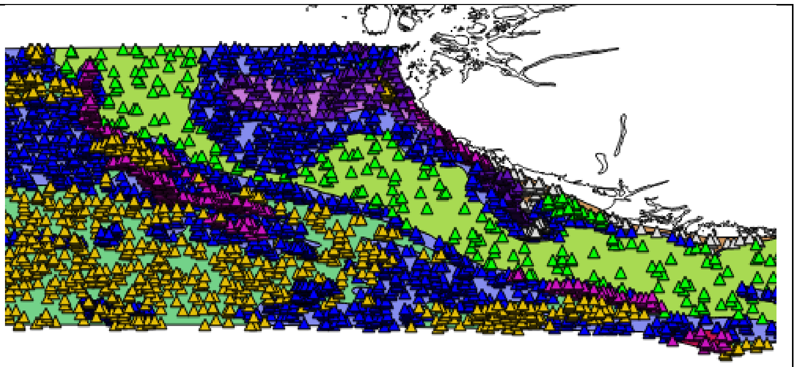
Polygon-to-Points Operations for Integration of Polygon Datasets

CJ Jenkins

June 2018



Point cloud rendition of polygon mappings for sediments and moraines, south of Norway; data courtesy of NGU.

The Problem.

Polygon data show whole areas of the seabed classified as one sediment/rock type or attribute. This form (topology) of data is very difficult to integrate with general point data and also polygon data from other surveys. Remember that dbSEABED works with point data so that samplings from the overlapping surveys of an area can be interspersed and then integrated into griddings, etc.

A polygon allows no exceptions inside. But we know that in areas of 'rock' there are often small basins and channels of sediment – exceptions. By converting the polygon to a point cloud, exceptions sampled in additional, overlapping surveys are allowed to show through into the final griddings of an area.

The Processing.

A Python 2.7 program "dbS_POLYGON_DATA_*.py" (v.4 at present) handles the generation of the point clouds. This allows geometry options beyond the GIS operation like "Vector>Random Points" or ">Regular Points" in QGIS. The program reads a control file such as "py9_euFT.txt" for eu9-affiliated polygon datasets (editing is done in the formatted DOCX equivalent "py9_euFT.docx"). From the control file the shapefile name, new layer name, source of the attributes (column in the shapefile attribute table; or fixed text), the point cloud randomness, point spacing (dg), and the shapefile LL,UR limits (dg) are specified. The program then creates a raster according to the LL,UR dimensions and point spacing (cellsize).

The shapefile geometry is read using "collections" in "fiona" (https://pypi.org/project/Fiona/) with further operations from "Shapely" (https://pypi.org/project/Shapely/). Each polygon is written into the dbSEABED py9 project using a PYG record. The point of each raster cell about the polygon is tested using point-in-polygon from Matplotlib (https://matplotlib.org/) and if inside its location is written to a PIP dbSEABED format line nested under the PYG entry. Later the integrative program "dbS_DPR*.bas" collects the data and interprets the seabed description of the SFT (or other) record under the PYG entry usual techniques to output to a PRS file. Only PRS and CLC data come from polygon data.

Point Cloud Options.

The options for point-cloud construction are: 'fullgrid'– rectilinear (lat/lon) point set, no gaps; 'cloud' – random sampling of the internal space using a statistical subsample of x and y cords, usually of same number as the nodes in the defining polygon; 'outline' – just the nodes of the defining polygon; 'reppoint' - single representative point (Shapely function). It is also possible for a fullgrid or cloud to withdraw from the polygon edge.

Results.

Integration of polygons and point datasets is especially important for rock areas that may have been identified using sidescan sonar. The methods described here ensure that seabed mapping are not biased to just the sediments which are sampled by devices, but include hard-to-sample areas of rock or moraine, for example. A drawback of the method is that the points of the point-cloud are statistical 'pseudo-sample sites' – representative, but never actually sampled.