GEODESY AND CARTOGRAPHY
Vol. 66, No 1, 2017, pp. 23-44

DOI: 10.1515/geocart-2017-0004

Area collapse algorithm computing new curve of 2D geometric objects

Michał Mateusz Buczek

AGH University of Science and Technology
Department of Engineering Surveying and Civil Engineering
30 Mickiewicza St., 30-059 Krakow, Poland
e-mail: mibuczek@agh.edu.pl

Received: 6 January 2017 / Accepted: 5 April 2017

Abstract. The processing of cartographic data demands human involvement. Up-to-date algorithms try to automate a part of this process. The goal is to obtain a digital model, or additional information about shape and topology of input geometric objects. A topological skeleton is one of the most important tools in the branch of science called shape analysis. It represents topological and geometrical characteristics of input data. Its plot depends on using algorithms such as medial axis, skeletonization, erosion, thinning, area collapse and many others. Area collapse, also known as dimension change, replaces input data with lower-dimensional geometric objects like, for example, a polygon with a polygonal chain, a line segment with a point.

The goal of this paper is to introduce a new algorithm for the automatic calculation of polygonal chains representing a 2D polygon. The output is entirely contained within the area of the input polygon, and it has a linear plot without branches. The computational process is automatic and repeatable. The requirements of input data are discussed. The author analyzes results based on the method of computing ends of output polygonal chains. Additional methods to improve results are explored. The algorithm was tested on real-world cartographic data received from BDOT/GESUT databases, and on point clouds from laser scanning. An implementation for computing hatching of embankment is described.

Keywords: shape analysis, Polygon collapse, medial axis, geometry processing, spatial database

1. Introduction

There are many ways to collect data for spatial databases. Elements of the environment its surroundings are measured with classical survey methods such as tacheometry, as well as with modern ones, like terrestrial and airborne laser scanning. Cartographic data, like maps and drawings, can be scanned, vectorized and digitalized.

Nowadays, the progress in the field of obtaining automated data is outpacing the processing of this data. The existing geometrical algorithms still do not match the empirical abilities of a human mind. This is the reason many branches of science, such as point cloud modeling or cartography, require human work. The results of the computerized determination of geometrical and topological features and relations are much worse than those obtained by humans. The branch of science called shape analysis tries to find a solution to these problems through developing even more sophisticated algorithms.

One of the main tools used in shape analysis are algorithms used for the automated determination of geometrical features of datasets, namely symmetry and main axes, skeleton lines, edges, planes, and many others. Processes for fitting geometrical objects into point clouds were designed for 3D modeling. In digital cartography, algorithms for generalization and simplification of shapes are used during the process of changing the map scale.

Mainly, the algorithms for computation of topological skeletons, such as medial axis (MA) or skeletonization, are exploited in determining the shape and the geometric features. These algorithms compute new geometrical objects by simplifying the shapes of the input data. Therefore, the obtained objects require less hard disc space and less computational resources in further processing. These advantages facilitate further processing, as well as spatial and statistical analyses. It is especially essential for big data such as point clouds. Output geometric objects, such as skeletons, can be used in shape analyses for GIS purposes (Haunert and Sester, 2008).

New computational methods are still being researched and developed. Researches try to achieve several goals. Algorithms must be characterized by their effectiveness, result accuracy, topological correctness with the input data, and usage simplicity.

The author introduces an algorithm calculating a polygonal chain approximating the mean axis of 2D polygon. The computational schema is presented and described by a pseudo-code. The purposes, conditions and limitations of the algorithm are discussed. The author indicates the methods of improving the final results. The tests were conducted in simple theoretical cases, as well as on the real-world spatial data from BDOT databases. The implementation of the algorithm, in accordance with the Polish GESUT and BDOT law in force, was presented. The usefulness of the new algorithm in processing big data was tested with data from laser scanning.

2. Algorithms in Shape Analysis

Researches concerning area collapse are a part of a scientific field called shape analysis. Shape analysis studies the automatic determination of geometrical features for digital visualization. It can be obtained by revealing a shape or a boundary of an object (point clouds, 3D models) or new geometries corresponding to the input's shape (geometrical axes, skeleton lines, and directional lines) (Pavlidis, 1980).

