Polygon triangulation

From Wikipedia, the free encyclopedia

In computational geometry, **polygon triangulation** is the decomposition of a polygonal area (simple polygon) **P** into a set of triangles,[1] i.e., finding a set of triangles with pairwise non-intersecting interiors whose union is **P**.

Triangulations may be viewed as special cases of planar straight-line graphs. When there are no holes or added points, triangulations form maximal outerplanar graphs.

Contents [hide]

Polygon triangulation without extra vertices [edit]

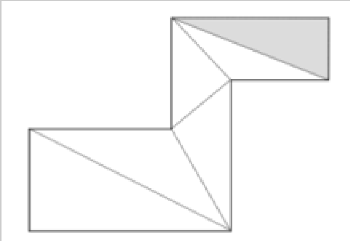
Over time a number of algorithms have been proposed to triangulate a polygon.

Special cases [edit]

A convex polygon is trivial to triangulate in linear time, by adding diagonals from one vertex to all other vertices. The total number of ways to triangulate a convex n-gon by non-intersecting diagonals is the $(n-2)$-th Catalan number, which equals $\frac{n\cdot(n+1)\cdots(2n-4)}{(n-2)!}$, a solution found by Leonhard Euler.[2]

A monotone polygon can be triangulated in linear time with either the algorithm of A. Fournier and D.Y. Montuno,[3] or the algorithm of Godfried Toussaint.[4]

Ear clipping method [edit]



A polygon ear

One way to triangulate a simple polygon is based on the fact that any simple polygon with at least 4 vertices without holes has at least two 'ears', which are triangles with two sides being the edges of the polygon and the third one completely inside it (and with an extra property unimportant for triangulation).[5] The algorithm then consists of finding such an ear, removing it from the polygon (which results in a new polygon that still meets the conditions) and repeating until there is only one triangle left.

This algorithm is easy to implement, but slower than some other algorithms, and it only works on polygons without holes. An implementation that keeps separate lists of convex and concave vertices will run in $O(n^2)$ time. This method is known as *ear clipping* and sometimes *ear trimming*. An efficient algorithm for cutting off ears was discovered by Hossam ElGindy, Hazel Everett, and Godfried Toussaint.[6]

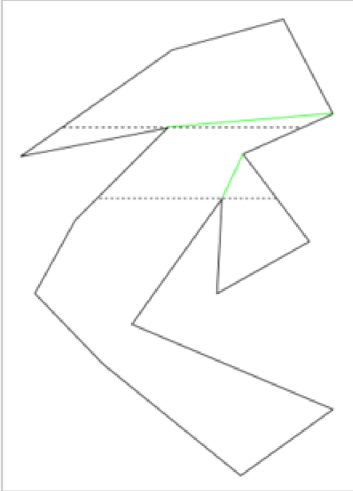
Using monotone polygons [edit]

A simple polygon may be decomposed into monotone polygons as follows.[1]

For each point, check if the neighboring points are both on the same side of the 'sweep line', a horizontal or vertical line on which the point being iterated lies. If they are, check the next sweep line on the other side. Break the polygon on the line between the original point and one of the points on this one.

Note that if you are moving downwards, the points where both of the vertices are below the sweep line are 'split points'. They mark a split in the polygon. From there you have to consider both sides separately.

Using this algorithm to triangulate a simple polygon takes $O(n \log n)$ time.



Breaking a polygon into monotone polygons

Dual graph of a triangulation [edit]

A useful graph that is often associated with a triangulation of a polygon P is the dual graph. Given a triangulation T_P of P, one defines the graph $G(T_P)$ as the graph whose vertex set are the triangles of T_P, two vertices (triangles) being adjacent if and only if they share a diagonal. It is easy to observe that $G(T_P)$ is a tree with maximum degree 3.

