Software demo: HoPeS

Cloud segmentation and skeletons

Vitaliy Kurlin, http://kurlin.org
Microsoft Research Cambridge
and Durham University, UK

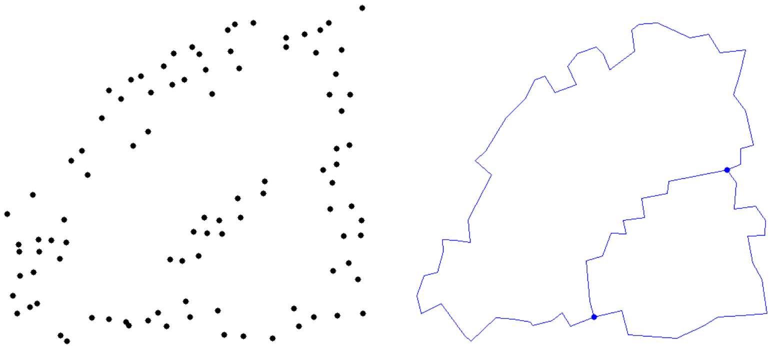
The secondment at Microsoft is supported by the EPSRC *Impact Acceleration Account*

2D cloud software: HoPeS

Input : n points $C \subset \mathbb{R}^2$ with real coordinates

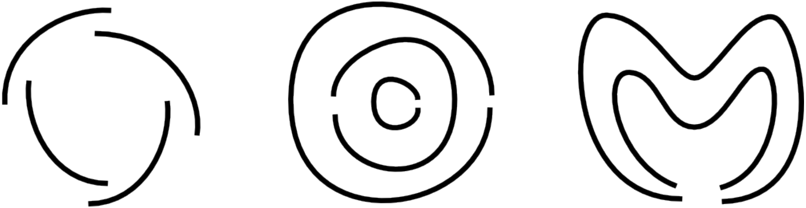
Time: guaranteed $O(n \log n)$ in the worst case

Output : persistent hole boundaries, skeletons



Computer Graphics application

Problem: complete all closed contours or paint all regions that they enclose (a *segmentation*).

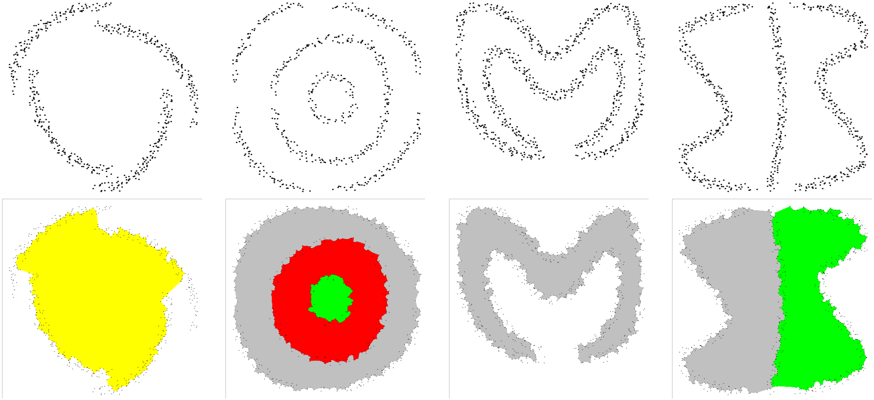


A user drawing a sketch on a tablet might be happy with our fast automatic ‘best guess’:

make contours closed so that I can paint areas

(a scale is easy to find, but we can’t ask for it).

Cloud segmentation into regions

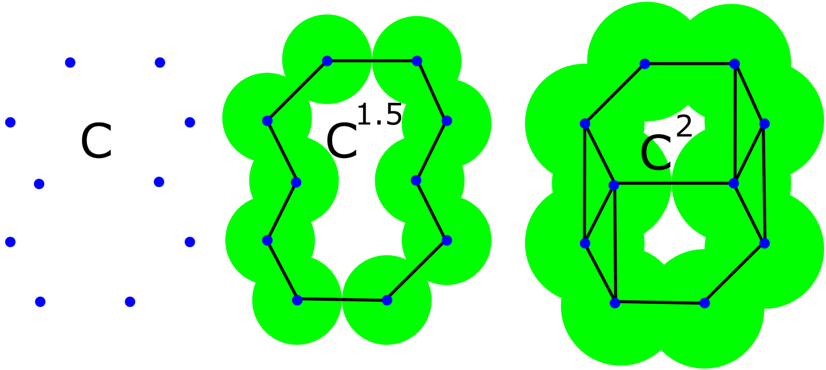


Proved: contours are close to the ground truth.

VK, Pattern Recognition Letters, to appear in 2016

From a cloud to a filtration

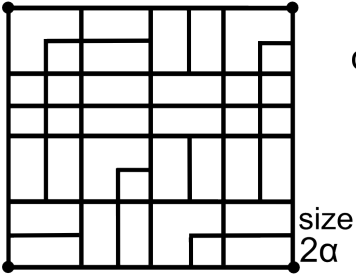
Def : the α-offset of a cloud $C \subset \mathbb{R}^2$ is the union of closed balls $C^\alpha = \cup_{p \in C} B(p; \alpha)$ of a radius α.

