


Triangles in divergent concave region

[19] Peethambaran J. 2015, "Non-parametric shape reconstruction and volume constrained Polyhedronization of point sets", *PhD thesis, IIT Madras*

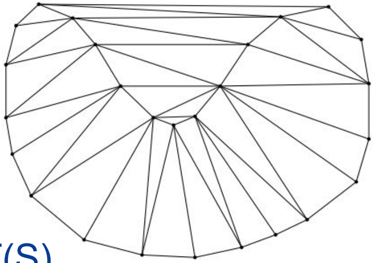
Characterization of object boundaries: Divergent Concavity

- Triangles in divergent concave regions are:
  - Obtuse
  - Longest edge facing towards the extremal BT

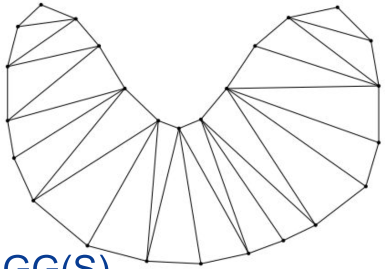


Relaxed Gabriel Graph[15]

- Consists of all Gabriel edges and a few non-Gabriel edges
- RGG(S) retains a non-Gabriel edge (p, q) of DT(S) if it satisfies either of the following:
    - Circumcenter of the Delaunay triangle $\triangle pqr$ for which (p,q) is the characteristic edge, lies internal to $\partial RGG(S)$.
    - Removal of (p,q) violates regularity in RGG(S).



DT(S)

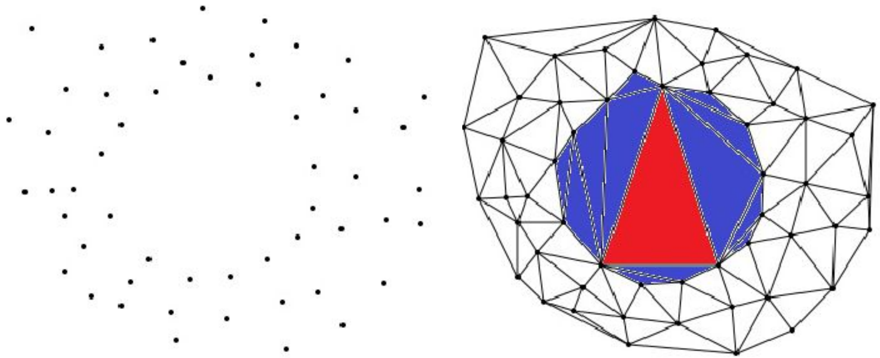


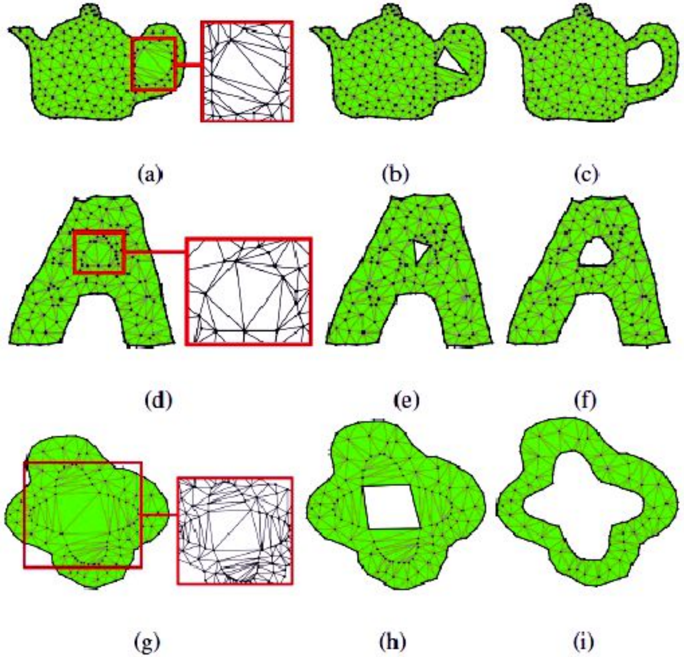
RGG(S)

[15] Peethambaran et al. 2015, “A nonparametric approach to shape reconstruction from planar point sets through Delaunay filtering”, *Computer Aided Design*

Relaxed Gabriel Graph

- Hole structure: fat triangle surrounded by sets of thin triangles



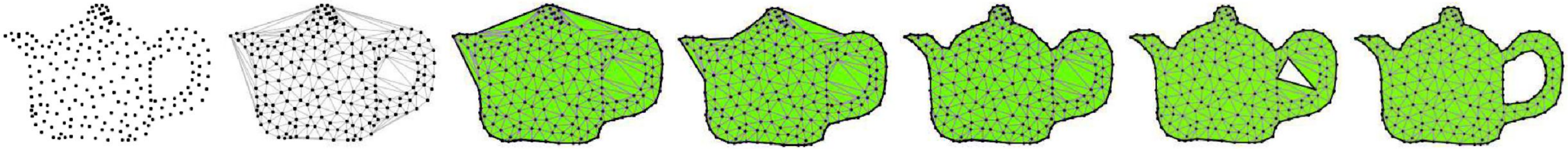


Relaxed Gabriel Graph

- Order the boundary triangles based on their circum-radii (priority queue)
- Remove the boundary triangles if they are deletable

