

French Onion Soup, Ipelets for Points and Polygons

Klint Faber ✉

University of Maryland, College Park, MD, USA

Auguste H. Gezalyan ✉ 

Department of Computer Science, University of Maryland, College Park, MD, USA

Adam Martinson ✉

University of Maryland, College Park, MD, USA

Aniruddh Mutnuru ✉

University of Maryland, College Park, MD, USA

Nithin Parepally ✉

Department of Computer Science, University of Maryland, College Park, MD, USA

Ryan Parker ✉

University of Maryland, College Park, MD, USA

Mihil Sreenilayam ✉

University of Maryland, College Park, MD, USA

Aram Zaproshyan ✉

University of Maryland, College Park, MD, USA

David M. Mount ✉ 

Department of Computer Science, University of Maryland, College Park, MD, USA

Abstract

There are many structures, both classical and modern, involving point-sets and polygons whose deeper understanding can be facilitated through interactive visualizations. The Ipe extensible drawing editor, developed by Otfried Cheong, is a widely used software system for generating geometric figures. One of its features is the capability to extend its functionality through programs called Ipelets. In this media submission, we showcase a collection of new Ipelets that construct a variety of geometric structures based on point sets and polygons. These include quadtrees, trapezoidal maps, beta skeletons, floating bodies of convex polygons, onion graphs, fractals (Sierpiński triangle and carpet), simple polygon triangulations, and random point sets in simple polygons. All our Ipelets are programmed in Lua and are freely available.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Hilbert metric, Macbeath Regions, Polar Bodies, convexity

Digital Object Identifier 10.4230/LIPIcs.SoCG.2025.83

Category Media Exposition

Related Version *Previous Version:* <https://arxiv.org/abs/2503.01979>

Supplementary Material *Software:* <https://github.com/nithin1527/onion-soup> [8]

archived at `swb:1:dir:8d9d49631e2688049de55f2c874e85e56cb50620`

1 Introduction

We present several **Ipelets** for sets of points and polygons in the $\mathbb{R}^2$ plane. These Ipelets include computing quadtrees, trapezoidal maps, onion decompositions, random point sampling in polygons, beta skeletons, floating bodies, the Sierpiński carpet and triangle fractals, and simple polygon triangulations. Some of these are also implemented in CGAL [9]. All the Ipelets are programmed in Lua and are freely available at <https://github.com/nithin1527/onion-soup>. To install an Ipelet, download the file and place it in the `ipelets` subfolder of your Ipe folder.



© Klint Faber, Auguste H. Gezalyan, Adam Martinson, Aniruddh Mutnuru, Nithin Parepally, Ryan Parker, Mihil Sreenilayam, Aram Zaproshyan, and David M. Mount;
licensed under Creative Commons License CC-BY 4.0

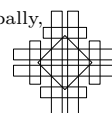
41st International Symposium on Computational Geometry (SoCG 2025).

Editors: Oswin Aichholzer and Haitao Wang; Article No. 83; pp. 83:1–83:6



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

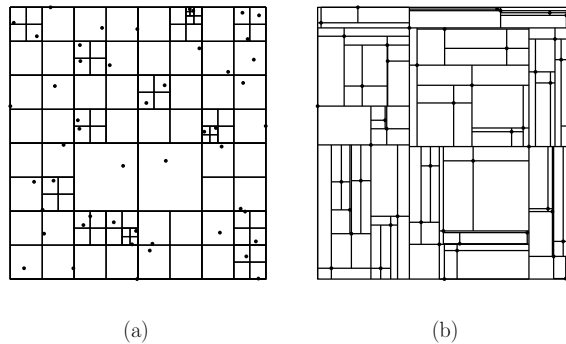


2 Points

In this section, we describe the geometric structures that our Ipelets compute on points. There are a few Ipelets available that already build geometric structures on point sets, such as Voronoi diagrams [5, 14] and Delaunay triangulations [5]. We contribute to this with the additions of the following Ipelets.

