Recall

Triangulations of point sets/polygons. Recall what we've seen:

- Delaunay triangulation of point set in R^d, O(n log n) algorithm in R^2.
- O(n) algorithm to triangulate any polygon in R^2 (Chazelle's hard algorithm)

Applications and criteria (this is the outline for the next lectures)

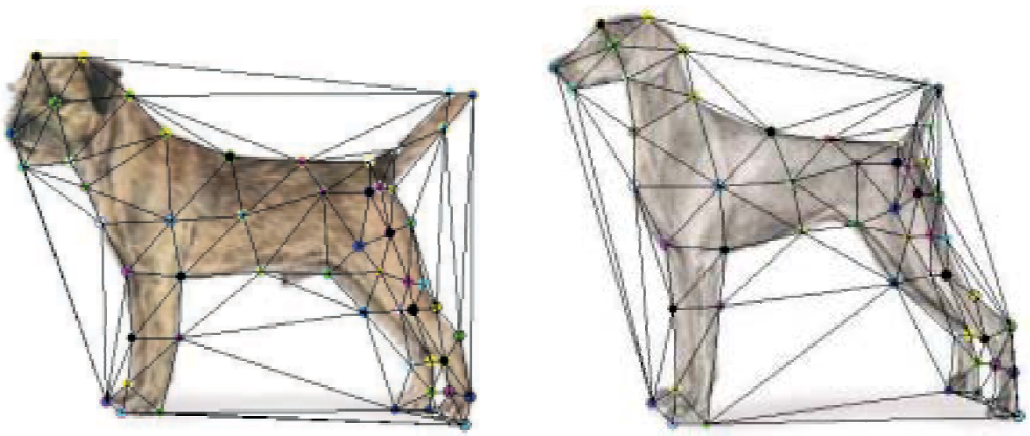
- angle criteria - for meshing
- length criteria: minimum weight triangulation
- constrained triangulations (when certain edge must be included)
- meshing - triangulations with Steiner points
- flip distance
- morphing
- curve and surface reconstruction
- medial axis and straight skeleton

today

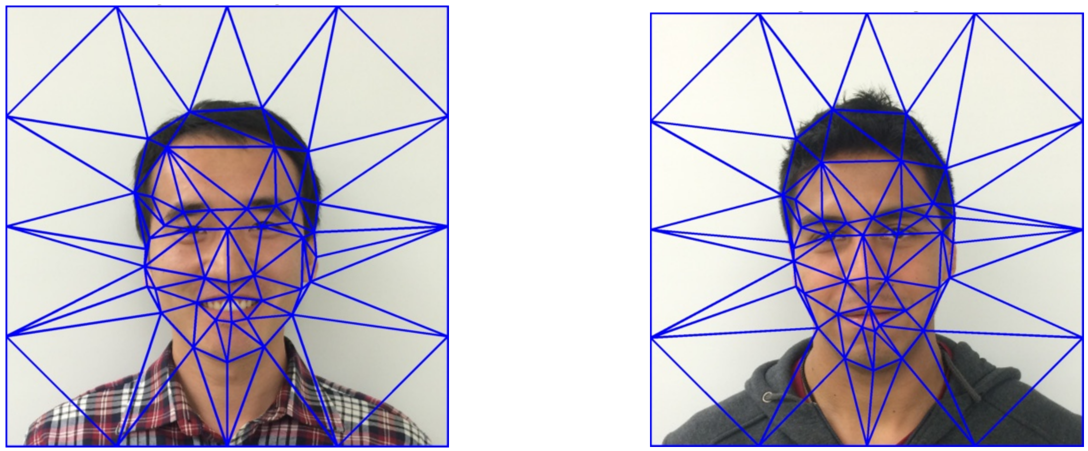
Application of Triangulations: Morphing

500 Years of Female Portraits in Western Art
https://www.youtube.com/watch?v=nUDIoN-_Hxs

Choose corresponding points, and make the "same" triangulation on both.
Then morph the triangles.



Alexei Efros



http://vision.gel.ulaval.ca/~jflalonde/cours/4105/h16/tps/results/tp3/JIZHA16/index.html

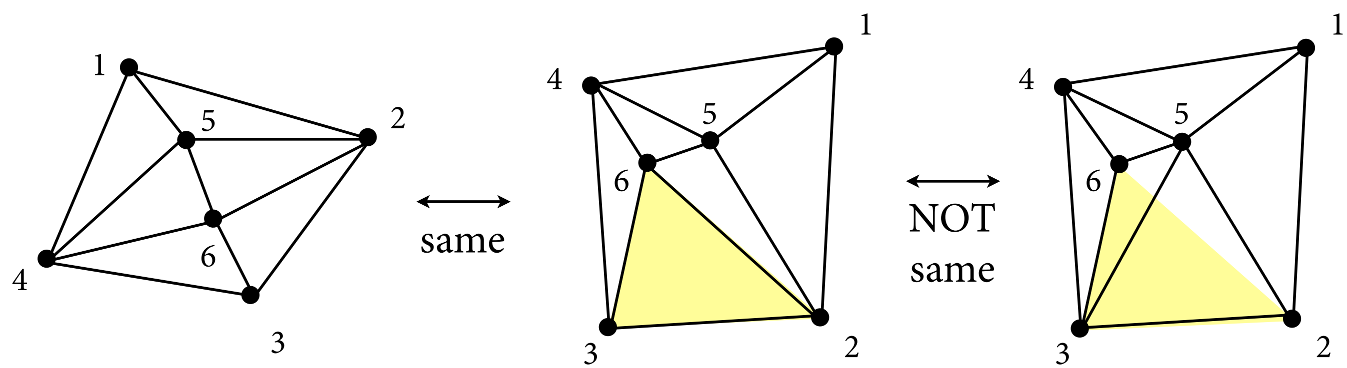
Two aspects to this morphing approach:

1. how to triangulate “compatibly”

2. how to morph compatible triangluations

Compatible triangulations

Given two (unlabelled) point sets, triangulate them the “same” way.



Two triangulations are *compatible* if we can map the points p of the first set to points f(p) of the second set (one-to-one, onto) s.t. pqr is a clockwise triangle iff f(p)f(q)f(r) is a clockwise triangle.

Compatible triangulations

an interesting open side question:
Conjecture: Given two points sets each with n points total, and h points on the convex hull, they have a compatible triangulation.

This assumes no 3 points collinear (otherwise false).

Aichholzer, Oswin, Franz Aurenhammer, Ferran Hurtado, and Hannes Krasser. "Towards compatible triangulations." *Theoretical Computer Science* 296, no. 1 (2003): 3-13.

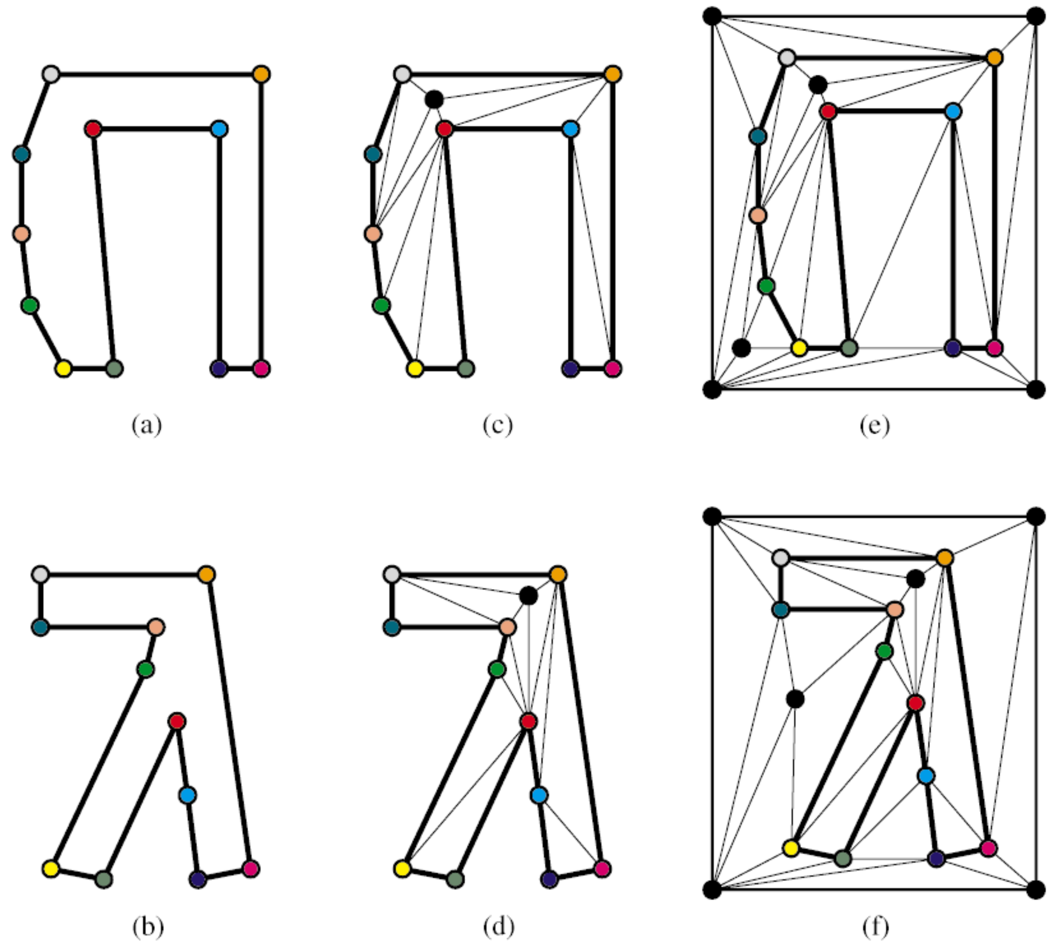
d https://doi.org/10.1016/S0304-3975(02)00428-0

also see Devadoss O'Rourke book

back to what's relevant for morphing:
Theorem. Two simple polygons on n vertices can be compatibly triangulated with Theta(n^2) Steiner points.