The result objects are used in the further analyses of the topology and shape of the input data, and in their visualization. Shape analysis is implemented in computer graphics (Amenta et al., 2001), processing of digital pictures (Brandt et al., 1991), medicine (Bharatkumar et al. 1994), human visual form perception systems (Loncaric, 1998) and many other fields of study. In survey applications, they are primarily applicable in cartography (Haunert and Sester, 2008) and 3D modeling of point clouds (Bucksch et al., 2009).

Asymptotic computational complexity (e.g. big Oh notation) is one of most important indicators of an algorithm's efficiency. It describes the dependency between the size of the input data and the time needed to process them. It does not deal with specific computational steps of algorithms, only their complexity. It can be well illustrated with an example of a FOR loop, existing in almost any programming language (e.g. for (i=0; i < n; i++)). This loop executes a collection of actions for every element in an n-elements set. In total, the operation is repeated n-times, which means that the asymptotic notation equals *O(n)*. Computation time depends on the number of elements via input (n) and the methods in which the algorithm deals with consecutive iterations. Big Oh notation is commonly used in the comparison of an algorithm's efficiency.

A topological skeleton is one of the most important tools in the field of shape analysis. It emphasizes the topological and geometrical properties of the input data. The change of an area into a polygonal chain is called area collapse or dimension change. Computation and plot of the topological skeleton depends on the type of algorithm that is used, for instance, medial axis, skeletonization, erosion, thinning and others.

One of the oldest methods to compute the topological skeleton are Voronoi diagrams (Voronoï, 1908). They divide a plane into regions based on the distance to points (seeds, Voronoi vertexes) which approximates the topological skeleton. Brandt (1994) has proved this with sample density approaching infinity seeds converging to the medial axis. Shamos and Hoey (1975) proposed using Voronoi diagrams in neighborhood analysis. Its advantages were the unambiguity of its results and the efficiency. Since then, the Voronoi diagrams are used as a part of other, more complicated methods.

Skeletonization algorithms are one of the most important groups of collapse algorithms. In the simple approach, the polygon's triangulation is used (Bader and Weibel, 1997). If a polygon's edge is at the same time the edge of a triangle, then the centers of the two remaining edges are points of a skeleton. Otherwise, additional actions are required in order to determine skeleton vertexes, such as Conformal Delaunay Triangulation or Constrained Delaunay Triangulation.

As an alternative, Aichholzer et al. (1996) have proposed the Straight Skeleton. The polygon's boundary is divided into single lines called wavefronts. The waveforms are propagated inside in a shrinking process. Each new skeleton edge is a bisector of two wavefronts (polygon edges). Skeleton vertexes lay on the intersections of consecutive pairs of skeleton lines, or on the point where the wavefront edge collapses to zero length. Adding points on the boundary can improve the skeleton's plot. The output shape resembles a rooftop.

Using the Straight Skeleton, Haunert and Sester (2008) described computing road centerlines from cadastral datasets (geometrical representation of parcels, forms of land use). After creating a topological skeleton, the authors remodeled junctions to fit the predefined demands – in each intersection there are at least three extensions of road axes, and all interception points within a junction must be connected by those extensions. The final result could be used in further GIS analyses.

Skeletonization algorithms are highly effective in the computation of a tree skeleton from airborne and terrestrial laser scanning. Bucksch et al. (2009) have presented the results of their algorithm SkelTre. They generated an octree graph from a point cloud divided into octree cells. The extracted skeleton was characterized by its centeredness and the topological correctness. Dong et al. (2015) have proposed their own method to extract a tree skeleton. They used the constrained Laplacian smoothing method directly on the point cloud, without any surface reconstruction. After the computation of the output object from sample points, the result is smoothened. In both approaches, the algorithm's results can be used in the management, stocktaking and parameterization of forest crops.

Medial axis (MA) algorithms belong to the second large group of collapse procedures. In comparison to the skeletonization, MA provides smoother geometrical results. However, the higher level of computing schema complexity can undoubtedly be treated as one of the main disadvantages of this approach.

The main goal of this algorithm is to find points equidistant from at least two edges of a polygon (Blum, 1967). Connecting those points results in a new geometrical object strictly related to the original polygon.

In order to present the idea behind this solution, Blum (1967) compared its working to a grass fire. Let us assume that the boundary of a shape is on fire. The fronts of the flames move inward at a uniform rate. The set of points indicated by places where two different fronts meet and both extinguish, is the set of medial axis points (Blum, 1967; Chin et al., 1999).