2.1 Quadtrees

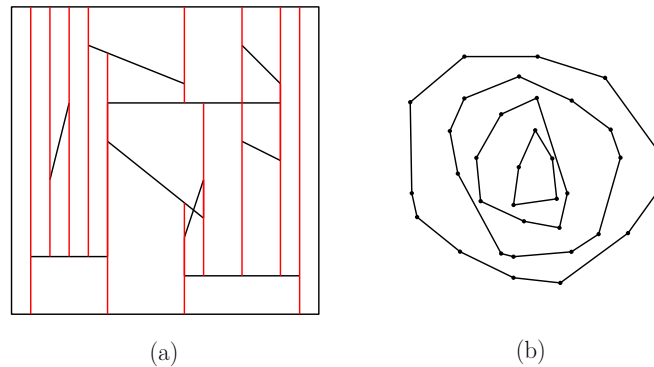
A quadtree is a data structure used to recursively divide two-dimensional space into four regions/nodes, usually named northwest, northeast, southwest, and southeast [6, Ch. 14]. A point quadtree is a specific type of quadtree in which each node houses a point and its four regions are determined by dividing its region into four sub-regions, with the point itself as the center (see Figure 1(b)). A point-region quadtree is another type of quadtree in which each node represents a region of space with a certain maximum capacity of points, such that points are only housed in leaf nodes (see Figure 1(a)). The importance of quadtrees arises from their enabling of efficient operations on data points in two-dimensional space. Consequently, their applications include areas of image representation/processing, mesh generation, and two-dimensional collision detection [6, 16]. We contribute an Ipelet for generating both these types of quadtrees, and we produce the array format for its contents in a copyable form.



■ **Figure 1** (a) Point-region quadtree and (b) point quadtree.

2.2 Trapezoidal Maps

Given a collection of line segments, a trapezoidal map is a subdivision of the plane that respects these segments [6, Ch. 6.1]. The trapezoidal map is created by shooting vertical rays up and down from each of the endpoints of the segments until the vertical line intersects another segment or the bounding box (see Figure 2(a)). In practice, trapezoidal maps allow users to perform planar point location, with respect to the trapezoids that make up the map, in $O(\log(n))$ time. This finds applications in many areas, including geographic information systems, computer-aided design, among others [6, 16]. We provide an Ipelet which, given a set of line segments and an optional bounding box, computes a trapezoidal map of these line segments. It also provides a method for copying the details of the map in an array format.



■ **Figure 2** (a) Trapezoidal map of a set of segments and (b) an onion decomposition of a point set.

2.3 Onion Decompositions

► **Definition 1** (Onion Decomposition). *Given a finite set of points $P \subset \mathbb{R}^n$, the onion decomposition of P is the nested sequence of convex sets $\{C_0, C_1, \dots, C_m\}$, where:*

- (1) C_i is the convex hull of P_i .
- (2) $P_0 = P$ and $P_i = P \setminus \bigcup_{j=0}^{i-1} C_j$, for $i \geq 1$.
- (3) P_m is the last P_i containing any points.

This is the sequence of convex polygons that result by repeatedly taking the convex hull of the set, and then removing these points from the set [15, Ch. 4] [3] (see Figure 2(b)). Onion decompositions of points have found a variety of uses in robust statistics [1, 7] and fractional cascading [4]. We provide an Ipelet that allows users to select a point set and generate its onion decomposition in the plane.

2.4 Beta Skeletons

► **Definition 2** (Beta Skeleton). *Given a set of points (vertices) V and a parameter $\beta \in \mathbb{R}^+$, the beta skeleton of the set of points is the graph $G = (V, E)$ where: There exists an edge $\{v_1, v_2\} \in E$ if and only if for all $v_3 \neq v_1, v_2$, the angle $\angle v_1 v_3 v_2$ is smaller than θ , where:*

$$\theta = \begin{cases} \arcsin \frac{1}{\beta} & \text{if } \beta \geq 1, \\ \pi - \arcsin(\beta) & \text{if } \beta \leq 1. \end{cases}$$

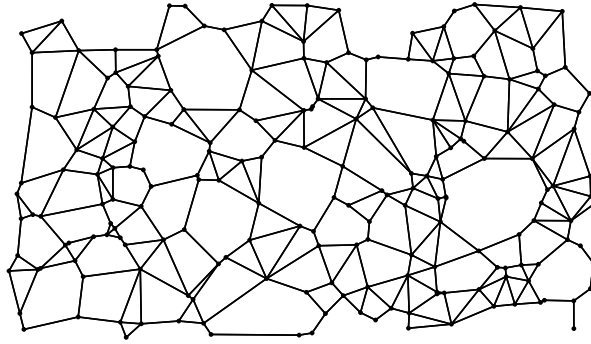
Note that the Gabriel graph and the relative neighborhood graph (RNG) of the point set arise as special cases when $\beta = 1$ and $\beta = 2$, respectively. The beta skeleton [18, Ch. 32] has applications ranging from machine learning [19] to astronomy [10].

3 Polygons

In this section, we describe the geometric structures that our Ipelets compute on polygons, for more, see “Ipelets for the Convex Polygonal Geometry” [13].

3.1 Floating Body

Given a convex body $\Omega \subset \mathbb{R}^d$, a *cap* is the intersection of Ω with a halfspace H . Letting h denote the bounding hyperplane of H , the *base* of the cap is $h \cap \Omega$. When $d = 2$, the *area* of a cap is its Lebesgue measure.