Aronov, Boris, Raimund Seidel, and Diane Souvaine. "On compatible triangulations of simple polygons." *Computational Geometry* 3.1 (1993): 27-35.

d https://doi.org/10.1016/0925-7721(93)90028-5 compatible triangulations of polygons

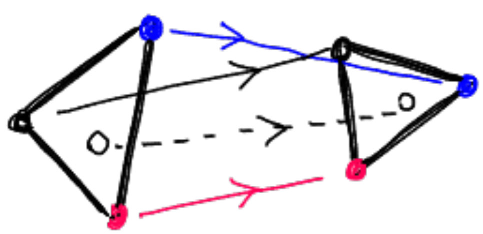


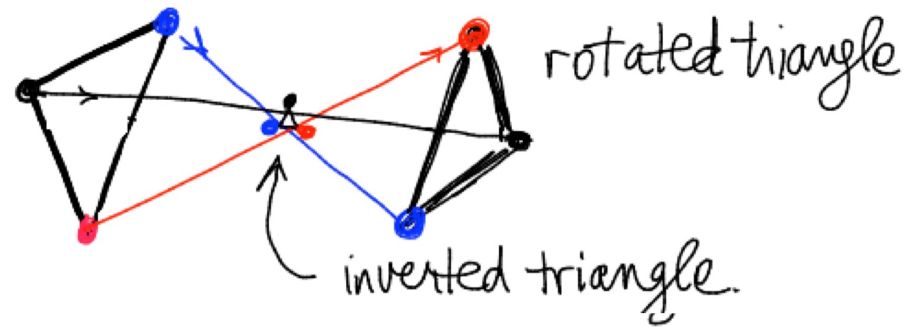
compatible triangulations using 1 Steiner point inside and 1 Steiner point outside

Craig Gotsman, Vitaly Surazhsky

Morphing compatible triangulations

The face morphing projects just use a linear mapping of each triangle.





such morphs do not preserve planarity in general

Planarity preserving morphs

- existence first proved by Cairns, 1944

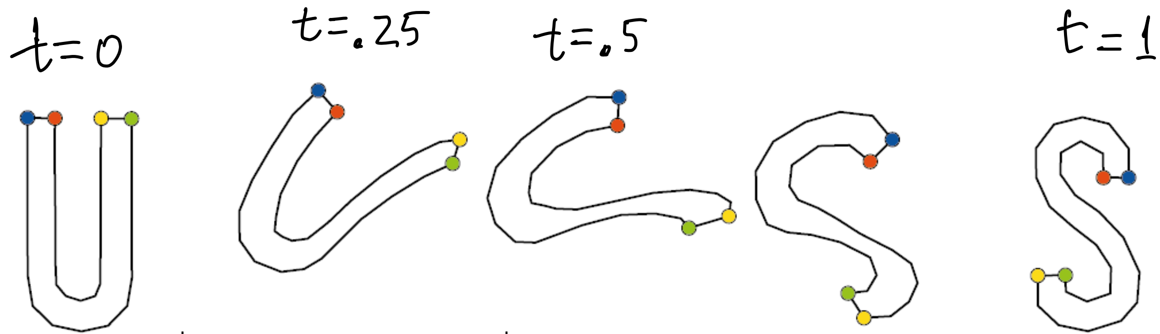
- solution by Floater, Gotsman, Surazhky 2000, using Tutte's graph drawing algorithm. No explicit vertex trajectories.

- piecewise linear soluton

Alamdari, S., Angelini, P., Barrera-Cruz, F., Chan, T.M., Da Lozzo, G., Di Battista, G., Frati, F., Haxell, P., Lubiw, A., Patrignani, M., Roselli, V., Singla, S., Wilkinson, B., 2017. How to morph planar graph drawings. SIAM J. Comput.

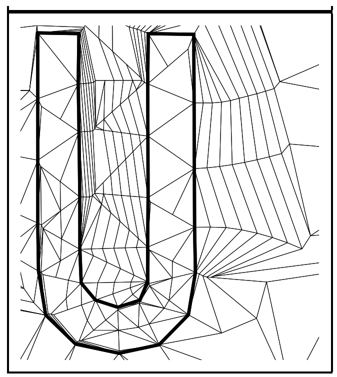
d https://doi.org/10.1137/16M1069171

morphing using Floater, Gotsman, Surazhky method



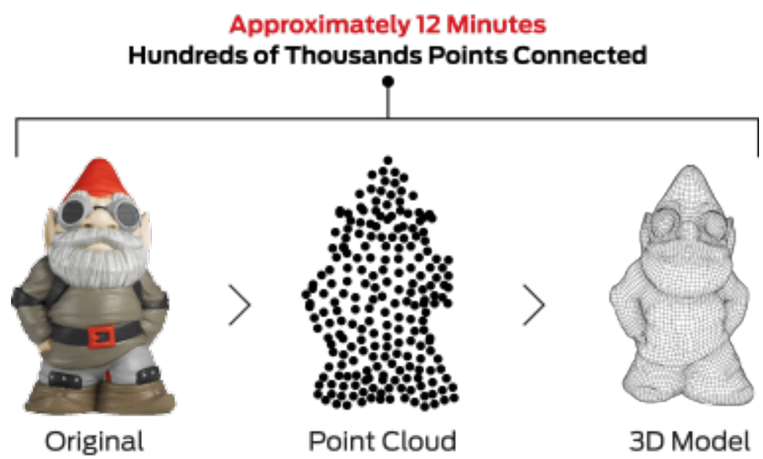
Their algorithm takes a time $t \in [0,1]$ and produces the polygon at time t.

They use many values of t to make it look continuous.

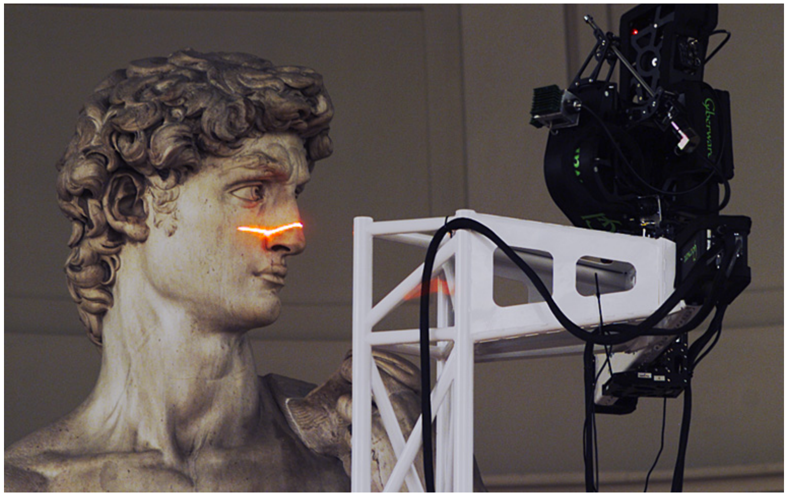


Craig Gotsman, Vitaly Surazhky

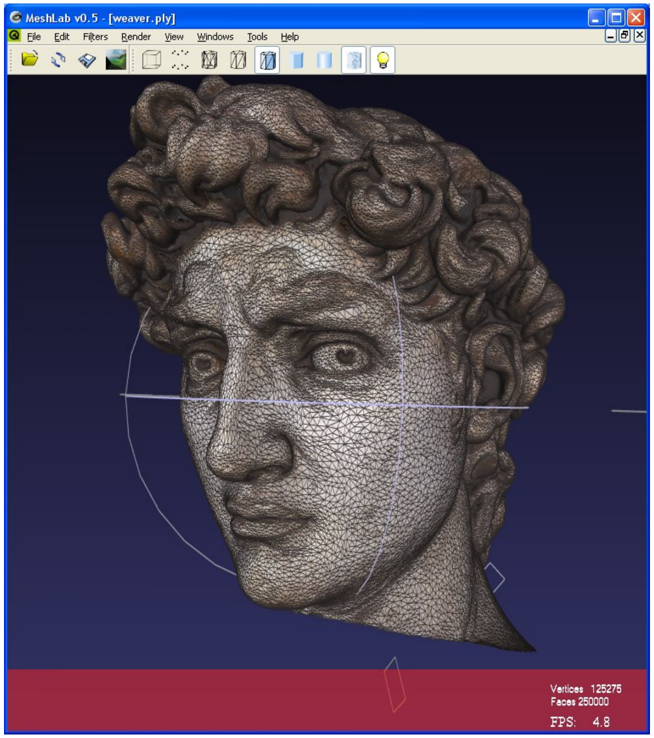
Curve and surface reconstruction

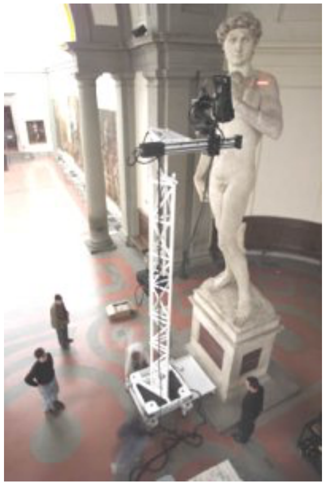


Curve and surface reconstruction



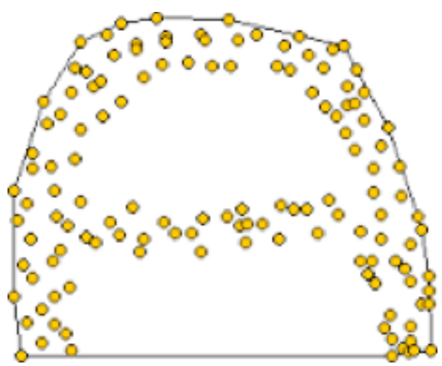
digital Michaelangelo project



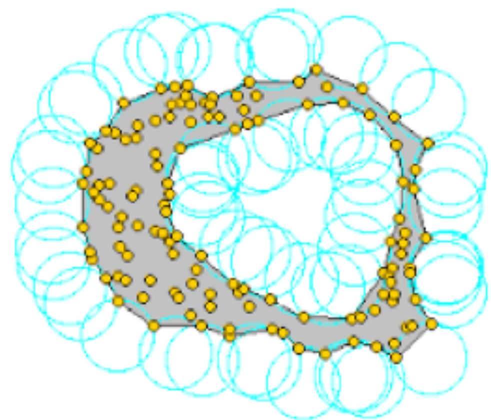


Curve and surface reconstruction

alpha-shapes and alpha-hulls

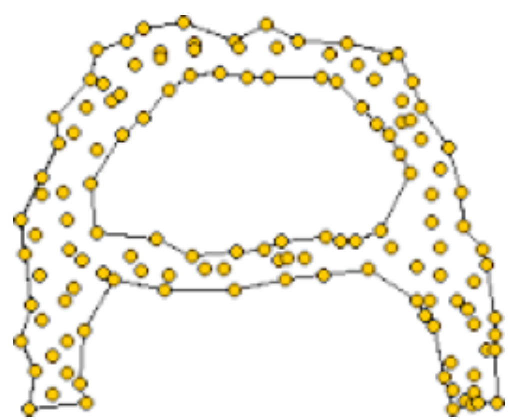


pushing lines against a point set
gives the convex hull
line = infinite radius circle