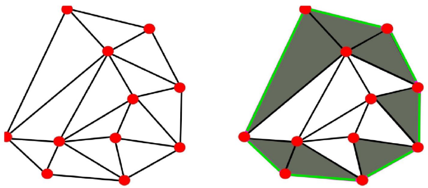
  deletable□ circum-center lie outside the intermediate boundary and the removal does not violate regularity of the simplicial complex.

- O(n log n) complexity

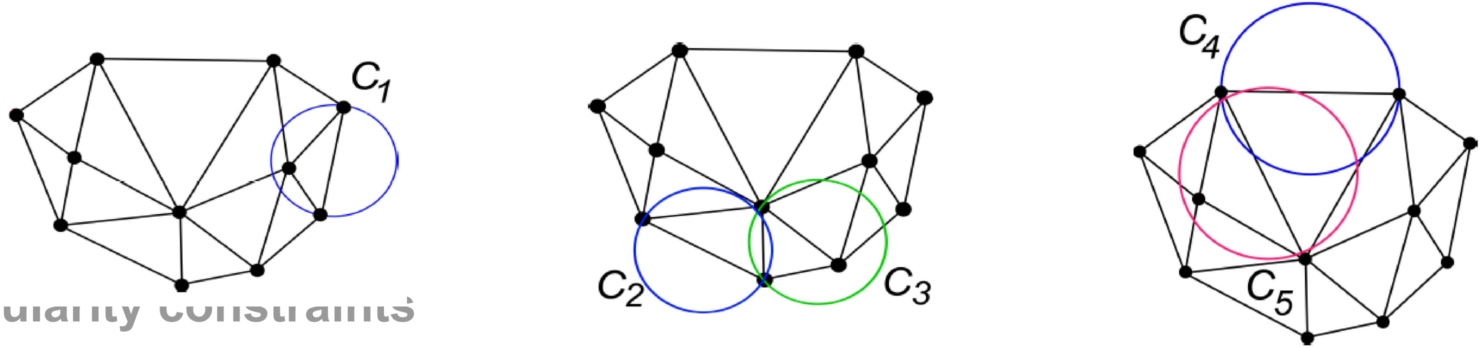


EC-shape[16]

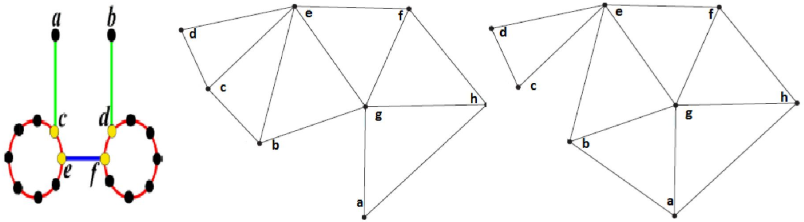
- Exterior triangle and exterior edge



- **Circle constraint**: Non-empty diametric, chord and midpoint circles



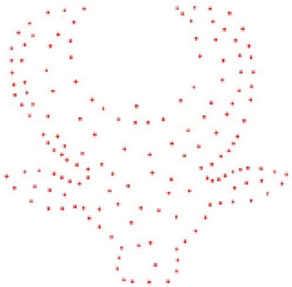
- **Regularity constraints**



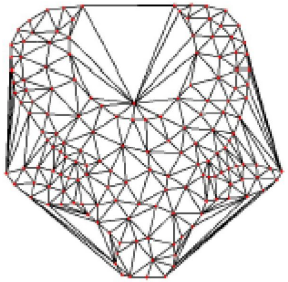
[16] Methirumangalath et al. 2015, "A unified approach towards reconstruction of a planar point set.", *Computers & Graphics*

EC-shape

- Remove the exterior edges if it satisfy circle constraints and regularity constraints
- Illustration:
- Construct Delaunay