Computational complexity [edit]

For a long time, there was an open problem in computational geometry whether a simple polygon can be triangulated faster than $\mathrm{O}(n \log n)$ time.[1] Then, Tarjan & Van Wyk (1988) discovered an $\mathrm{O}(n \log \log n)$-time algorithm for triangulation,[7] later simplified by Kirkpatrick, Klawe & Tarjan (1992).[8] Several improved methods with complexity $\mathrm{O}(n \log^* n)$ (in practice, indistinguishable from linear time) followed.[9][10][11]

Bernard Chazelle showed in 1991 that any simple polygon can be triangulated in linear time, though the proposed algorithm is very complex.[12] A simpler randomized algorithm with linear expected time is also known.[13]

Seidel's decomposition algorithm and Chazelle's triangulation method are discussed in detail in Li & Klette (2011). [14]

The time complexity of triangulation of an n-vertex polygon *with* holes has an $\Omega(n \log n)$ lower bound.[1]

See also [edit]

- Nonzero-rule
- Tessellation
- Catalan number
- Point set triangulation
- Delaunay triangulation
- Tiling by regular polygons
- Minimum-weight triangulation, for a point set and for a simple polygon
- Planar graph
- Polygon covering#Covering a polygon with triangles

References [edit]

1. ^ *a b c d* Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf (2000), *Computational Geometry* (2nd revised ed.), Springer-Verlag, ISBN 3-540-65620-0 Chapter 3: Polygon Triangulation: pp.45–61.
2. ^ Pickover, Clifford A., *The Math Book*, Sterling, 2009: p. 184.
3. ^ Fournier, A.; Montuno, D. Y. (1984), "Triangulating simple polygons and equivalent problems", *ACM Transactions on Graphics* **3** (2): 153–174, doi:10.1145/357337.357341, ISSN 0730-0301.
4. ^ Toussaint, Godfried T. (1984), "A new linear algorithm for triangulating monotone polygons," *Pattern Recognition Letters*, **2** (March):155–158.
5. ^ Meisters, G. H., "Polygons have ears." American Mathematical Monthly 82 (1975). 648–651
6. ^ ElGindy, H., Everett, H., and Toussaint, G. T., (1993) "Slicing an ear using prune-and-search," *Pattern Recognition Letters*, **14**, (9):719–722.
7. ^ Tarjan, Robert E.; Van Wyk, Christopher J. (1988), "An O(*n* log log *n*)-time algorithm for triangulating a simple polygon", *SIAM Journal on Computing* **17** (1): 143–178, doi:10.1137/0217010, MR 925194.
8. ^ Kirkpatrick, David G.; Klawe, Maria M.; Tarjan, Robert E. (1992), "Polygon triangulation in O(*n* log log *n*) time with simple data structures", *Discrete and Computational Geometry* **7** (4): 329–346, doi:10.1007/BF02187846, MR 1148949.
9. ^ Clarkson, Kenneth L.; Tarjan, Robert; van Wyk, Christopher J. (1989), "A fast Las Vegas algorithm for triangulating a simple polygon", *Discrete and Computational Geometry* **4**: 423–432, doi:10.1007/BF02187741.
10. ^ Seidel, Raimund (1991), "A Simple and Fast Incremental Randomized Algorithm for Computing Trapezoidal Decompositions and for Triangulating Polygons", *Computational Geometry: Theory and Applications* **1**: 51–64
11. ^ Clarkson, Kenneth L.; Cole, Richard; Tarjan, Robert E. (1992), "Randomized parallel algorithms for trapezoidal diagrams", *International Journal of Computational Geometry & Applications* **2** (2): 117–133,

doi:10.1142/S0218195992000081, MR 1168952.
12. ^ Chazelle, Bernard (1991), "Triangulating a Simple Polygon in Linear Time", *Discrete & Computational Geometry* **6**: 485–524, doi:10.1007/BF02574703, ISSN 0179-5376
13. ^ Amato, Nancy M.; Goodrich, Michael T.; Ramos, Edgar A. (2001), "A Randomized Algorithm for Triangulating a Simple Polygon in Linear Time", *Discrete & Computational Geometry* **26** (2): 245–265, doi:10.1007/s00454-001-0027-x, ISSN 0179-5376
14. ^ Li, Fajie; Klette, Reinhard (2011), *Euclidean Shortest Paths*, Springer, doi:10.1007/978-1-4471-2256-2, ISBN 978-1-4471-2255-5.

External links [edit]

- Demo as Flash swf, A Sweep Line algorithm.
- Song Ho's explanation of the OpenGL GLU tesselator

Categories: Triangulation (geometry)