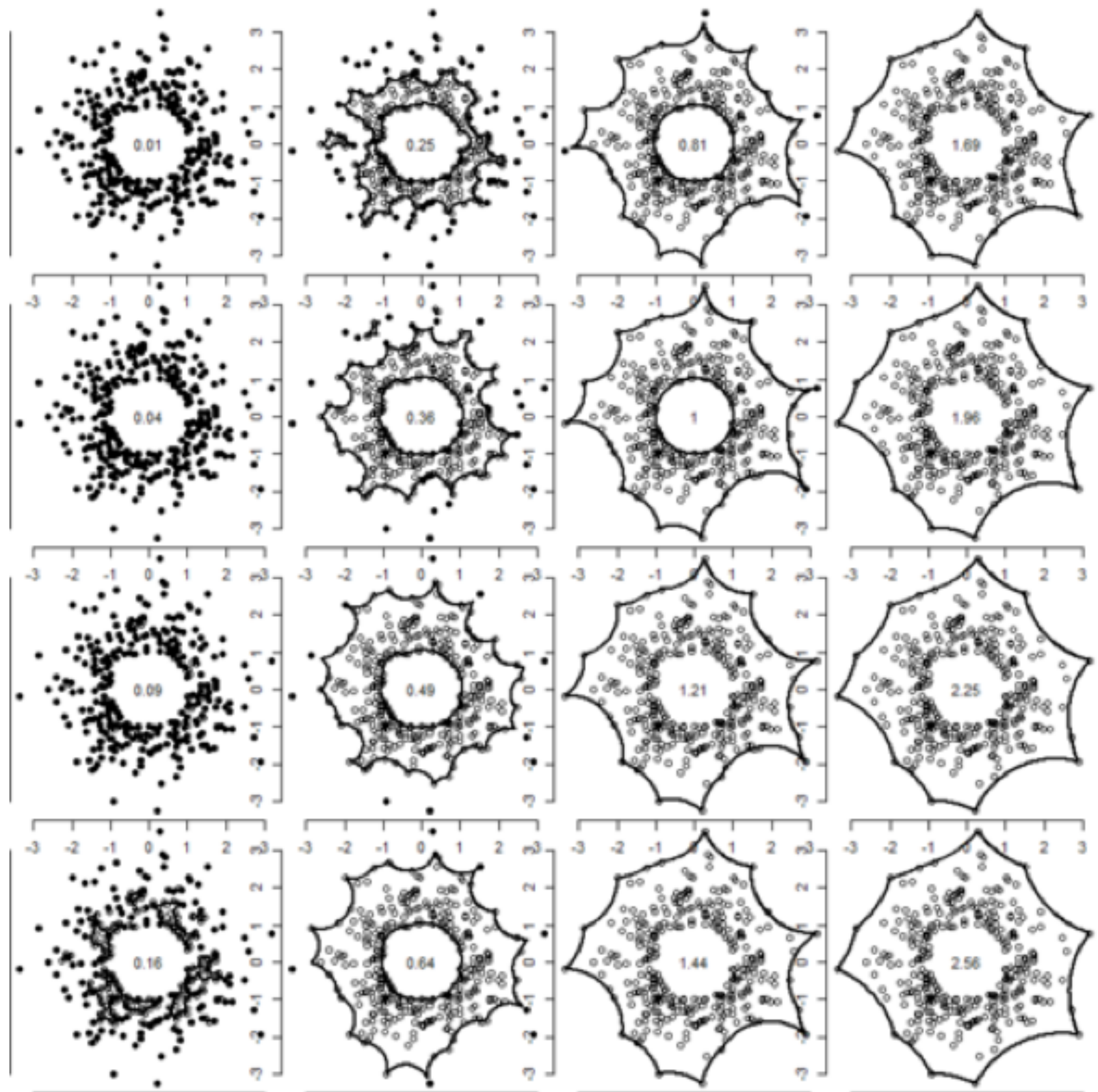
pushing discs of smaller radius
gives more refined “shape”
and detects holes

the alpha-hull,
alpha = disc radius

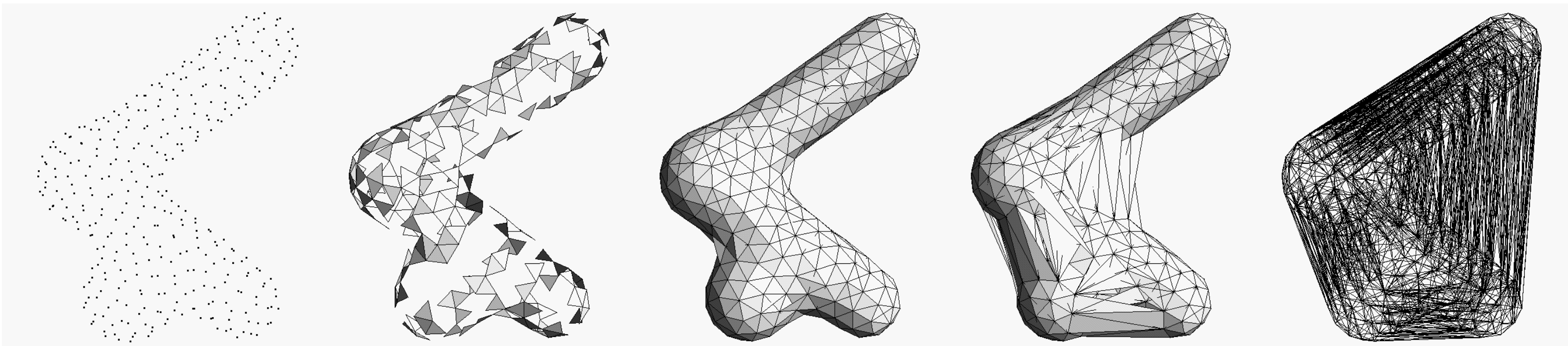


alpha-shapes and alpha-hulls

when alpha is small, the points remain isolated; when alpha is large the alpha-hull approaches the convex hull



alpha-shapes and alpha-hulls



Teichmann, Capps

issues:

- what is the “right” value of alpha?
- if points are not uniform then no single value of alpha will work.

Edelsbrunner, Herbert, and Ernst P. Mücke. "Three-dimensional alpha shapes." *ACM Transactions on Graphics (TOG)* 13.1 (1994): 43-72.
cited by 1939

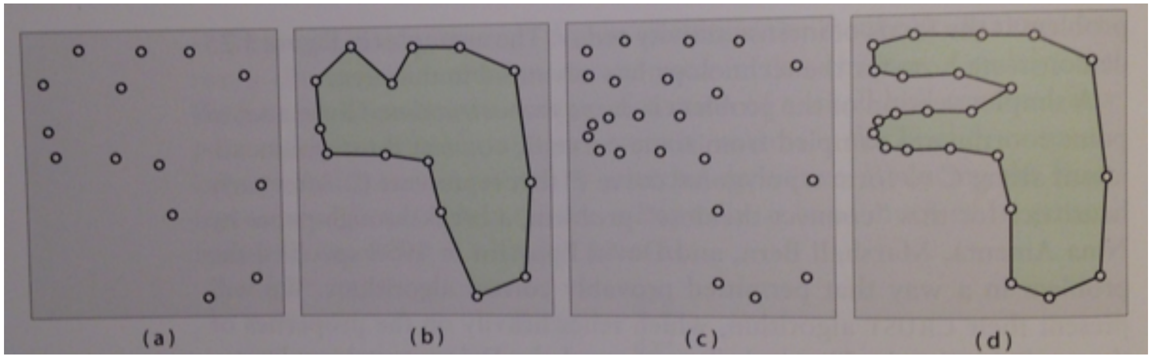
d https://doi.org/10.1145/174462.156635

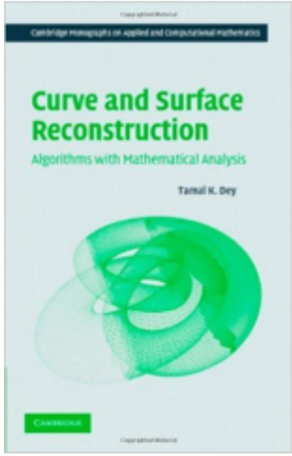
Crust Algorithm for surface reconstruction