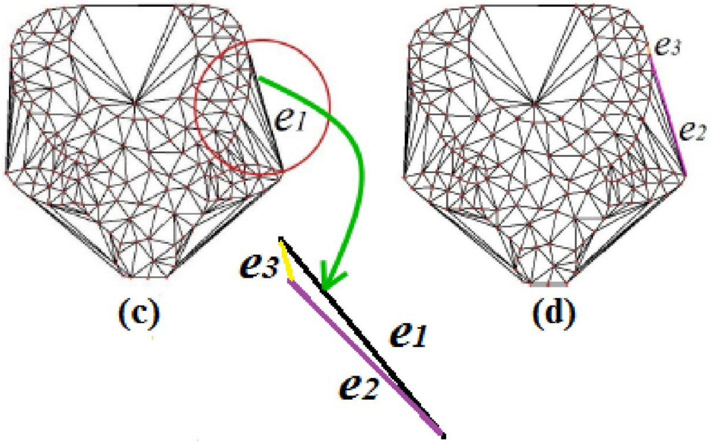
(a)

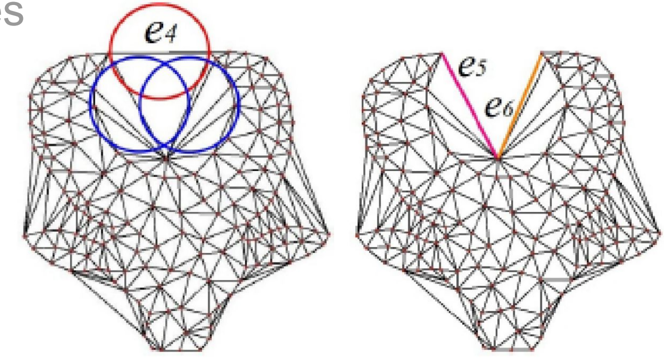


(b)

EC-shape

- Remove the exterior edges if it satisfy circle constraints and regularity constraints
- Illustration:
- Non-empty diametric circle
- Empty diametric circle and non-empty midpoint circles

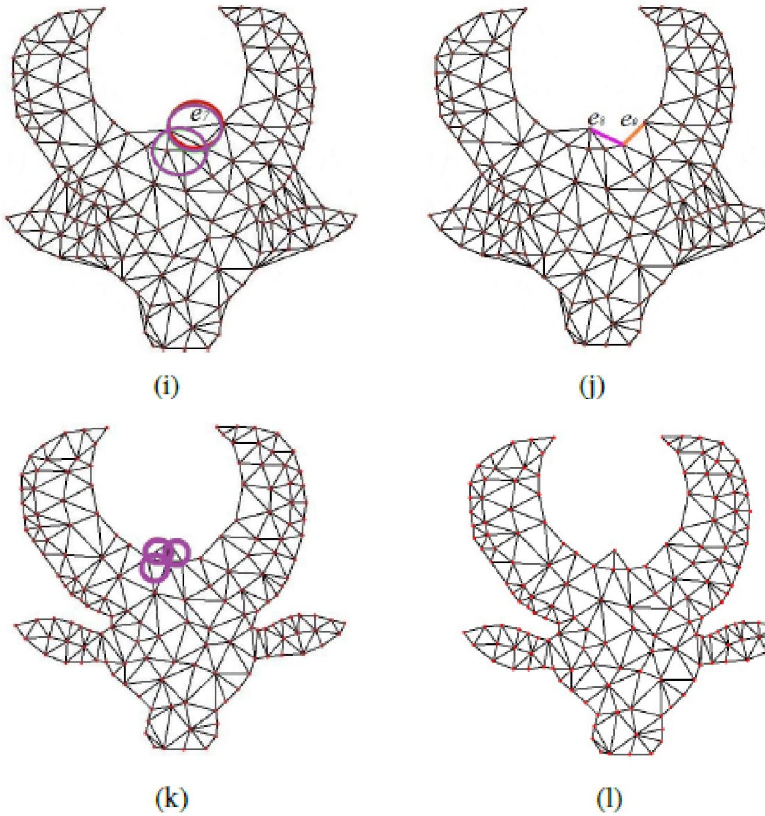




(f) (g)

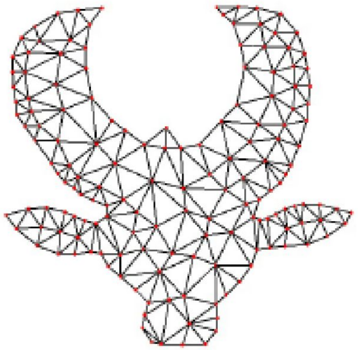
EC-shape

- Remove the exterior edges if it satisfy circle constraints and regularity constraints
- Illustration:
- Empty diametric and non-empty chord circles
- All circles empty

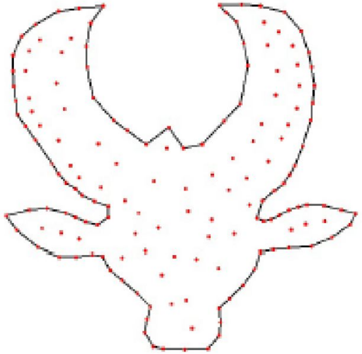


EC-shape

- Final shape
- Under r-sampling, EC-shape is homeomorphic to a simple closed curve