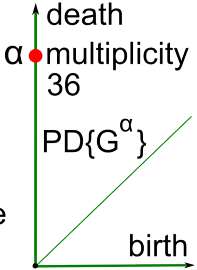


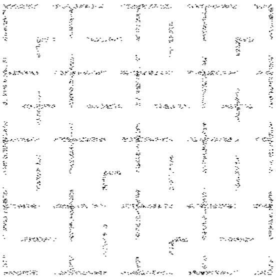
Filtration $C = C^0 \subset \cdots \subset C^\alpha \subset \cdots \subset C^{+\infty} = \mathbb{R}^2$.

Counting holes in C may be easy

The graph G has H_1 of rank 36, hence any ε-sample C of G will probably have 36 holes.

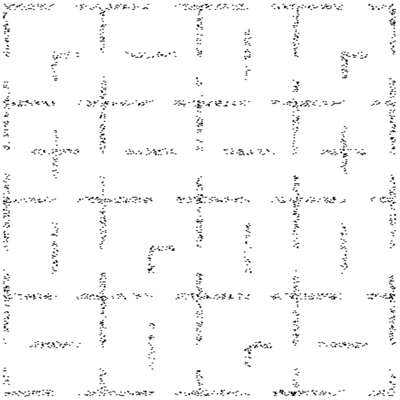


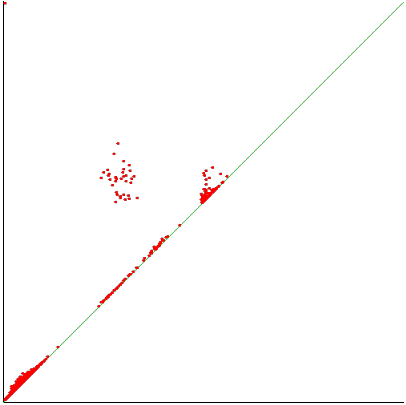




How can we see that there are 36 holes in C?

Using stability of persistence

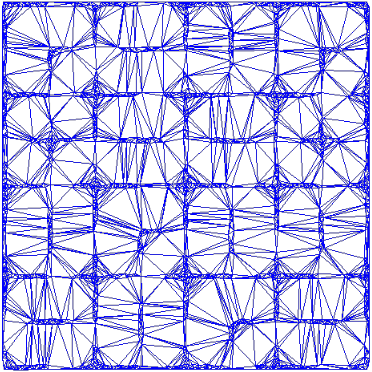


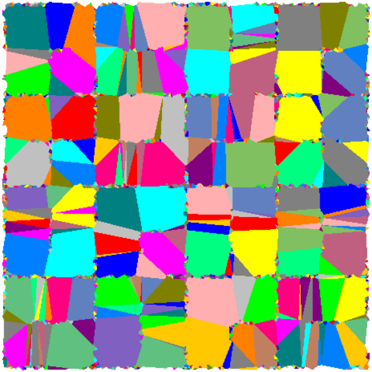


We can find the *widest diagonal gap* separating 36 points from the rest of persistence diagram.

An initial segmentation of C

Acute Delaunay triangle is a ‘center of gravity’.

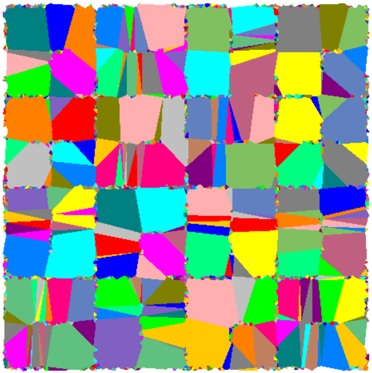


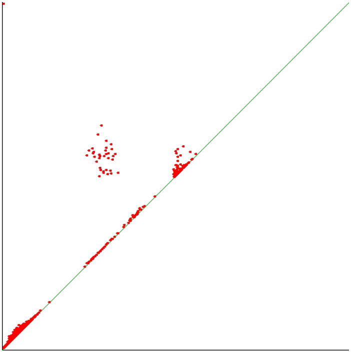


We attach all adjacent non-acute triangles to get an initial segmentation on the right hand side.

Harder than counting cycles

Initial regions $\leftrightarrow$ red dots in PD (too many).

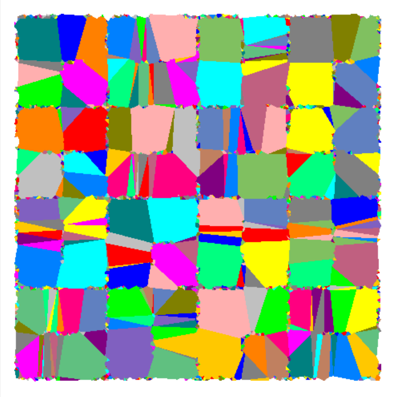


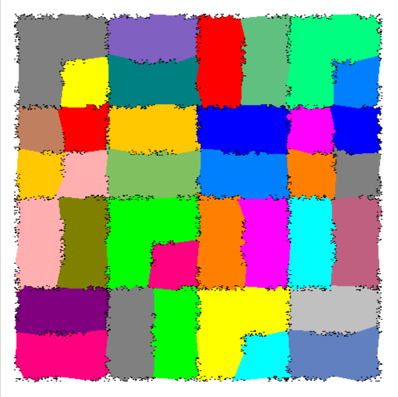


We should merge 36 regions of high persistence with all remaining regions of lower persistence.