Despite the simplicity of this idea, the computation of equidistant points proves problematic. Blum proposed a theoretical approach to determine the distance to the shape's boundaries from every point within the shape. It is mathematically impossible to check all points in finite time. Researchers are looking for new solutions using, among others, discrete points to compute new geometrical objects approximating the medial axis (Aichholzer and Aurenhammer, 1996; Dey and Zhao, 2004).

The second main area of medial axis development concerns its efficiency. In 1982, Lee presented an algorithm for simple shapes with n-vertexes in time $O(n \cdot log(n))$. A few years later an algorithm designed to calculate Voronoi diagrams and medial axis in linear time $O(n)$ was devised (Aggarwal et al., 1989). Since then, new algorithms have to show similar efficiency and they must simplify computational schema in order to facilitate implementation in CAD programs.

Additional mathematical tools were proposed in order to improve calculations and approximation of medial axis, e.g. Lemma decomposition (Wee, 1997), histogram decomposition (Djidjev and Lingas, 2004), Sturm series (Culver et al., 2004), and many others. They have a positive influence on the processing, particularly for more complicated geometrical objects, especially in 3D.

To generalize Blum's assumptions, a medial axis is a result of an infinite union of balls (circles in 2D). Each medial axis vertex may be a center of a sphere (circle) tangent to a boundary at least in two points. Amenta et al. (2001) assumed the MAT approximation as a finite union of balls and defined a new, piecewise-linear approximation called Power Crust.

From a sufficiently dense sample of points, the subsets of the Voronoi balls are selected. The result is a surface reconstructed from the sample points. It is also a boundary of a polyhedral solid. The authors proved that, as the sampling density goes to infinity, Power Crust's results converge with the ones provided by the medial axis.

Medial axis algorithms are implemented in survey applications, including those for point cloud processing. Lee (2000) presented an approach to approximate a simple curve without self-intersections. The input data was a set of unorganized points in the form of a point cloud.

The point cloud was reduced to a thin curve-like shape using the improved moving least-squares technique. The next step was to reconstruct a smooth curve. The author suggested the implementation of his algorithm to reconstruct surface and model point sets for e.g. pipelines.

Fogg et al. (2014) described the usage of medial axis in the process of generating mesh. At the beginning, medial axis is used in the effective determination of mesh singularities. Topological and geometrical information is used in order to eliminate these singularities. The final mesh is adequate to the geometric shape and deals effectively with concavities of the input objects.

The shape analysis also deals with the problem of shape matching between 2D and 3D objects, and reconstruction of 3D objects from 2D objects. Matching algorithms face problems with the optimality and quality of the correspondence, for example.

Scientists proposed different approaches in finding 3D helical curve from 2D data, like using generalized helicoids (Piuze et al., 2011), computing based on matching the input curve by its orthogonal projection (Cherin et al., 2014), or based on the best approximation of noisy input (Cordier et al., 2016).

Willcocks et al. (2016) introduced an algorithm to find 3D parametric curves in three steps. In the first step, they compute the main structural curve, as the main part of the computed 2D morphological skeleton. Then, the image is straightened on the structural curve and, finally, the 3D parametric curve is computed from the image. The usage of the skeleton for more complicated input objects makes the computation impossible, because it's hard to identify the main structural curve from a no tree-shaped skeleton.

Lähner et al. (2016) presented an automatic method for non-rigid 2D-to-3D matching. The continuous curve on the 3D target surface is computed from the input 2D closed planar curve. Both input objects are discretized and can be described by *m*

(2D curve) and *n* vertices (3D surface). The algorithm recognizes matching vertices optimally, based on the minimal energy function using the function of distance. Because input objects are compared locally, the shape of 3D surfaces may vary from the input 2D curve. The result can be obtained in $O(mn^2 \log(n))$ time.

The modern digital cartography deals with the problems of dimension change, area collapse, and searching for characteristic vertexes. Kozioł et al. (2012) created an automated algorithm which searches for so-called cartographic control points. It is used in the generalization process of input data. The net of connected control points is similar to the topological skeleton. However, its vertexes density is lower. That is why certain elements may intersect the boundary of the original shape.

Another approach to the generalization process was presented by Szombara (2013). He combined the medial axis transformation with the generalization tool using the Perkal method (the construction of an elemental circle) and Chrobak's elemental triangle. It provided a new, automatic tool for the cartographic generalization of digital data stored in spatial databases. The algorithm's output is repeatable and unaffected by the operator's actions.