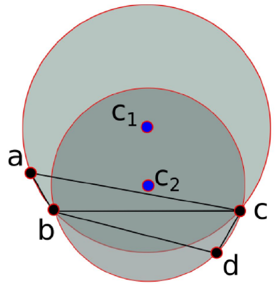
(l)



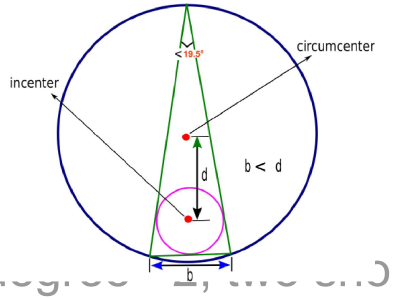
(m)

CT-shape[17]

- **Coordinated triangles**: If the circumcenters of neighboring triangles lie on the same half plane made by the shared edge



- **Skinny triangles**: non-obtuse triangle with ba ... e distance between its circumcenter and incenter

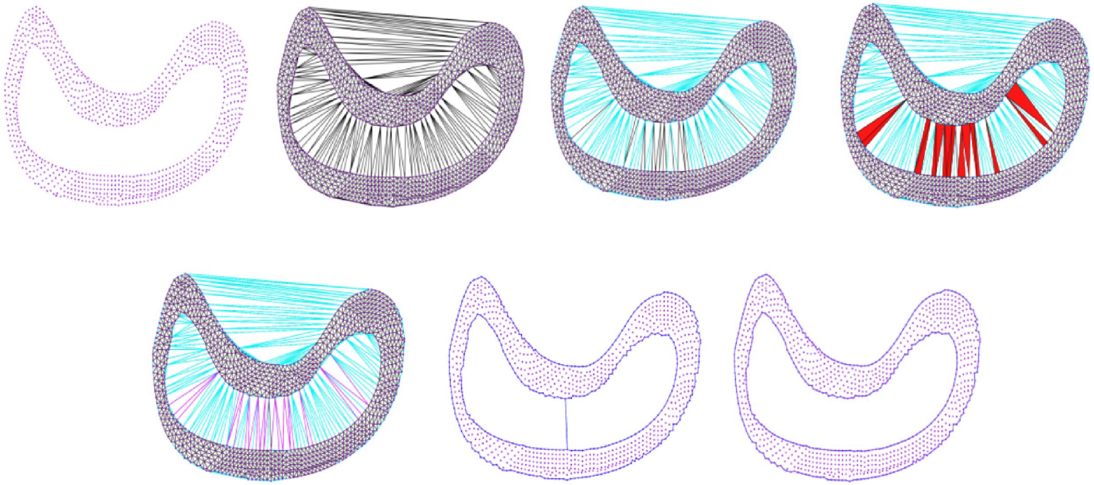


- **Degree constraints**: for vertices with c ... rter edges are retained.

[17] Thayyil et al. 2020, “An input-independent single pass algorithm for reconstruction from dot patterns and boundary samples.”, *Computers Aided Geometric Design*

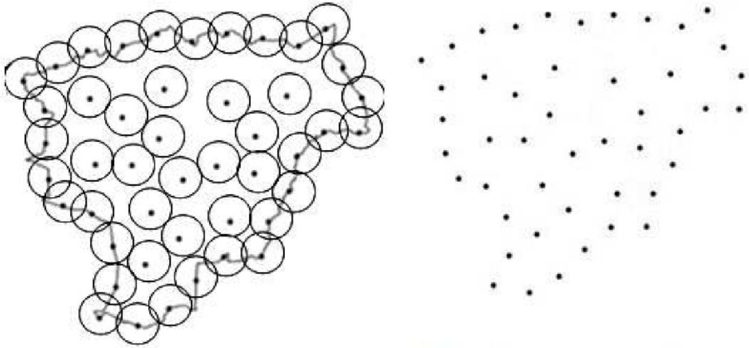
CT-shape

- Mark all the shared edges of the coordinated triangles, two longer edges of the skinny triangles
- Create a graph consisting of all unmarked edges
- Apply degree constraints to get the final shape
- Theoretical guarantees under r-sampling



Sampling Models: r-sampling

- A point set *S* sampled from an object *O* is said to be r-sample if
  - Every pair of adjacent boundary samples p, q lies at a distance of at most 2r.
  - Every pair of samples p, q from the interior of O lies at a minimum distance of 2r.



