# Edge Detection in Segmented Non-convex Space Polygon

I. Kožar*, D. Lozzi-Kožar ** and Ž. Jeričević***
* Faculty of Civil Engineering /Department of Computer Modeling, Rijeka, Croatia
** Croatian Water Authority, Rijeka, Croatia
*** Engineering Faculty/Department of Computer Engineering, Rijeka, Croatia
ivicak@gradri.hr

**Abstract - Survey data in use in Civil Engineering consists of points described with XYZ coordinates. Closed shapes, like lakes described with a number of such points are immediately recognizable to humans although points are grouped in unconnected and unrelated lines. Automatic processing of those closed shapes is required if computer analysis is to be performed, i.e. we would like to produce a finite element mesh of the lake. In order to perform meshing one has to be able to determine whether a point is inside or outside of the space polygon, i.e. polygon edges have to be detected. Special computer procedures have been developed for edge detection. The most challenging part is detection of overlaps of edge lines which has been solved with specially modified Fourier transform procedure. Above procedures are applied on a real-life example for creating a finite element mesh of lake "Botonega" in Istria.**

## I. Introduction

Formation of finite element meshes is among the most important tasks in computer modeling (with finite elements used for discretization of the continuum problem). Great effort is devoted to generation of finite element meshes that have to obey some rules to be useful: mesh must correspond to the element type that will be used in the analysis regarding number of nodes per element side, number of nodes and elements has to be kept within given limits, ration of element sides should be between one and about two (distorted elements should be avoided).

Generation of finite element meshes is especially difficult for natural (not man-made) objects since they are highly irregular. Large natural objects like lakes present an additional difficulty by being huge and thus difficult to measure. Traditionally, data is collected through geodesic survey and the main concern is representation of large areas of terrain [1]. In civil engineering it is common to gain information about objects through geodesic survey, too. But here one is more interested in object limits then in areas. Results are groups of uncorrelated points suitable for human interpretation (humans immediately recognize interesting shapes and make conclusions about objects). Mesh of points for computer simulation is quite often produced manually [2], [3].

This work presents a procedure that starts with results of a geodesic survey and finishes with data suitable for automatic finite element mesh generation. Real-life data of Lake "Botonega" in Istria serve as an example.

## II. Edge detection

### A. Data preparation

The starting point in analysis is data from geodesic survey. In our example they consist of points in XYZ coordinate system describing edges and bottom of the lake. Details about data format could be found in our previous paper [4].

The main problem with geodesic data is that they are grouped and each group is not ordered in a way that forms a closed polygon. Namely, survey starts and ends at convenient points and the survey procedure is rarely performed in one "run". The result is a group of unordered points, see Fig.1.

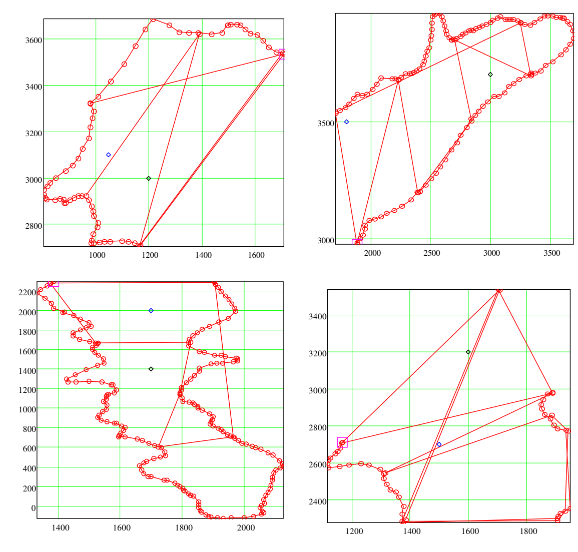

Figure 1. Ordering of points in the geodesic surway (points projected onto the surface of the land). Note the imaginary internal points.

Of course, human can immediately recognize a closed polygon that is represented when points are plotted but it is not so with a computer. Points have to be ordered so that in a vector that contains them the first point is the starting point and the last point is the ending one and the points are connected neighbor to neighbor. That vector represents a closed polygon when the first and last point are connected.

Proper ordering of points in one group is achieved by applying an original algorithm that starts from a point inside the polygon. As a starting point we have taken center of gravity of unordered points. Area of the imaginary triangles is calculated, increase in the area sorts points in the counter-clockwise direction along the edge, see Fig.2.

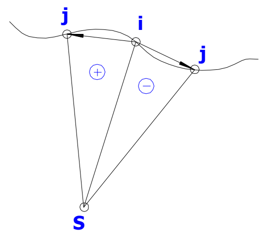


Figure 2. Edge points are oredered regarding the sign of the surface of an imaginary triangle.

Marching through the points is achieved with a proper selection of the next point applying a minimum distance algorithm.

For non-convex polygons this simple rule is not sufficient because not all points could be seen from one pole. In that case new internal point is generated as the gravity center of not visible points and the rule is applied again. The procedure is repeated until there are no “unreachable” points. Since there is known kinematical relation between internal points, proper ordering can be established. Process is repeated until periodicity is detected (using DFT). Resulting polygon is presented in Fig.3.