in 2D this is curve reconstruction

figures from Devadoss, O'Rourke

points on the curve must be sufficiently dense in order to reconstruct the curve





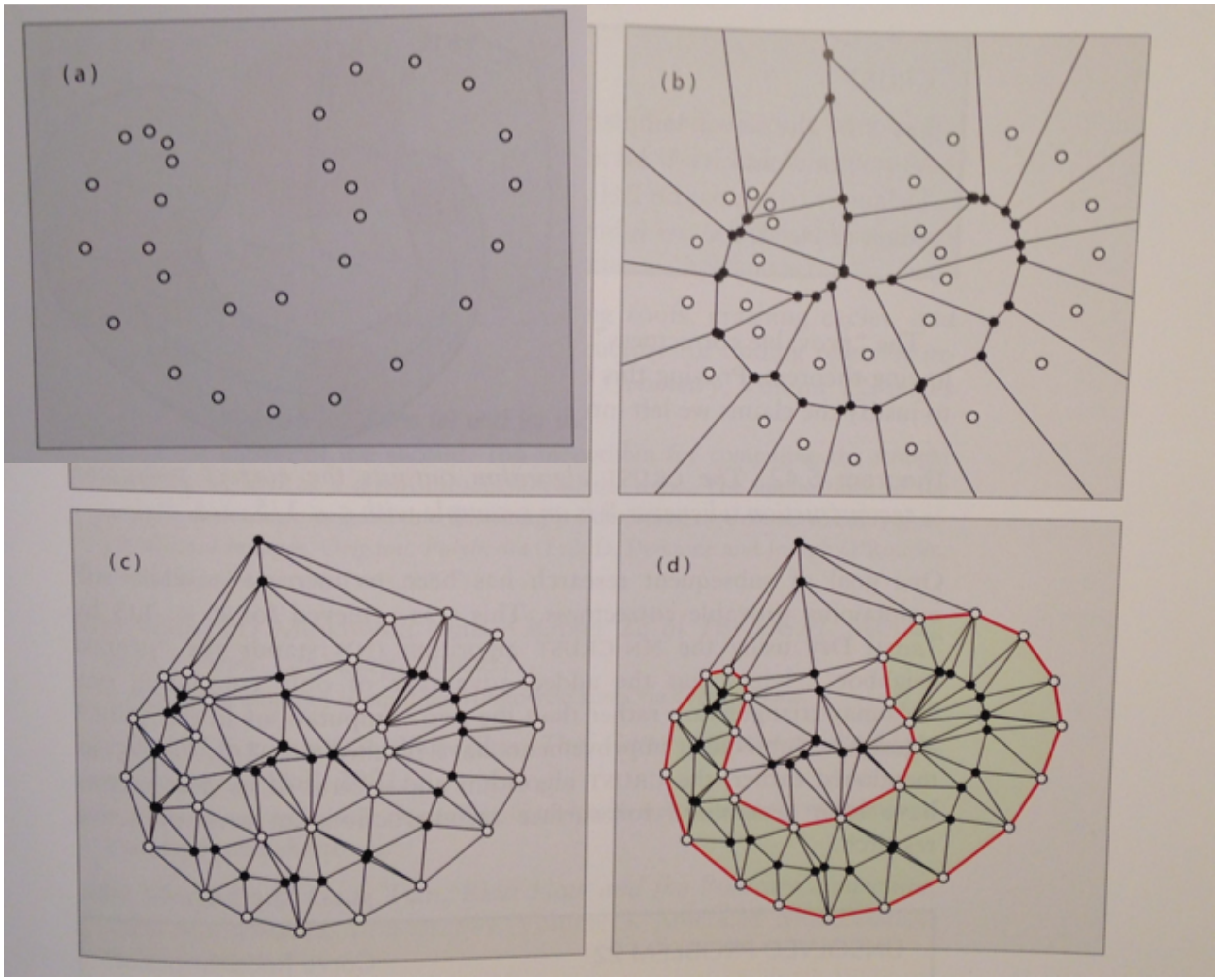
Dey, Tamal K. *Curve and surface reconstruction: algorithms with mathematical analysis*. Vol. 23. Cambridge University Press, 2006.

Amenta, Nina, Marshall Bern, and David Eppstein. "The crust and the β-skeleton: Combinatorial curve reconstruction." *Graphical models and image processing* 60.2 (1998): 125-135.

d https://doi.org/10.1006/gmip.1998.0465

input points

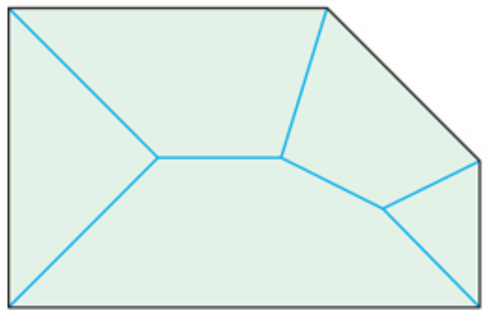
Voronoi diagram



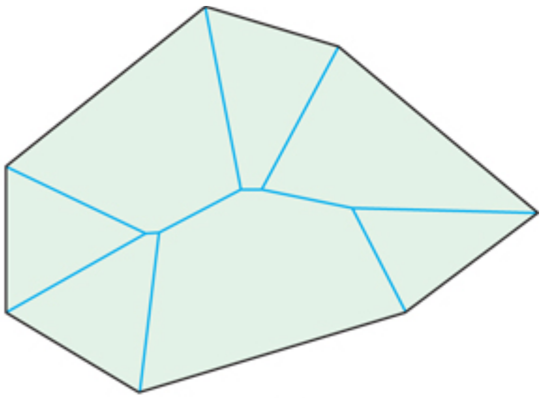
Delaunay triangulation of original points S + Voronoi vertices

edges with both endpoints in S

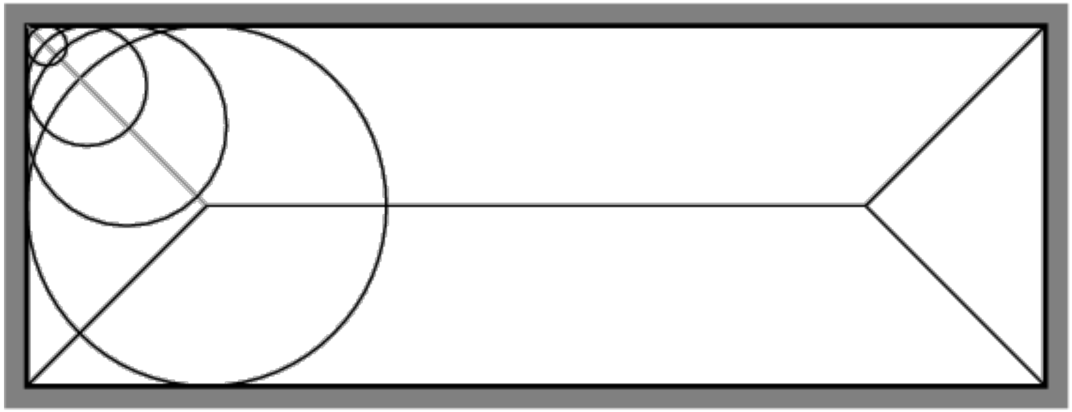
Medial axis of a convex polygon = Voronoi diagram of edges of polygon



(a)



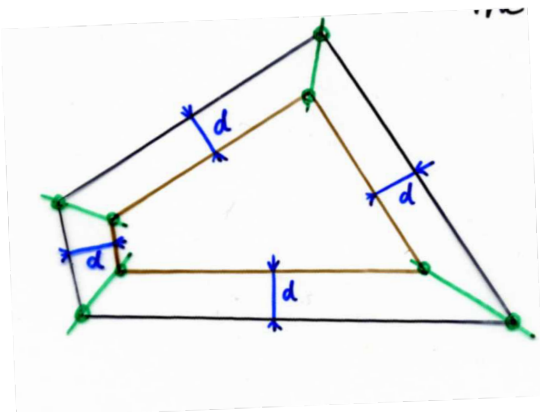
(b)

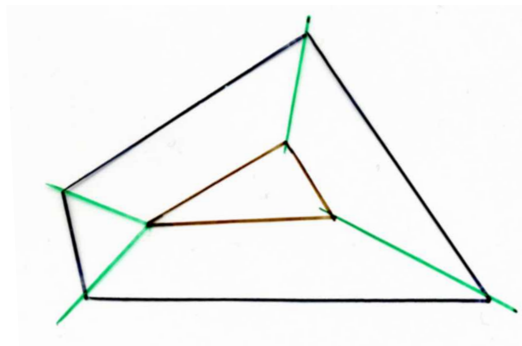


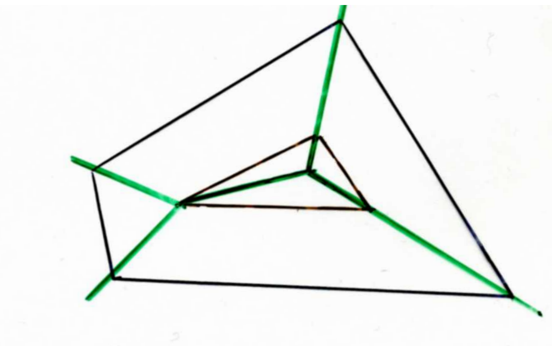
= locus of centers of circles inside polygon that touch boundary at 2 or more points
(centers of maximal inscribed discs)

Medial axis of a convex polygon = Voronoi diagram of edges of polygon

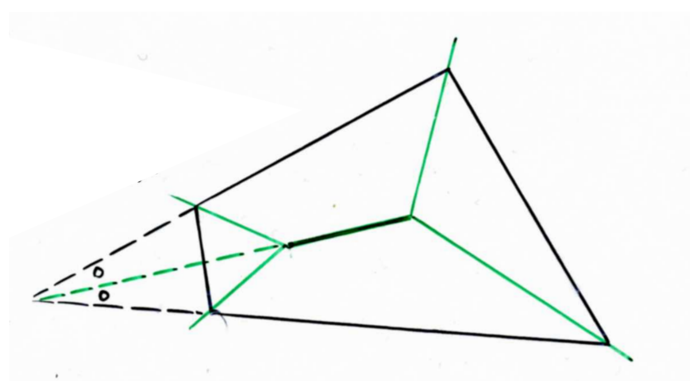
= grow the vertex angle bisectors by shrinking the polygon. The trajectories of the vertices form the medial axis.