(a). r-sampling (b). Point samples

Sampling Models: Directed Boundary Sample

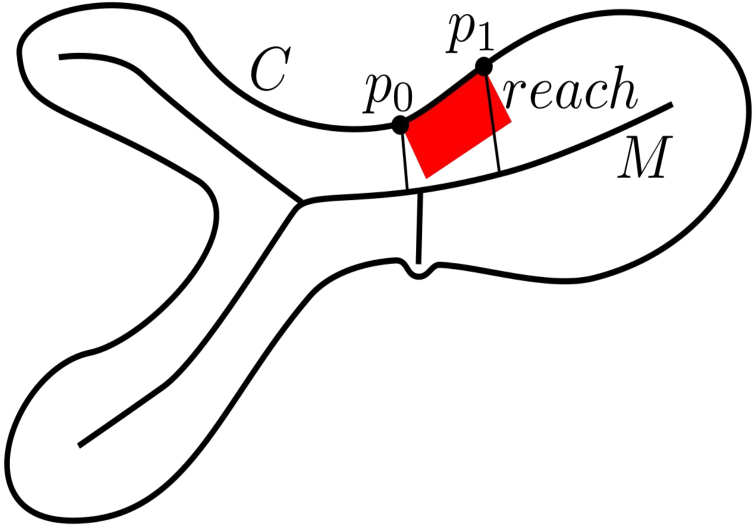
ECCV TEL AVIV 2022

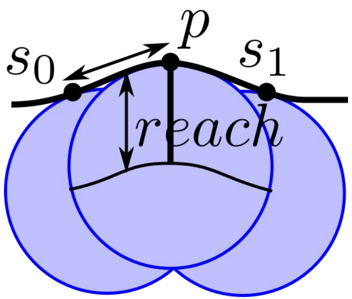
- Directed boundary sample is an r-sampling of object O which possess a divergent boundary
- Theoretical analysis and topological correctness of RGG is provided under directed boundary sample

- Lemma: *Let S be a* (r, ↑)*-sample of an object O,* ∂*RGG*(*S*) *contains an edge between every pair of adjacent samples of* ∂*O.*

Sampling Models: Minimal Reach Sampling

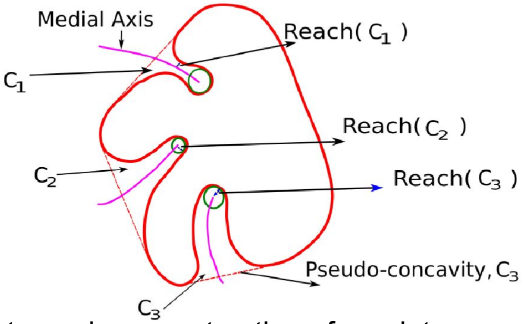
- Interval I(p) = [p_0, p_1] is the set of curve points between p_0, and p_1
- Reach of a curve interval I: inf lfs(p) : p ∈ I

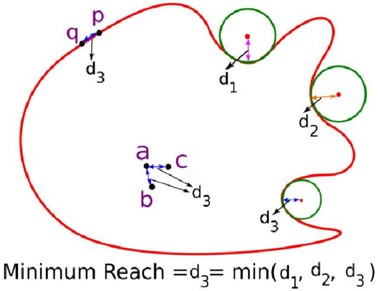




Sampling Models: Minimal Reach Sampling [18]

- Consider pseudo-concavities with extremal bi-tangent points as the intervals
- Local feature size is computed w.r.t exterior medial axis
- Compute the minimum reach of γ all the pseudo-concave intervals
- MRS: the closest neighboring point of any p in S lies at exactly γ





[18] Thayyil et al. 2021, “A sampling type discernment approach towards reconstruction of a point set in R^2”, *Computers Aided Geometric Design*

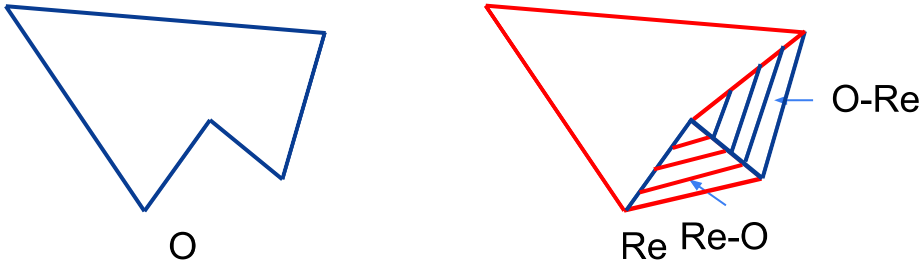
Evaluation Practices: L^2 error norm

- Quantitative analysis based on L^2 error norm[13]

$$L^2 \text{error norm} = \frac{area((O - Re) \cup (Re - O))}{area(O)}$$

O: original object, Re: reconstructed polygon

- L^2 error norm of zero□ both the areas are equal, and the boundaries are structurally alike.



