

Enumerate all triangulations of a given pointset

This is a Python assignment which shall use CGAL objects to generate *all* triangulations of (the convex hull of) a pointset in the plane. These include regular and nonregular triangulations, as well as triangulations defined by a subset of the given points (the smallest such subset is the vertex set of the convex hull). The graph of all triangulations in the plane is connected.

You should submit a tarfile to compgeom@di.uoa.gr including your code and its execution on at least 3 representative examples.

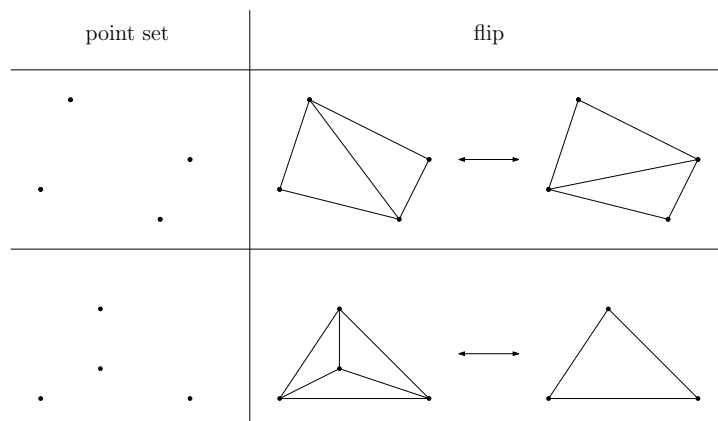
1. Construct an initial triangulation. Write a Python function that asks the user to give a set of 2-dimensional points (using the mouse). Your program should check that the points lie in generic position (no 2 coincident, no 3 colinear) and exit otherwise.

Then construct one, not necessarily regular, triangulation of this pointset. One way to do this is by using CGAL's triangulation via the corresponding Python binding. The manual of CGAL's triangulation might be useful:

http://www.cgal.org/Manual/3.6/doc_html/cgal_manual/Triangulation_2/Chapter_main.html (link)

especially the representation section.

2. Implement flips. There are two kinds of flips, shown below, for 4 points in generic position, depending on whether they are all vertices of their convex hull or not. Write Python functions that implement both flips.



3.Enumeration. Write a Python program that implements a BFS (or DFS) to enumerate all triangulations of the given pointset. The program uses (1) to construct an initial triangulation, and (2) to construct a new triangulation from an existing one. It may store all triangulations to ensure no triangulation is repeated.

4.Bonus. Improve the enumeration. A straightforward improvement is on the way triangulations are stored and searched for. More advanced is to implement reverse search in order to optimize memory consumption.