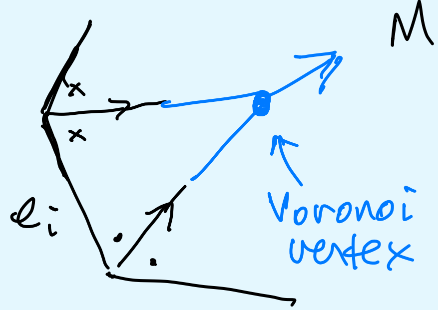


every edge of the medial axis is a bisector of two polygon edges



Medial axis of a convex polygon = Voronoi diagram of edges of polygon

There is an O(n) time algorithm.
Here is a simpler O(n log n) algorithm:



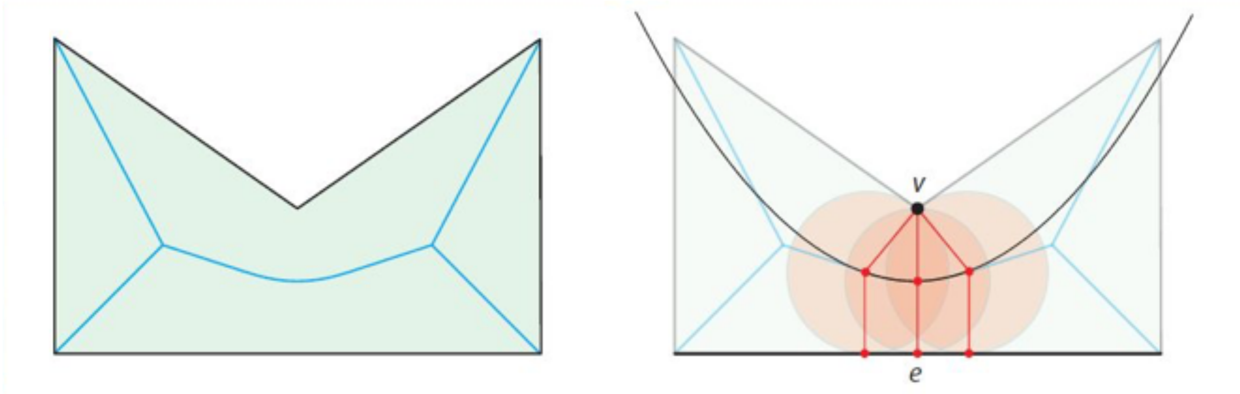
Maintain

- for each pair of consecutive edges their angle bisector ray
- for for pair of consecutive rays, their meeting point and time t when they meet
- a priority queue of these times.

Update:

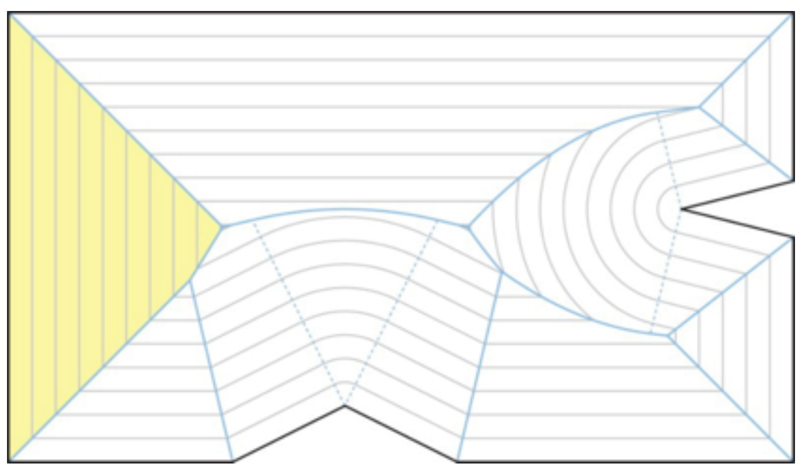
- find & delete min time, output Voronoi vertex
- delete e_i and 2 rays.
- find new ray bisector of e_{i-1} and e_{i+1}
- & compute its meeting point with its two neighbour rays.

Medial axis of a non-convex polygon = locus of centers of maximal inscribed discs



Joseph O'Rourke

Figure 5.6: The central arc lies on the parabola determined by the vertex v and the edge e, where the maximal disks centered on that arc touch e and v.



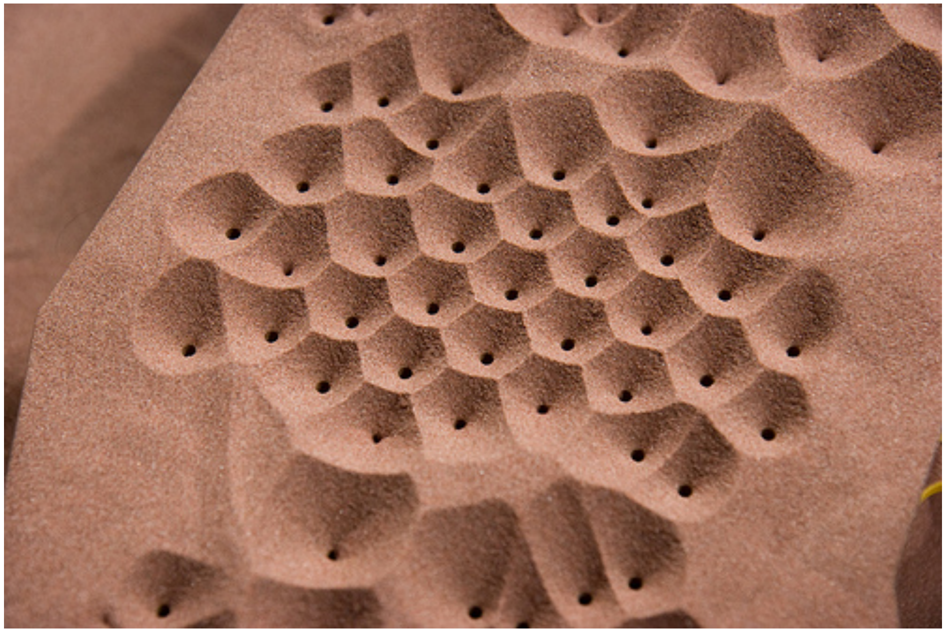
can be found in time O(n)

A physical model for medial axis

- Imagine the polygon is drawn on the prairie, and you light fires along the boundary. Medial axis = points where fire is quenched (fire meets other fire)

