1.1 Theory

A 2d **Delaunay triangulation** (DT) for a set P of points in the plane is a triangulation such that no point in P is inside the circumcircle of any triangle in the triangulation. It can be shown that for all possible triangulations of P, a Delaunay triangulation maximizes the minimum angle of all angles of the triangles in the triangulation.

Thus, a Delaunay triangulation tends to avoid “skinny” triangles. This property makes it the triangulation of choice for many purposes, including Digital Terrain Modelling (DTM).

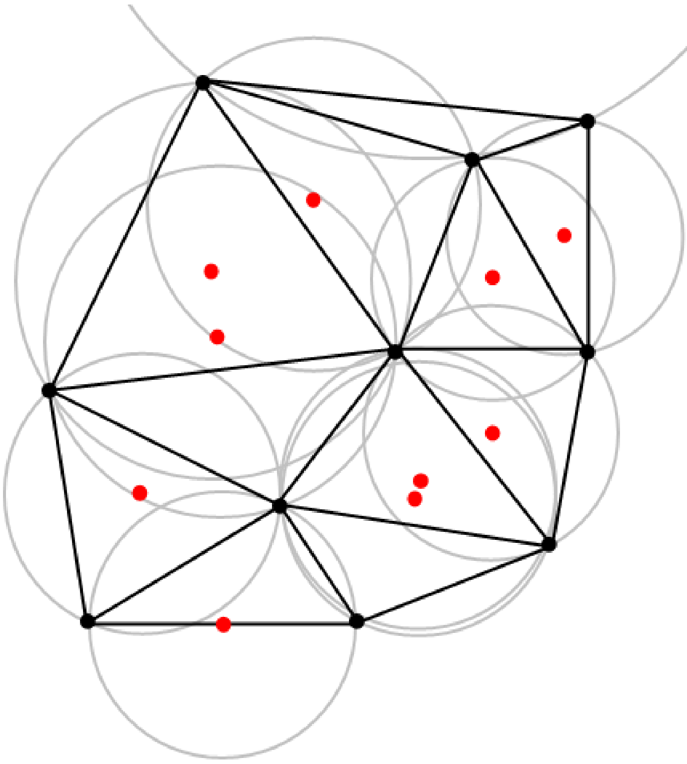

Figure 1: A Delaunay triangulation with all circumcircles and their centres. Image available under GNU license at http://en.wikipedia.org/wiki/File:Delaunay_circumcircles_centers.png

A Delaunay triangulation is unique in a general case. It is not unique if four triangulation points lie on the same circle.

A 2d **conforming Delaunay triangulation** (CDT) is a Delaunay triangulation that respects constraints (edges). This is done by iteratively inserting additional points (called Steiner points) until no triangle crosses a constraint.

Computational power

In general, triangulating pure point data is much faster than triangulation pure constraint data. This is because constraints are processed iteratively until no constraint crosses a triangle. Benchmarks for pure constraint data cannot be given because the number of iterations depends on the distribution of the constraints.

On an average 32bit machine, about 0.5M to 1M points are triangulated in about one minute. This may extend to several millions of points on a 64bit machine. The computational complexity is $O(n \cdot \log(n))$, resulting in near-linear computation time over the number of triangulation points. However, the maximum number of triangulation points is limited by the available amount of memory.