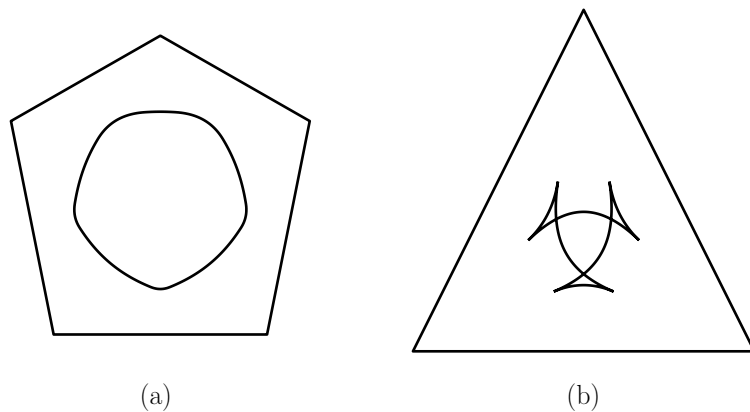


■ **Figure 3** Beta skeleton for a point set for $\beta = 1$, which is equivalent to the Gabriel graph.

► **Definition 3** (Dupin Floating Body). *Given a convex body $\Omega \subset \mathbb{R}^n$ and $\delta \in [0, 1]$, the Dupin floating body, F_δ , is the (possibly non-convex) body defined by the set of points that are the centroids of the bases of all caps that cut off a volume of $\delta \cdot \text{vol}(\Omega)$ [2, Ch. 3].*

► **Definition 4** (Convex Floating Body). *Given a convex body $\Omega \subset \mathbb{R}^n$ and $\delta \in [0, 1]$, the convex floating body K_δ is the intersection of all halfspaces whose defining hyperplanes cut off a set of volume $\delta \cdot \text{vol}(\Omega)$ from Ω [2, Ch. 3] [17].*

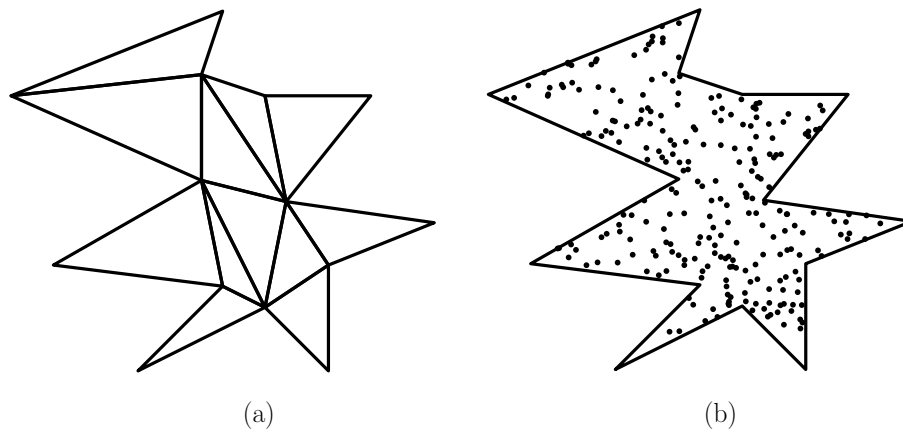
Generally, the Dupin floating body need not be convex (see Figure 4(b)), but when it is convex it equals the convex floating body [2]. We provide an Ipelet for generating the Dupin floating body in arbitrary convex polygons (see Figure 4). For the purposes of the Ipelet we take δ as the percentage of the area of the given polygon in $\mathbb{R}^2$ to cut away with caps. Floating bodies are used for understanding the boundary structure of convex sets through affine surface area [17].



■ **Figure 4** Two examples of the Dupin floating body, (a) convex and (b) non-convex.

3.2 Polygon Triangulations and Point Sampling

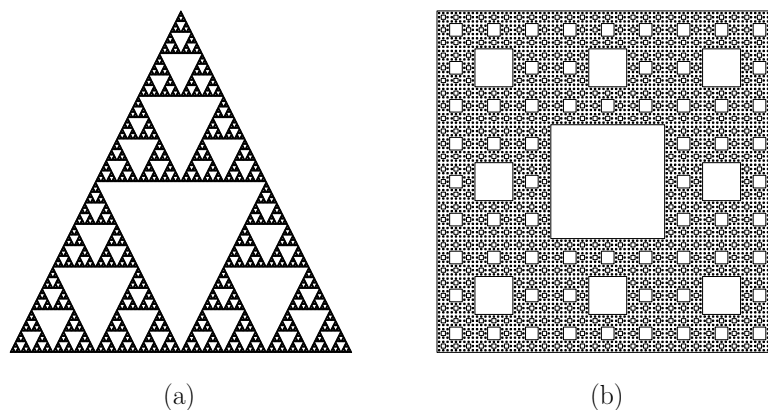
Polygon triangulations are a ubiquitous tool in computer science [6, Ch. 3]. One application of such triangulations is that it allows efficient point sampling inside non-convex polygons. To do this, points are randomly chosen from inside the triangles (weighted by area) using barycentric coordinates. We provide an Ipelet which, given a polygon, decomposes it into a triangulation (based on code by Nils Olovsson [12]) and randomly places a user specified amount of points in it using this method.