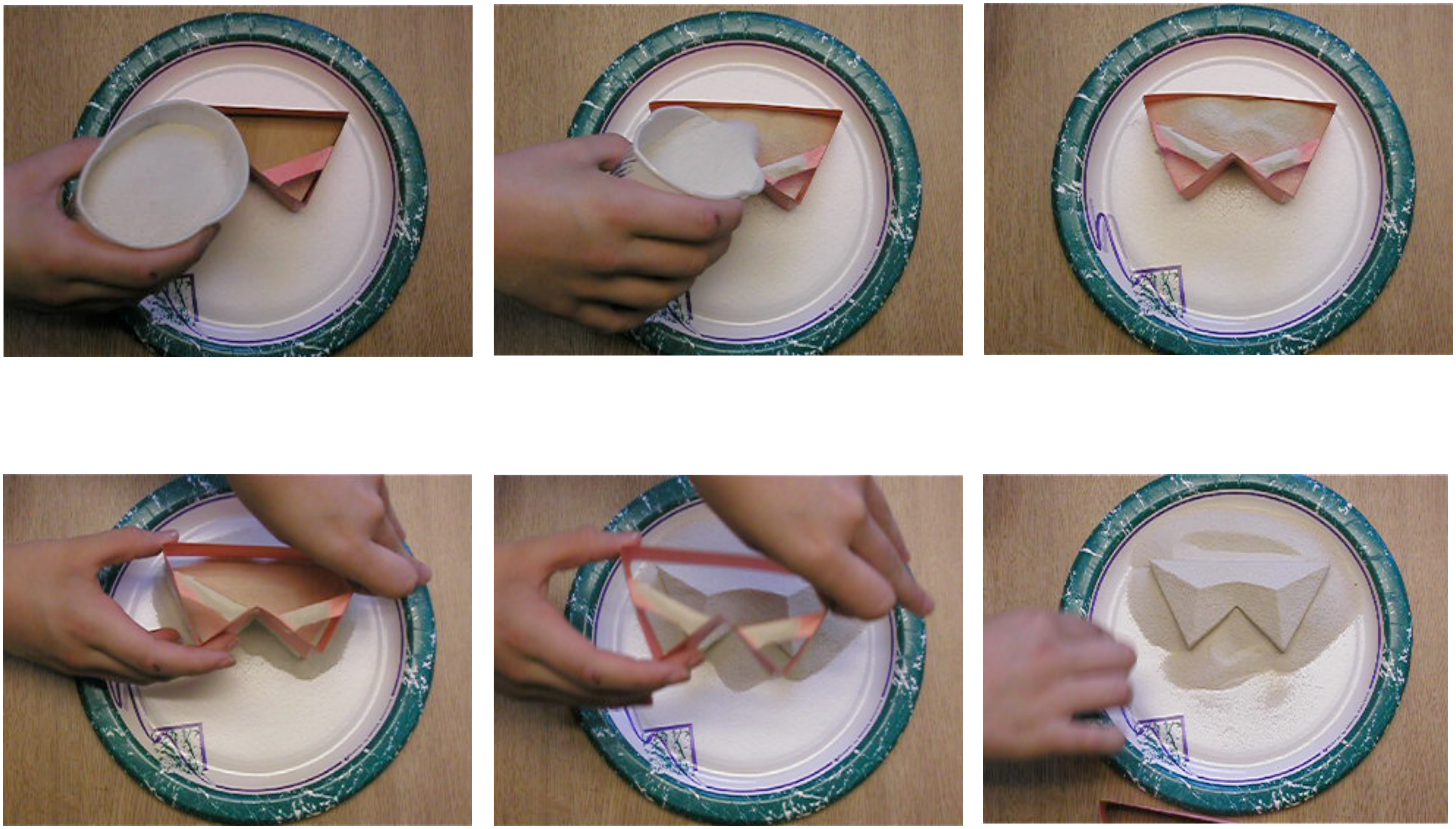
- pouring sand

Voronoi diagram



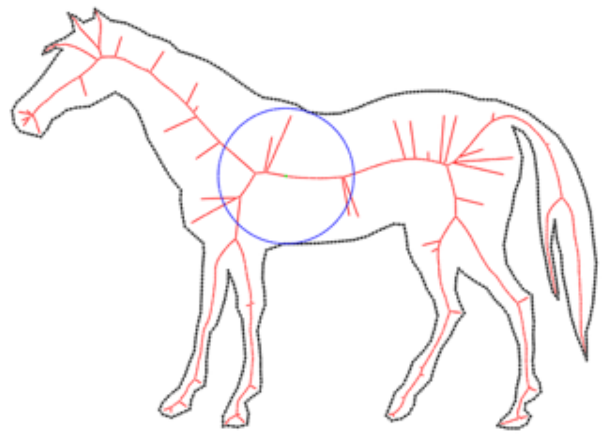
bradmohr

A physical model for medial axis



Applications of medial axis

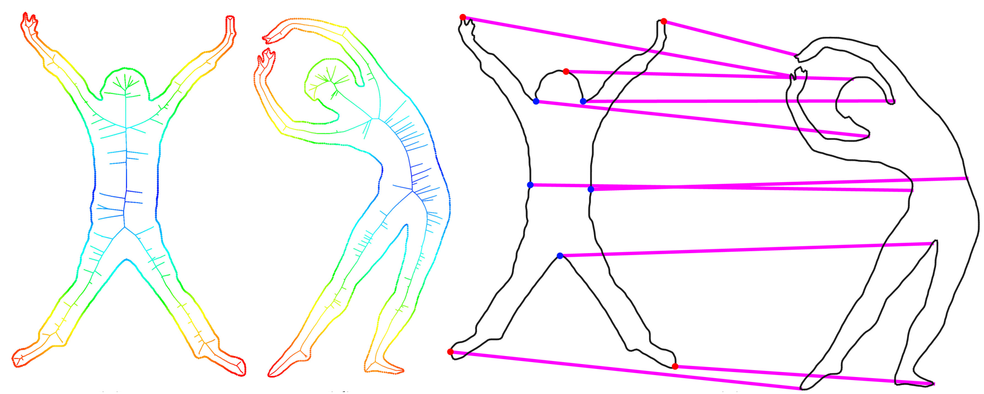
Blum transform
for shape recognition



Vadim Shapiro

character recognition

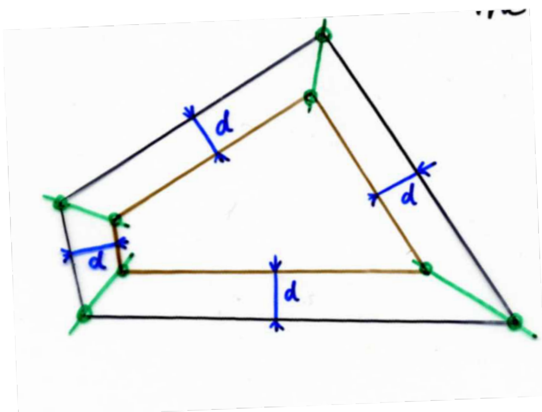
shape matching

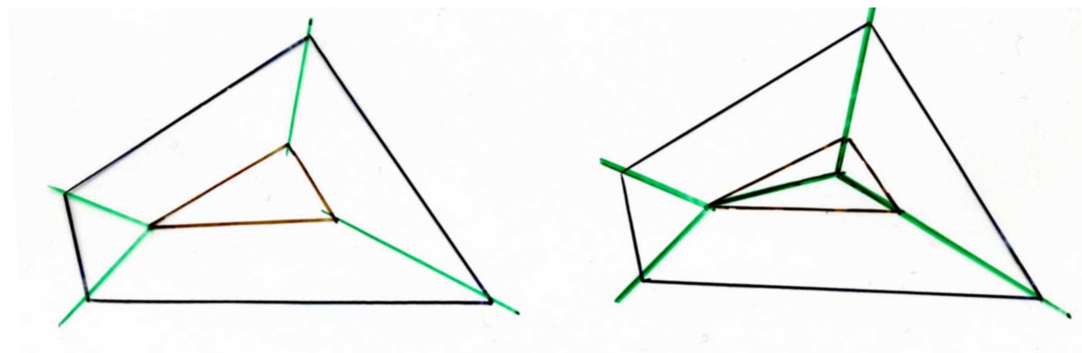


http://www.cs.wustl.edu/~taoju/research/ma_final.pdf

Straight Skeleton — similar to medial axis but avoids curved sections

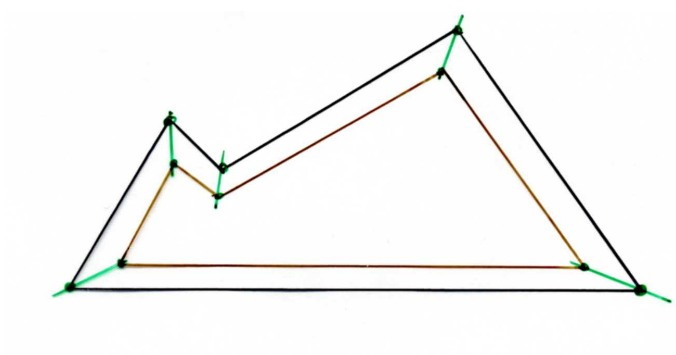
Grow the vertex angle bisectors by shrinking the polygon. The trajectories of the vertices form the straight skeleton.

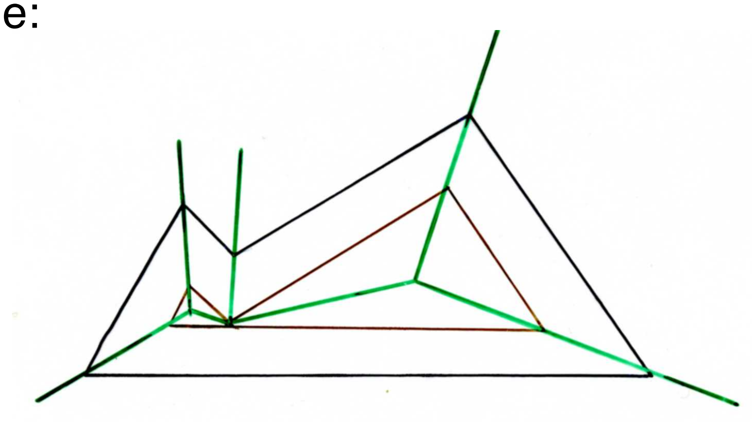




For a convex polygon, this is the same as the medial axis

But for a non-convex polygon, it is not the same:

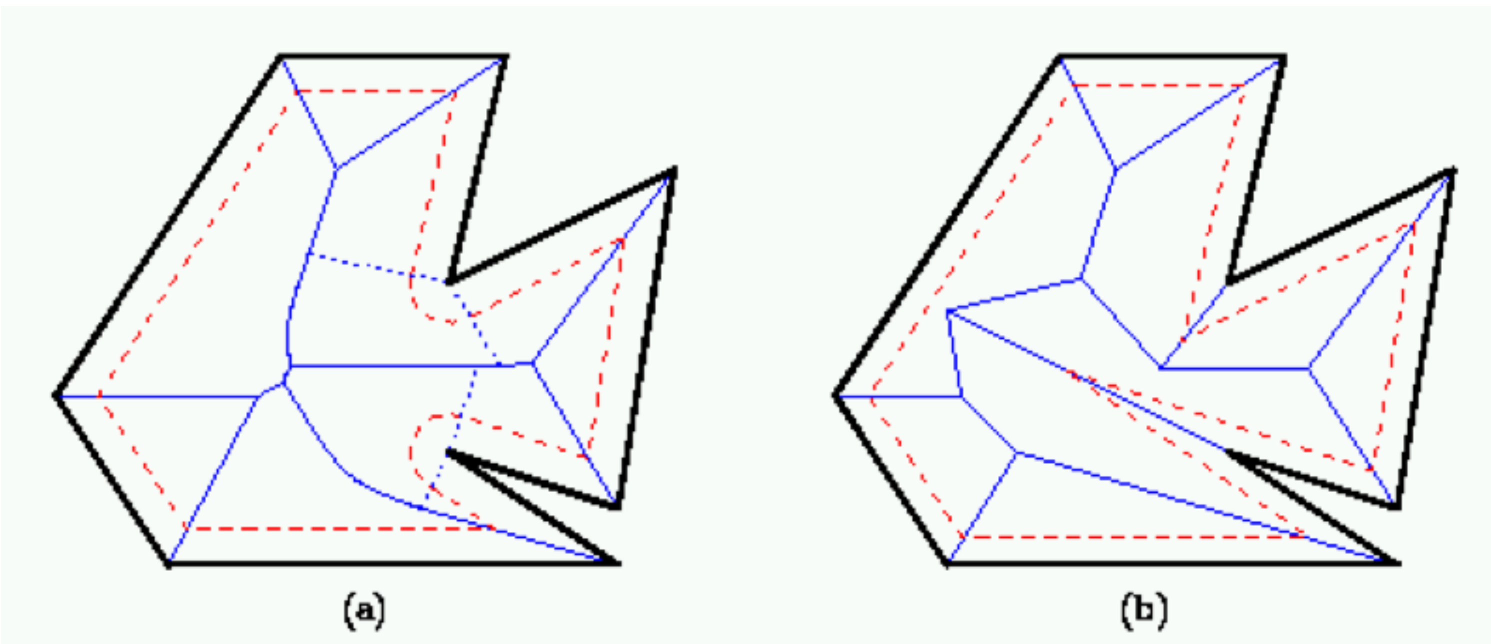




Straight Skeleton — similar to medial axis but avoids curved sections

Difference between medial axis and straight skeleton — only for non-convex polygons:

medial axis straight skeleton



offset curve with mitred caps

Straight skeleton algorithms

idea of previous algorithm gives O(n^2 log n) because the next ray intersection need not be between consecutive rays

improvements:

$O(n^{8/5+\varepsilon})$ for any fixed $\varepsilon > 0$

Eppstein, David, and Jeff Erickson. "Raising roofs, crashing cycles, and playing pool: Applications of a data structure for finding pairwise interactions." *Discrete & Computational Geometry* 22.4 (1999): 569-592

d https://doi.org/10.1007/PL00009479

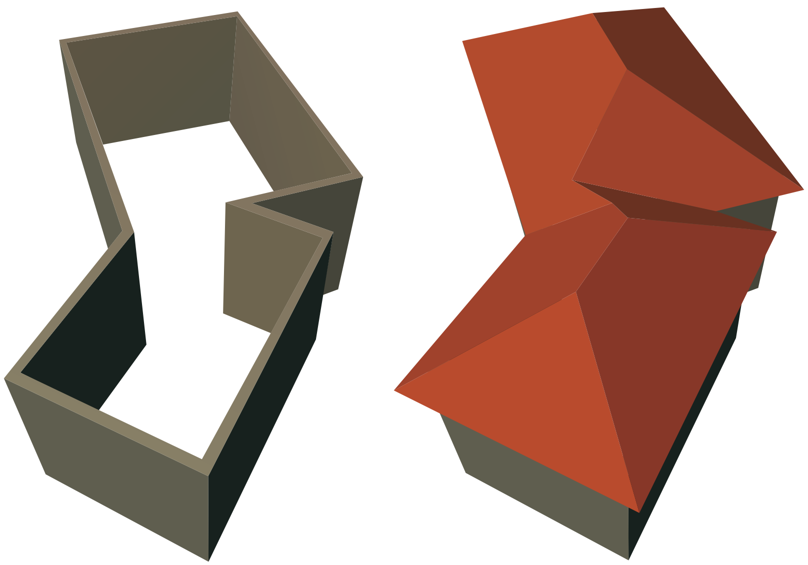
$O(n^{4/3+\varepsilon})$ time for any $\varepsilon > 0$

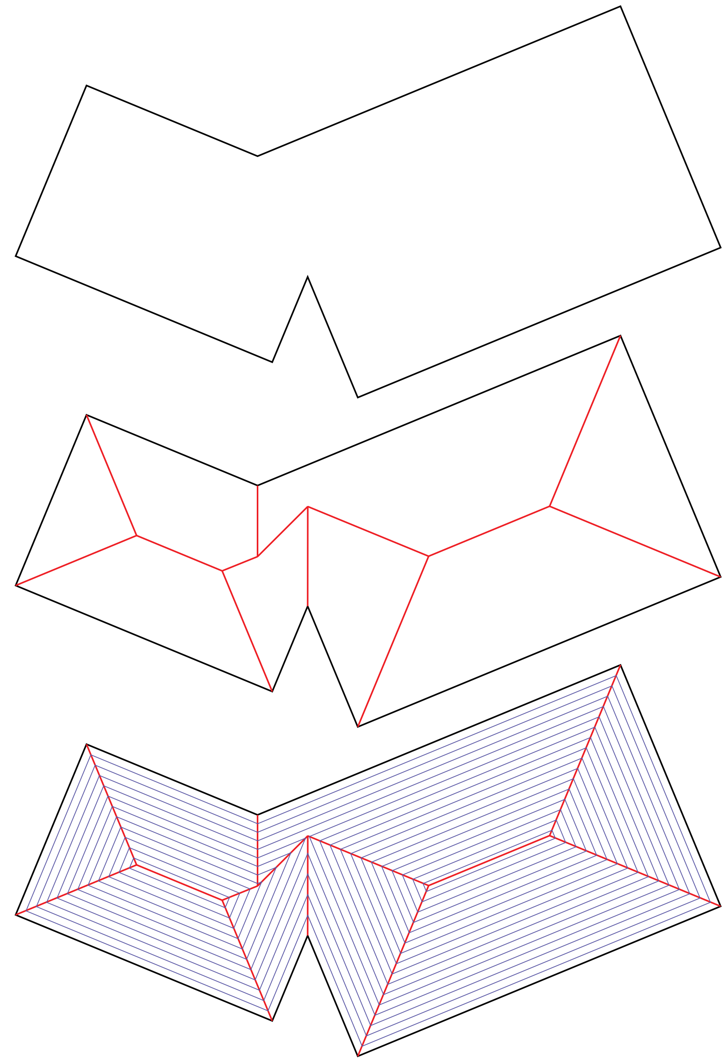
Vigneron, Antoine, and Lie Yan. "A faster algorithm for computing motorcycle graphs." *Discrete & Computational Geometry* 52.3 (2014): 492-514.

d https://doi.org/10.1007/s00454-014-9625-2

Straight skeleton applications: designing roofs

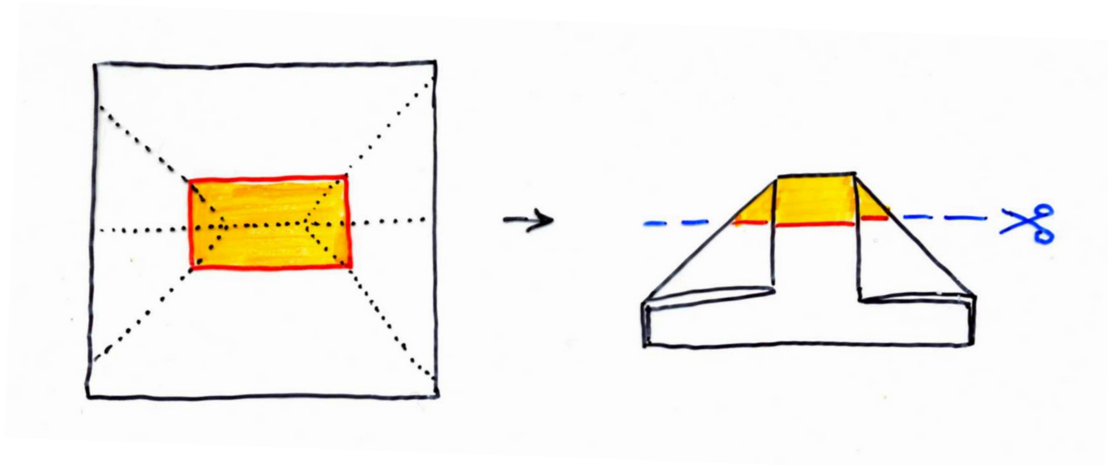
How to fit a roof to these walls?



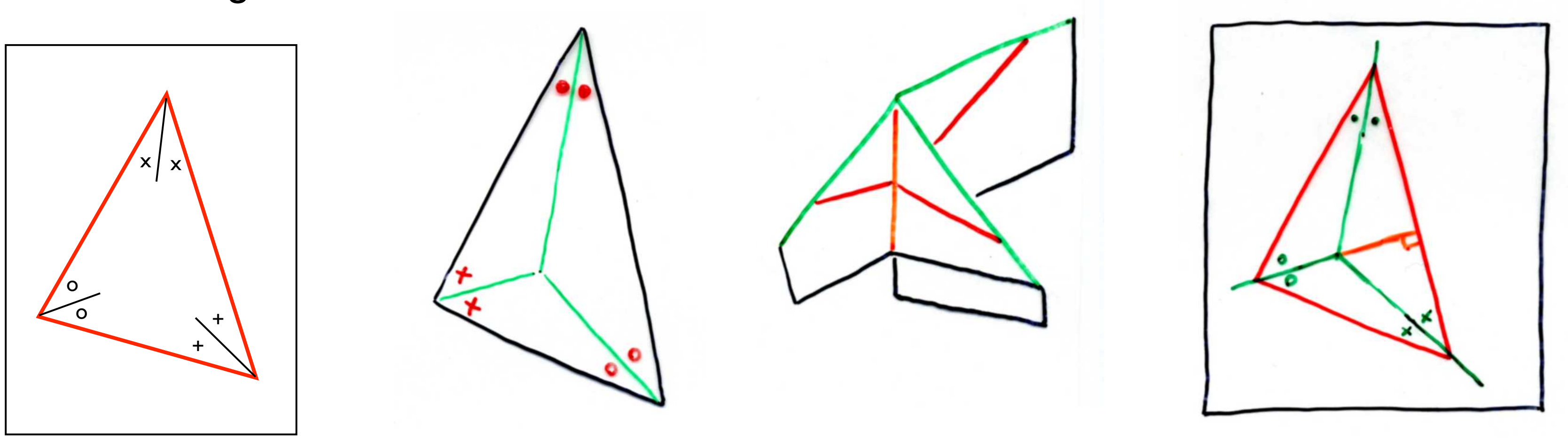


Straight skeleton application: fold and cut problem

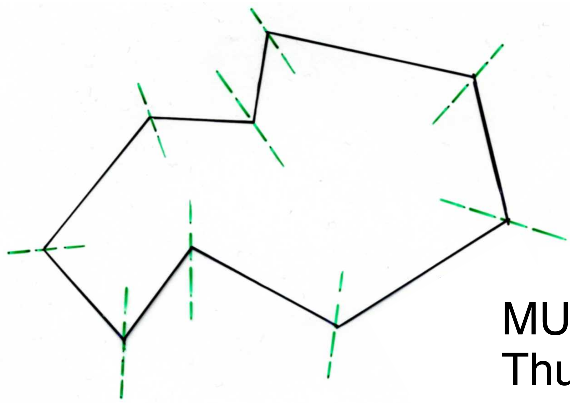
Fold and Cut Theorem. For any (slightly perturbed) polygon on a piece of paper there is a flat folding of the paper that puts all the polygon edges on one line and puts the inside and outside of the polygon on opposite sides of the line.