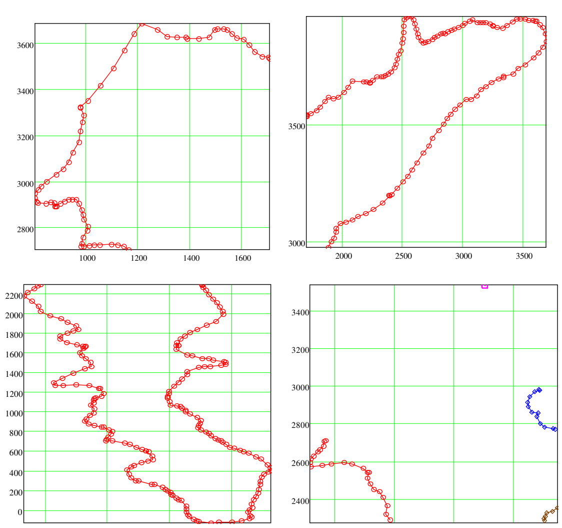

Figure 3. Ordering of points in the geodesic surway. Polygon is not necessarily convex.

Application of the mentioned procedure transforms data from Fig.1 into Fig.3. The basic problem with combination of group of points into a closed polygon can be illustrated with Fig.4.

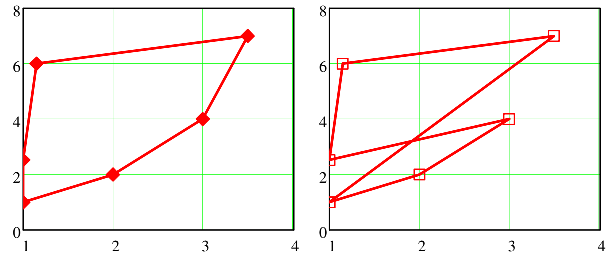

Figure 4. Connecting two groups of points to form a closed polygon.

In Fig.4 one ordering of points is “correct” and the other one is not. It can be easily proved that closed polygon has minimal area when calculated from partial sums along ordered points. However, application of this simple rule is not practical. For rotated points one would have to interchange x and y coordinates. That problem is not present if we take DFT of vector of points and compare their power spectra. It can be shown that the power spectra sum is proportional to the polygon area as described above. Consecutively applying the rule one can combine points into a closed polygon as shown in Fig.5.

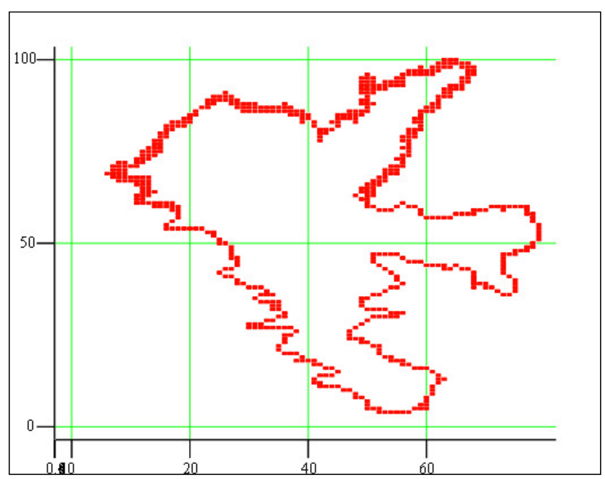

Figure 5. Closed non-convex polygon in normalized coordinates. Note that some points are multiple.

The whole lake “Botonega” is modeled using five polygons that are rotated and scaled in normalized coordinates. The resulting ordered points are presented in Fig.5. Take note that all four polygons from Fig.3 are present in Fig.5 (although scaled and rotated).

Note that due to rotation and scaling in normalized coordinates the edge is not everywhere represented with one line of points.

## *B. Detection of internal points*

Edge of the lake is represented with non-convex polygon from Fig.4. The fact that the edge is not a single line of points (observe multiple points in Fig.4) is the source of confusion in edge determination procedure. Usual polygon fill algorithms [5] or [6] could not correctly detect edges and fill the polygon (in order to determine inside points).

Edge detection is performed in normalized coordinates by dividing the data matrix into rows and columns. In each row edges are represented with 1 and the rest of terrain with 0 as in Fig.5.

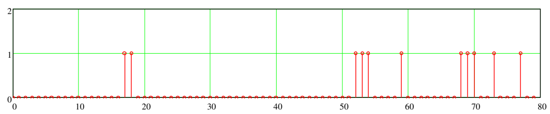

Figure 6. Representation of edges in data matrix. Note that some points are multiple.

Proper edge detection algorithm takes into account multiple points on the edge as well as existence of interior domains. Multiple points are taken into account by forming pairs of changes of data value. Internal domains are resolved by taking into account data from the previous and the next row. In Fig.6 we see red and blue line corresponding to in-line and neighbor rows parameters.