■ **Figure 5** (a) Triangulating a simple polygon and (b) randomly sampling points in it.

3.3 Sierpiński Triangle and Carpet

The Sierpiński Triangle and Carpet [11, Ch. 1] are two self similar fractals defined on simple polygons. They are known for having a few interesting properties, such as Lebesgue measure 0 and Hausdorff dimensions $\log_2(3)$ and $\log_3(8)$. They contain uncountably many points.



■ **Figure 6** (a) The Sierpiński Triangle and (b) The Sierpiński Carpet.

References

- 1 Vic Barnett. The ordering of multivariate data. *Journal of the Royal Statistical Society: Series A (General)*, 139(3):318–344, 1976.
- 2 Umut Caglar. Floating bodies. Master’s thesis, Case Western Reserve University, 2010.
- 3 Bernard Chazelle. On the convex layers of a planar set. *IEEE Transactions on Information Theory*, 31(4):509–517, 1985. doi:10.1109/TIT.1985.1057060.
- 4 Bernard Chazelle, Leo J Guibas, and Der-Tsai Lee. The power of geometric duality. *BIT Numerical Mathematics*, 25(1):76–90, 1985. doi:10.1007/BF01934990.
- 5 Otfried Cheong. The IPE extensible drawing editor. Version 7.2.28, 2024-03-12. URL: <https://ipe.otfried.org/>.
- 6 Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition, 2010.

- 7 William F Eddy. Convex hull peeling. In *COMPSTAT 1982 5th Symposium held at Toulouse 1982: Part I: Proceedings in Computational Statistics*, pages 42–47. Springer, 1982.
- 8 Klint Faber, Auguste H. Gezalyan, Adam Martinson, Aniruddh Mutnuru, Nithin Parepally, Ryan Parker, Mihil Sreenilayam, Aram Zaproshyan, and David M. Mount. Ipelets for Points and Polygons. Software, swbId: `swb:1:dir:8d9d49631e2688049de55f2c874e85e56cb50620` (visited on 2025-06-05). URL: <https://github.com/nithin1527/onion-soup>, doi:10.4230/artifacts.23340.
- 9 Andreas Fabri and Sylvain Pion. Cgal: The computational geometry algorithms library. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 538–539, 2009. doi:10.1145/1653771.1653865.
- 10 Feng Fang, Jaime Forero-Romero, Graziano Rossi, Xiao-Dong Li, and Long-Long Feng. β -skeleton analysis of the cosmic web. *Monthly Notices of the Royal Astronomical Society*, 485(4):5276–5284, 2019.
- 11 Gilbert Helmborg. *Getting acquainted with fractals*. Walter de Gruyter, 2008.
- 12 Nils Olovsson. Ear clipping triangulation, 2021. Accessed: 2025-02-25. URL: https://nils-olovsson.se/articles/ear_clipping_triangulation/.
- 13 Nithin Parepally, Ainesh Chatterjee, Auguste H Gezalyan, Hongyang Du, Sukrit Mangla, Kenny Wu, Sarah Hwang, and David M Mount. Ipelets for the convex polygonal geometry (media exposition). In *40th International Symposium on Computational Geometry (SoCG 2024)*, pages 92–1. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024.
- 14 postechDNN. Ipelet. <https://github.com/postechDNN/postechDNN/commits/ipelet/dnn/IPELET>, 2022. (Commit hash: de82a8d).
- 15 Franco P Preparata and Michael I Shamos. *Computational geometry: an introduction*. Springer Science & Business Media, 2012.
- 16 Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.
- 17 Carsten Schütt and Elisabeth Werner. The convex floating body. *Mathematica Scandinavica*, pages 275–290, 1990.
- 18 Csaba D Toth, Joseph O’Rourke, and Jacob E Goodman. *Handbook of discrete and computational geometry*. CRC press, 2017.
- 19 Wan Zhang and Irwin King. Locating support vectors via /spl beta/-skeleton technique. In *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP’02.*, volume 3, pages 1423–1427. IEEE, 2002.