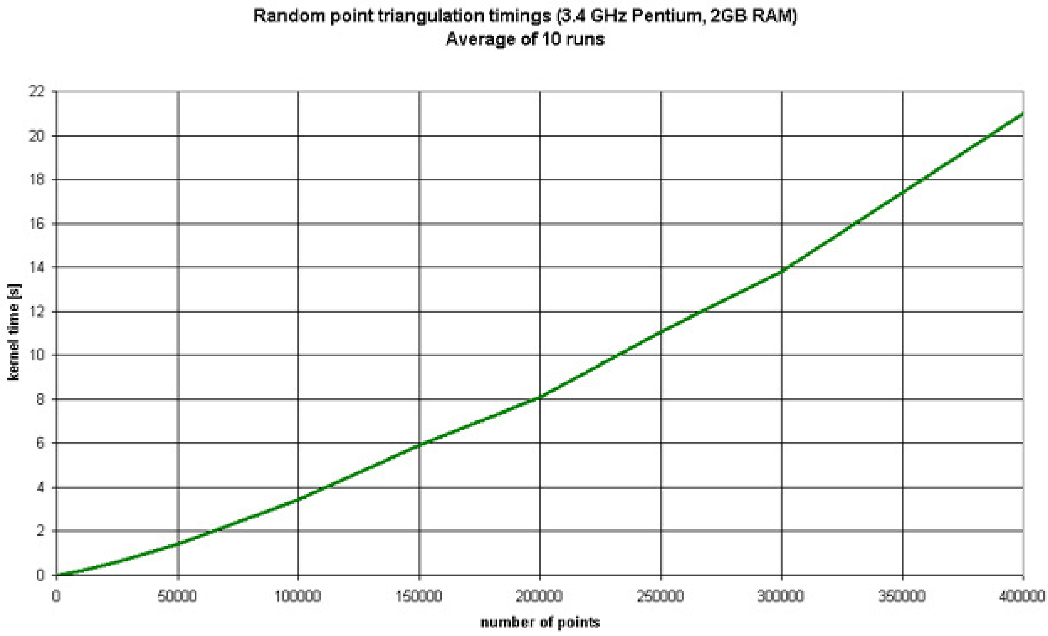

Figure 2: Triangulation timings

1.2 Input data

Following input data will be processed:

- **Triangulation points** are defined as arbitrarily spaced 3d point entities. All triangulation points will be vertices of the triangles in the CDT.

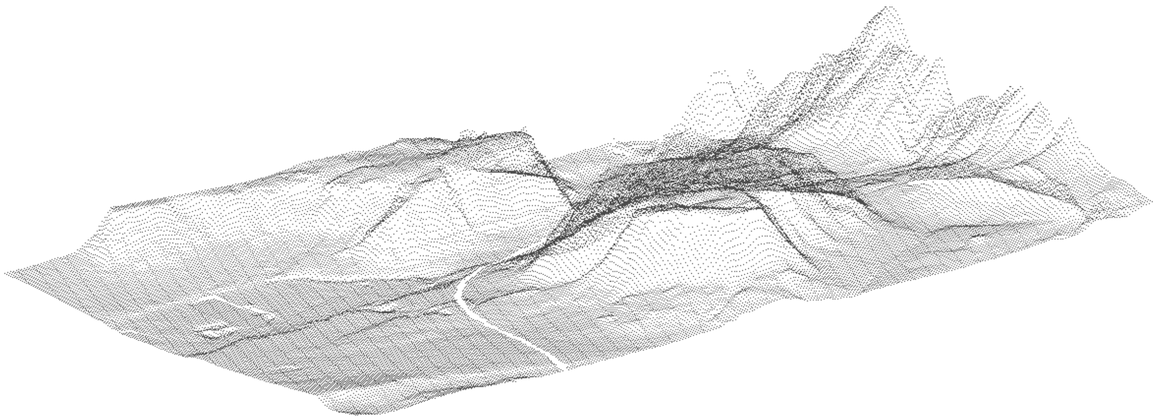

Figure 3: 54k triangulation points

- **Constraints** are defined as an unordered set of non self-intersecting, non-overlapping edge entities. The CDT will insert additional points so that no constraint will cross an edge of a triangle. Thus, spatial constraints become part of the triangulated surface.

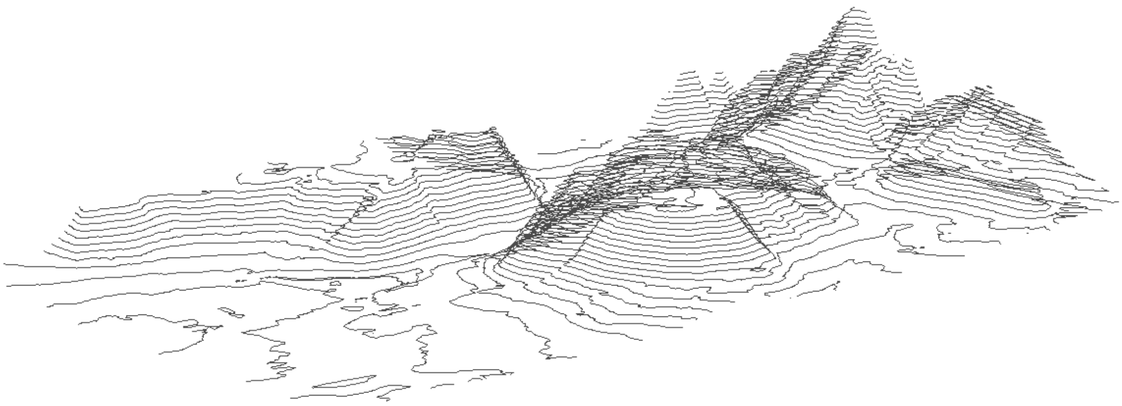

Figure 4: 35k contour lines (constraints)

- **Boundaries** are defined as a set of edges forming an arbitrary shaped closed polygon. Boundary regions may overlap. Boundaries are not exactly part of the triangulation but will be projected on the triangulated surface after the triangulation process. The CDT will insert additional points so that no projected boundary will cross an edge of a triangle. Boundaries allow defining arbitrary shaped convex or concave holes, islands and outer bounds of the triangulated surface. Boundaries have the **effect of a cookie-cutter** on a triangualted surface.