Evaluation Practices: Feature based Comparisons

- Typical features: simple closed curve, multiple components, holes, outliers, sharp corners

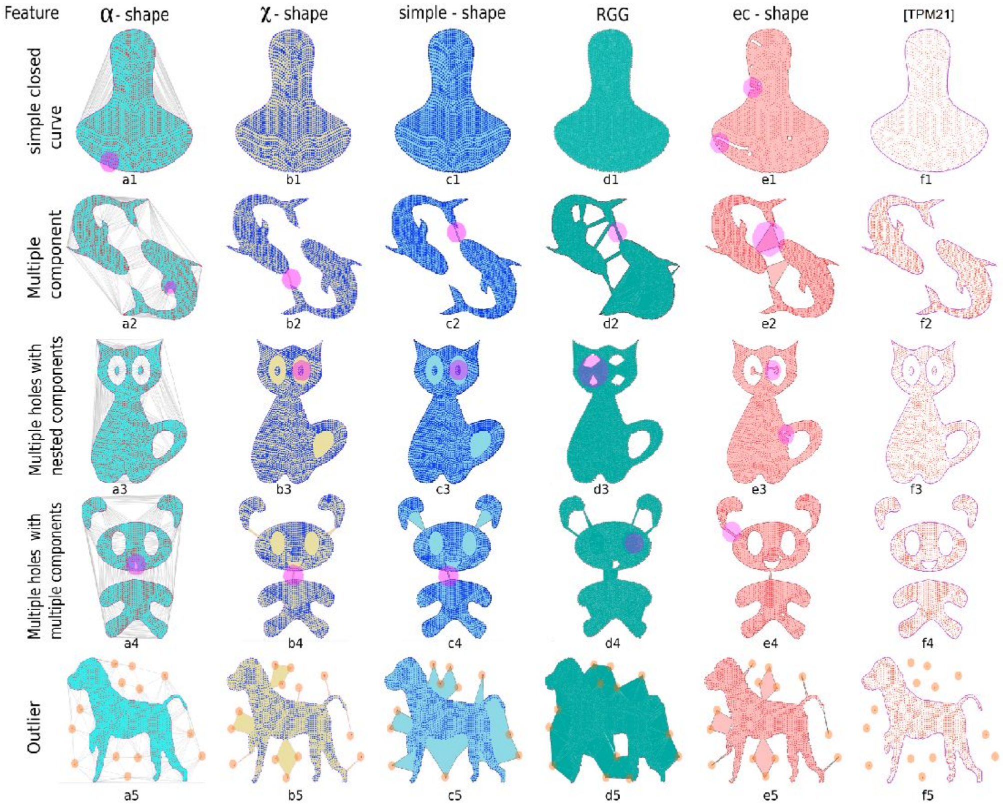


Image Courtesy: [18]