Merging initial regions

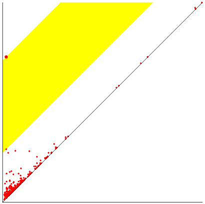
Building $\mathrm{PD}\{C^{\alpha}\}$, we keep adjacency relations of merged regions to enrich persistence info.

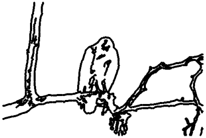


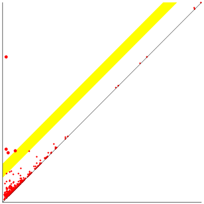


Hierarchy of segmentations

A user can choose to get exactly *k* regions by choosing 2nd widest diagonal gap in PD1 etc.

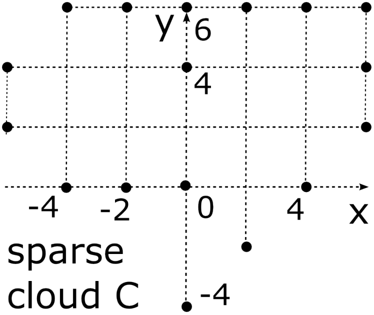


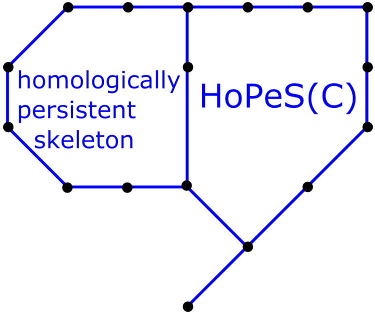




Parameterless skeletonisation

Def : Homologically Persistent Skeleton of a cloud C is $\mathrm{HoPeS}(C) = \mathrm{MST}(C) \cup$ critical edges representing all dots in 1D persistence of $\{C^\alpha\}$.





Properties of $\mathrm{HoPeS}(C)$

Optimality : for any scale α, reduced subgraph $\mathrm{HoPeS}(C;\alpha)$ is *shortest* among all graphs $G \subset C^\alpha$ inducing isomorphisms in H_0, H_1.

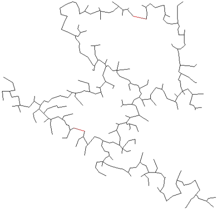
Reconstruction : if C is an ε-sample of a good G, derived $\mathrm{HoPeS}_{k,l}(C) \sim G$ are 2ε*-close* to G.

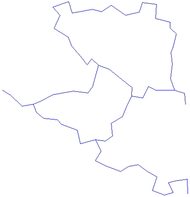
Global stability : $\mathrm{HoPeS}(C)$ remains in a small offset *after perturbing* C. Proofs and extension:

VK, Computer Graphics Forum 34-5 (2015), presented at SGP 2015: Symposium on Geometry Processing.

Recognising visual markers

Shop barcodes are not readable by humans.

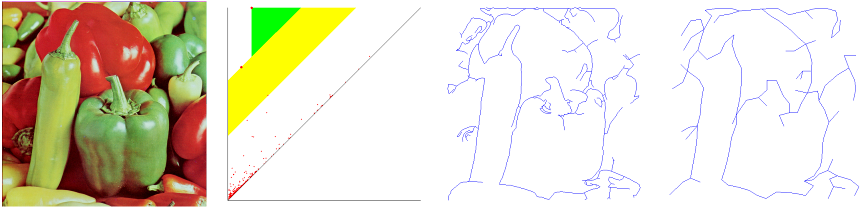




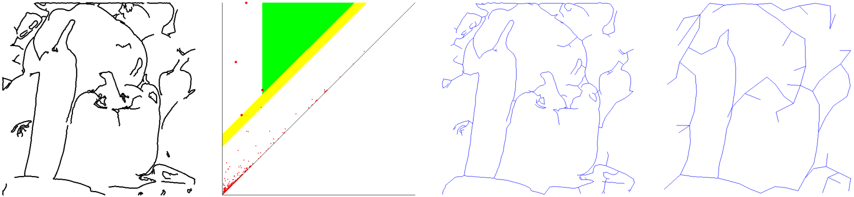
We can make *visual markers* like Egyptian hieroglyphs readable by *humans and robots*.

VK, CAIP'15: Computer Analysis of Images and Patterns

Fast simplification of images



1st widest gap gives contours of 2 large peppers



2nd widest gap gives 2 more small peppers.

Summary: C++ code HoPeS

- *time* $O(n \log n)$ for any input cloud $C \subset \mathbb{R}^2$
- *persistent structures* directly on data with guarantees: boundary contours, Homologically Persistent Skeleton HoPeS
- first persistence software *in England*

Papers and C++ code are at http://kurlin.org.

Collaborations and *applications* are welcome!