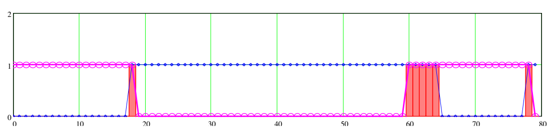

Figure 7. Row of the data matrix as interpreted by edge detection algorithm.

The result of the edge detection algorithm is completely filled non-convex polygon from Fig.4 i.e. internal points have been successfully detected.

### C. *Mesh generation*

With internal points determined we could proceed with generation of the finite element mesh. A computer program has been generated that loops through the data matrix. Only if a point is inside the polygon, loop through the depth is executed and a series of finite elements is generated.

In the example of Lake "Botonega" we have used mesh sized 40 m by 30 m by 1.5 m in depth. Elements of this size would be too distorted so the mesh has to be scaled. Scaling is performed on the differential equation level using the similarity principle. The resulting mesh has 10402 elements and 83216 nodes and is shown in Fig.7. In order to perform calculations, this mesh has to be further processed, i.e. coincident nodes have to me merged. In each node a virtual sphere of given radius is inserted and all inside nodes are coalesced resulting with 15158 remaining nodes. Afterwards nodes and elements are consecutively renumbered. Details could be found in [7].

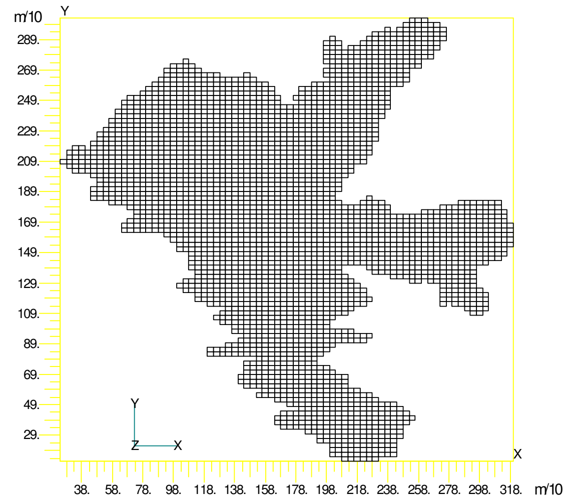

Figure 8. The resulting top layer of finite element mesh of Lake "Botonega".

Successful use of the obtained finite element mesh can be seen in Fig.8. Here we have presented thermal stratification in the lake as a result of calculations on the finite element model from Fig.7.

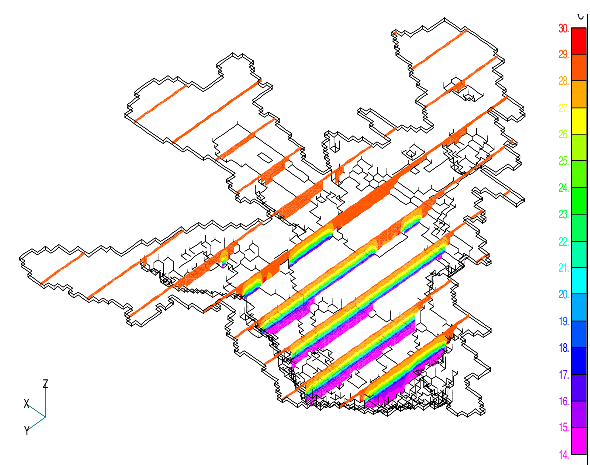

Figure 9. Calculation example using the resulting finite element mesh of Lake "Botonega".

## III. CONLUSION

We have briefly presented the procedure for generation of a finite element mesh from uncorrelated geodesic data whose first part that describes interpolation of geodesic data is given in [4]. In this part we have presented formation of edges of the domain and determination of inside points.

## ACKNOWLEDGMENT

This paper is supported in part by the University of Rijeka through grant no. 13.05.1.1.02.

## REFERENCES


[1] Hjelle Ø., Dæhlen M., Triangulations and Applications, Springer, 2006.

[2] Rathbun S.L., Spatial Modelling in Irregularly Shaped Regions: Kriging Estuaries, Environmetrics, 9, pp. 109-129, 1998.

[3] Rivest M., Marcotte D., Pasquier P., Sparse data integration for the interpolation of concentration measurements using kriging in natural coordinates, Journal of Hydrology, 416–417, pp. 72–82, 2012.

[4] I. Kozar, D. Lozzi/Kozar and Z. Jericevic, "Limit kriging in finite element environmental modeling," in MIPRO 11, P. Biljanovic and K. Skala, Eds. Opatija: MIPRO, 2011, pp. 342–345.

[5] Wayne O. C., Filling Polygons, http:// ezekiel.vancouver. wsu. edu/ ~cs442/ lectures/ raster /polyfill / poly.pdf.

[6] Finley D. R., Efficient Polygon Fill Algorithm, 2007., http://alienryderflex.com/ polygon_fill/

[7] D. Lozzi-Kožar, Three-dimensional Modeling of Thermal Distribution in Lake "Botonega" (in Croatian), Ph.D. thesis, The University of Split, Split, Croatia, 2012.