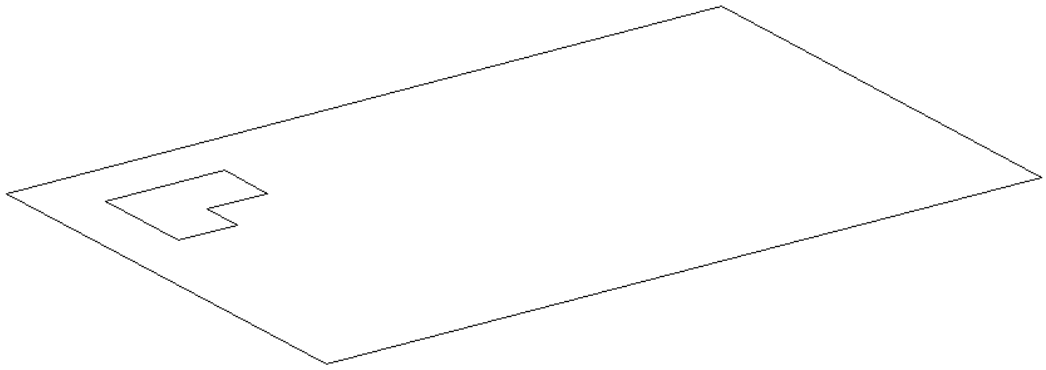

Figure 5: Two closed polygons forming two boundary regions

1.3 Restrictions

There are following general restrictions:

- **The triangulation points and constraints must be projectable**: a 2d CDT can intrinsically not triangulate points with identical xy-coordinates or recessing caves. Therefore, triangulation points with identical xy-coordinates will be removed during the triangulation process. The point with the highest z-coordinate is kept.

  👉 General rule: the CDT can only “see” the xy-projection of the triangulation points and constraints in UCS. It has no height information during the triangulation process.

  👉 Eliminate points with identical xy-coordinates before triangulating. This keeps your input data clean.

- **Constraints must not overlap and must not be self-intersecting**. Constraints may have identical start or end points and may be collinear. However, they must not overlap, be self-intersecting or coincident: since the CDT operates in the UCS xy-plane only, degenerate constraints define over-determined points along their intersection (i.e. possibly different z-heights at the same xy-coordinate).

  👉 Avoid overlapping or coincident constraint edges.

- **Boundaries must be closed and linear**. The CDT only handles bondaries defined by sets of edges forming closed polygons correctly.

  👉 Make sure that each boundary is closed.

- **Boundaries must lie inside the convex hull of triangulation points and constraints**. Boundaries allow defining arbitrary shaped convex and concave shaped holes, islands and bounds in the triangulated surface. This implies that boundaries can only be defined where a triangulated surface existed before, precisely being the area inside the convex hull of triangulation points and constraints.

- **A Delaunay triangulation is not unique over an evenly spaced rectangular raster.** As a consequence, the direction of the diagonal in a raster may alter when triangulating identical point data twice depending on the insertion order of the triangulation points.

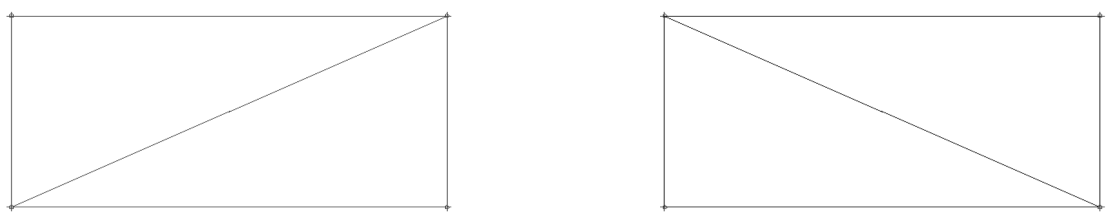

Figure 6: Two valid Delaunay triangulations over a rectangular raster