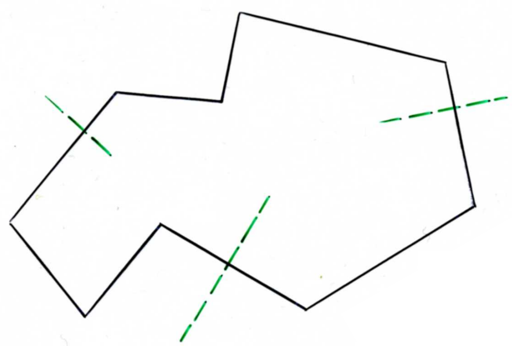
solution for triangle:



general solution to fold-and-cut



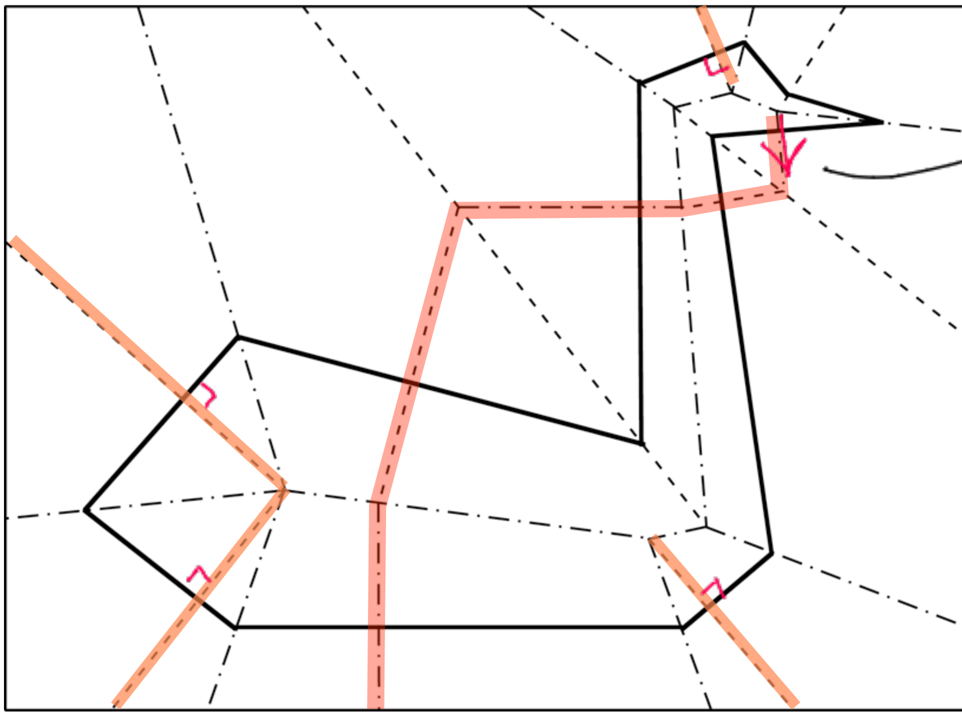
MUST use angle bisector at each vertex.
Thus, use straight skeleton.



MAY use perpendiculars on any edge
and we need some of these to get flat folding

Example.

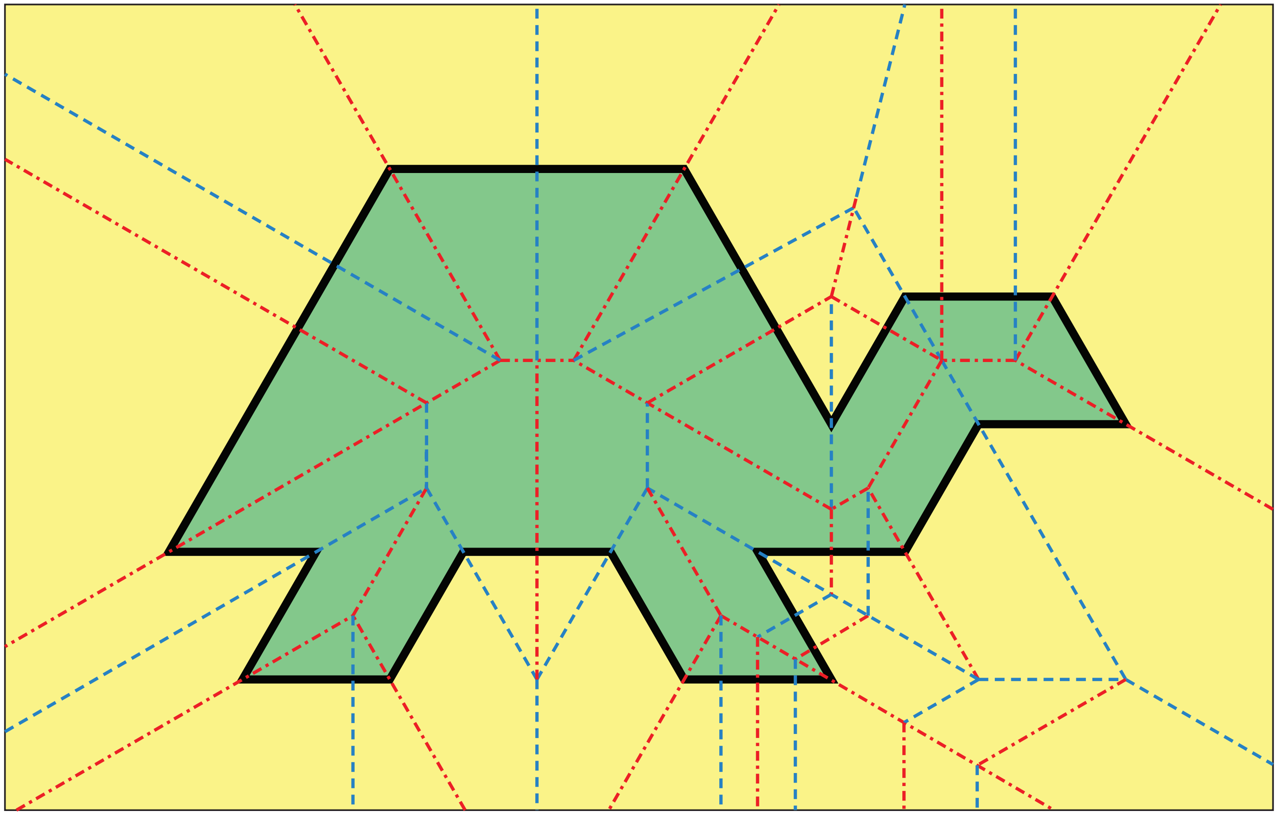
All folds except the pink ones are straight skeleton folds.



this fold "bounces" off other folds

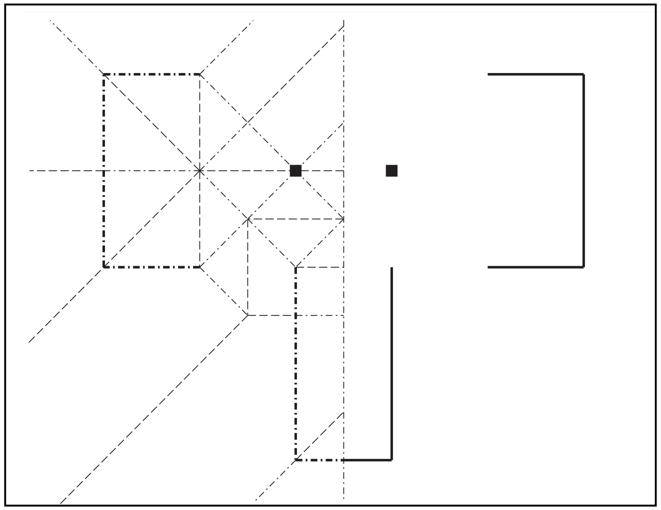
In degenerate cases, this bouncing can be infinite.
This is why we may need to perturb the input polygon slightly.

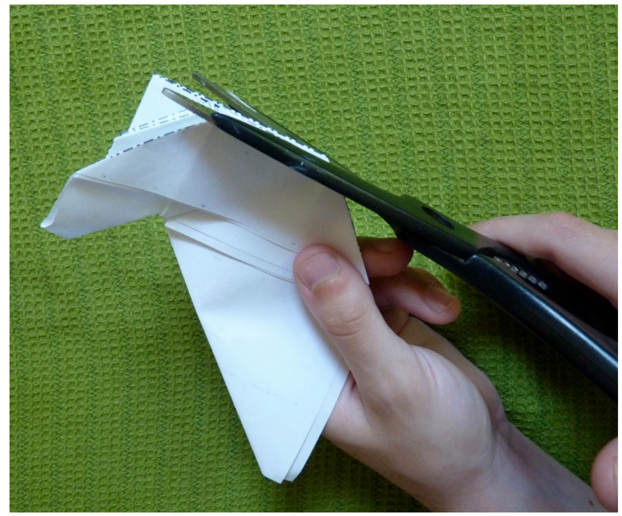
fold-and-cut examples

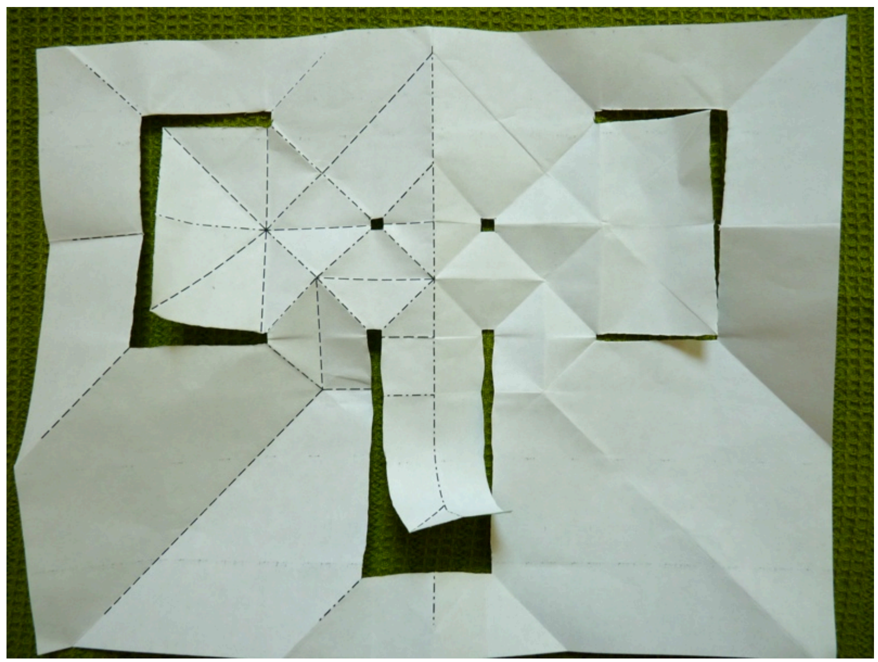


Demaine, Erik D., Martin L. Demaine, and Anna Lubiw. "Folding and one straight cut suffice." *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 1999.

http://erikdemaine.org/foldcut/







Summary

- compatible triangulations and morphing
- curve and surface reconstruction
- medial axis (Voronoi diagram of edges)
- straight skeleton

References

- papers and books listed throughout