Evaluation Practices: Point Set Density

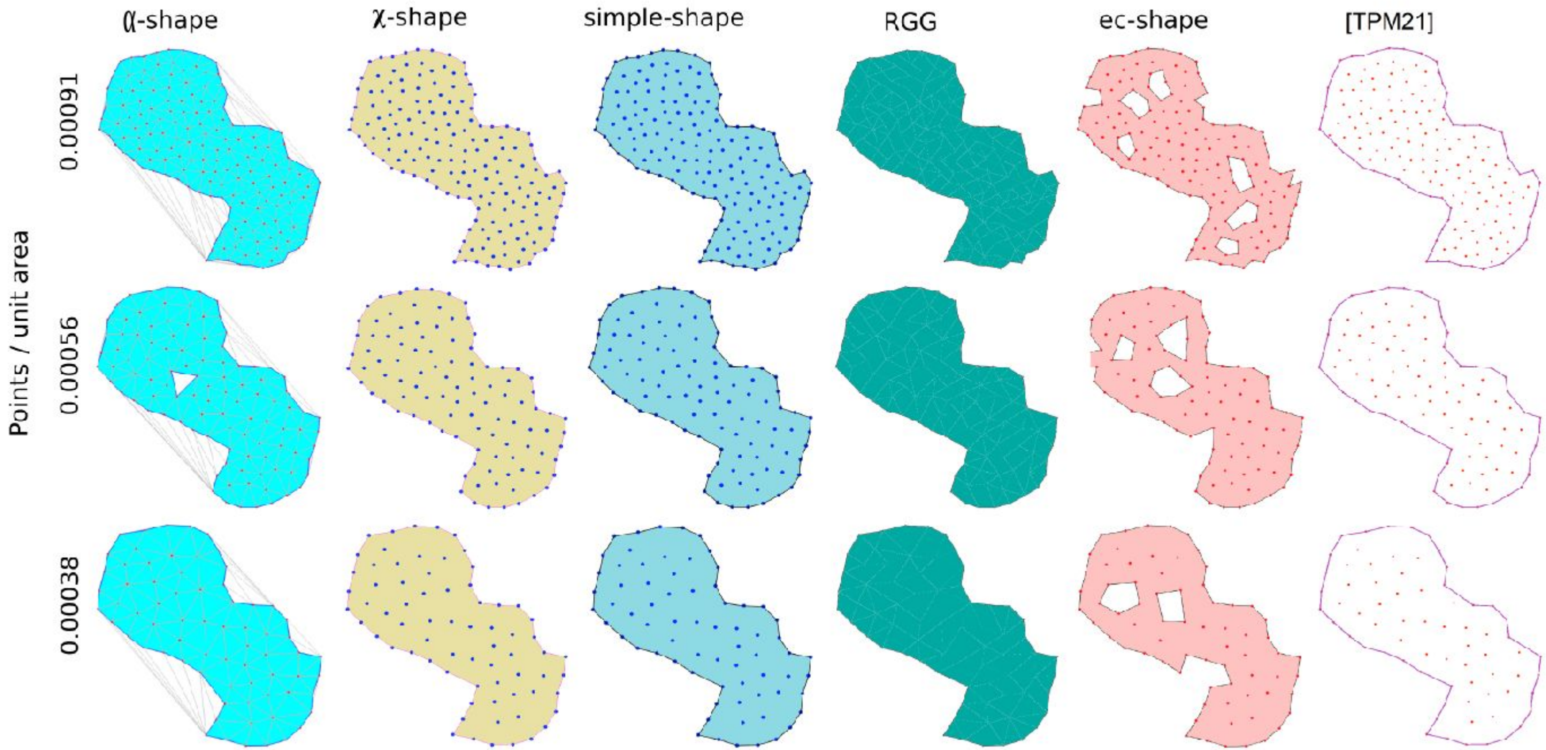


Image Courtesy: [18]

Evaluation Practices: Point Distributions

- DBDI: dense boundary and interior
- DBSI: sparse interior, dense bound
- SBDI: sparse boundary, dense int.
- SBSI: sparse boundary & interior
- Other options: truly random, semi-random etc.
- Not robust to noise/outliers

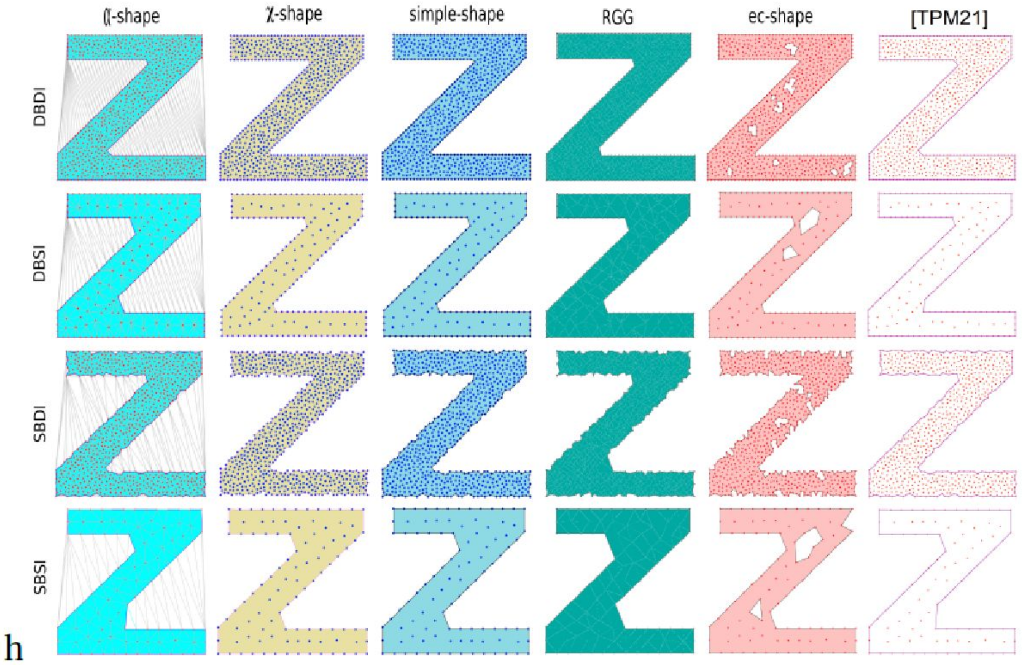
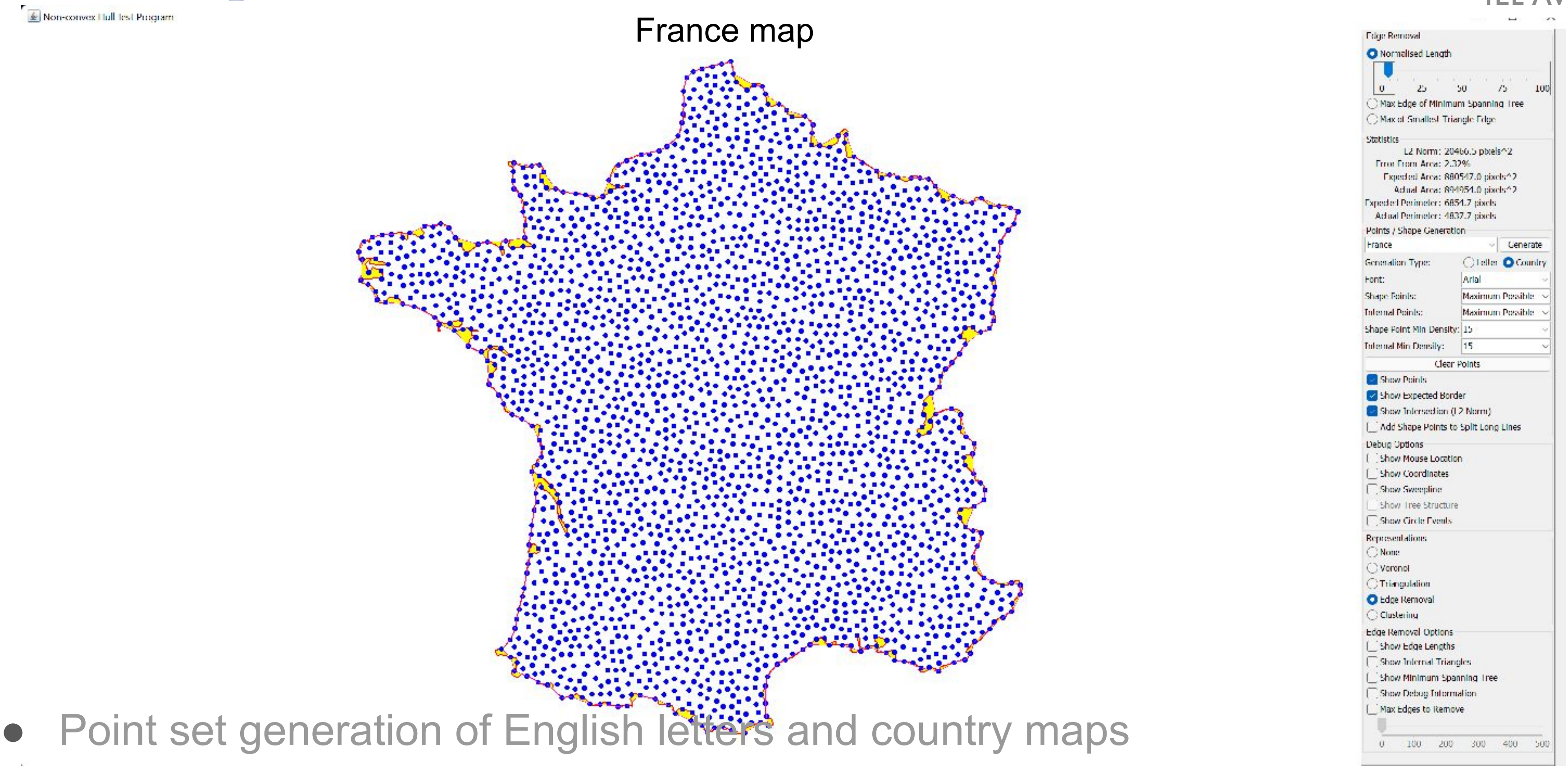


Image Courtesy: [18]

Chi Shape Software [13]



- Point set generation of English letters and country maps

http://duckham.org/matt/characteristics-shapes/

Other Software

- Alpha shape in CGAL Library
- C++ and CGAL predicates

Sl. No	Algorithm	URL
1	CT-shape	https://github.com/agcl-mr/Reconstruction-CTShape
2	Petal ratio	https://github.com/agcl-mr/Reconstruction-Discern
3	Shape-hull graph	https://github.com/jijup/Shapehull2D
4	EC-shape	https://github.com/ShyamsTree/HoleDetection

Future Directions

- Improving and simplifying sampling conditions, especially for non-smooth and self-intersecting curves, and region reconstruction
- Reconstructing curves from hand drawn sketches with varying stroke thickness and intensity
- Deep learning on curves and shapes (similar to 2D medial axis)
- Reconstruct parametric curves instead of piece-wise polygonal curves
- Reconstruction of surfaces from networks of 3D curves
- Kinetic shapes: Shapes of moving points?

Acknowledgements/Collaborations

Collaborators:

- Prof. Ramanathan Muthuganapathy
- Prof. Tamal Dey
- Dr. Stefan Ohrhallinger
- Dr. Amal Dev Parakkat
- Dr. Subhasree Methirumangalath
- Dr. Safeer Babu Thayyil

Acknowledgements:

- Prof. Edelsbrunner
- Prof. Nina Amenta
- Prof. Antony Galton
- Prof. Matt Duckham
- Prof. Mahmoud Melkemi and all other authors of various papers discussed in this tutorial

Thank you!!!

Questions?