Scientists also propose using mathematical tools like spline curves for modeling 2D objects and point sets. Researchers focus on creating new tools to achieve their smoothness, good characteristics, and properties etc. An example of such a method, weighted-quadratic trigonometric spline, was presented by Sarfraz et al. (2016). Other researchers focused on the application of those tools for 2D and 3D data (Lenda, 2006, Lenda and Mirek, 2013). The results of those methods depend strongly on the shape of the input objects. Preserving the function's smoothness is more important than presenting the true shape of the described objects.

In software, Rhinoceros was implemented for the simple solution of computing the average curve, called the "Mean Curve". The algorithm finds the pairs of points in proportional distances from the ends of the input curves. For each pair of points, the mean curve goes through the point in between them. For two, similar curves, the results are satisfying. For more complicated objects, the result objects can lay partly outside both input curves.

None of the solutions presented above are simultaneously highly efficient, easily implementable, and likely to produce a satisfying approximation of the real mean axis. Some geometrical operations do not require a high precision level offered by algorithms to compute medial axis or skeletonization. The author of this paper sets himself a goal of creating a computation schema that is simple in implementing, which would fulfill several conditions at the same time.

The main conditions are repeatability, and receiving a plot approximating the shape of the input data. This was achieved by following the strict method of computing vertexes. The second condition is that the output object should be contained in its entirety in the input data (the overlap is allowed). It requires an additional topological control during the computation of vertexes. The third condition is that, in opposition to the algorithms presented above, the plot should lack branches. The algorithm was tailored to satisfy the requirements of the Polish law in force (BDOT, 2015). It must compute an axis of elongated polygons, which is by definition, without branches. The asymptotic computational complexity of this algorithm equals *O(n)*, with n determined during the first step, while setting parameters.

3. Method

The presented algorithm requires the input data in the form of a line chain, hereafter referred to as "object" or "shape." A computation schema of vertexes is similar to the medial axis and the skeletonization.

The new output shape, obtained with the help of the described algorithm, will be called the "curve of minimal radii" (CMR) or "output curve." All vertexes of the CMR are equidistant to at least two elements of the input shape's boundary, just like the medial axis. The method of vertex determination is different than the one used for the MA. In the CMR, the algorithm does not check every point contained by the shape, but it computes consecutive vertexes using points on the line chain.

The algorithm requires input data in the form of a shape divided into two, approximately even, and similar parts. Before the computation of the consecutive CMR vertexes, one of these two parts is chosen to be the Main Curve (MC), and the other one – the Auxiliary Curve (AC). If the input shape contains two disjoint polygon chains, then the main curve is one of them, and the auxiliary curve is the other one (Figure 1).

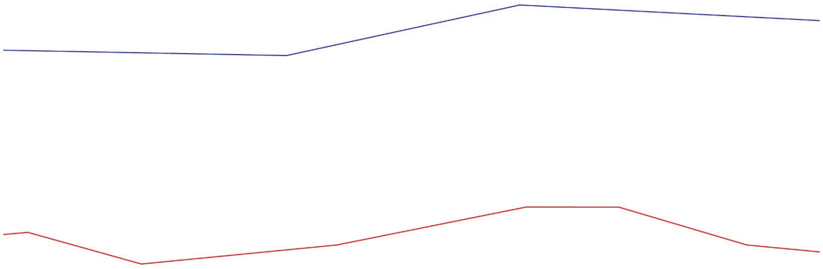


Fig. 1. Example of a disjoined object (red – MC, blue – AC)

If the input shape is a polygon, the MC is a continuous part of its boundary, and the AC – its remaining part (Figure 2). There cannot be a situation in which the boundary's segment belongs to neither of these two curves, or belongs to both of them at the same time. The proper choice of the MC and the AC determines the shape of the CMR.

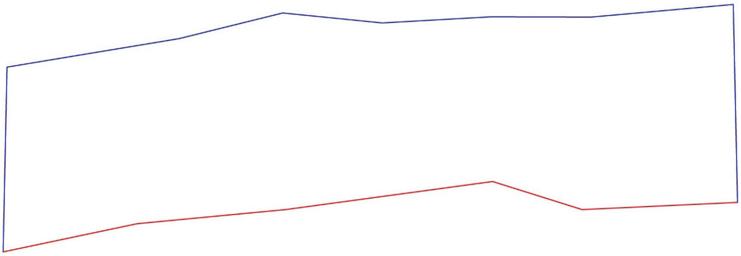


Fig. 2. Example of a polygonal object (red – MC, blue – AC)

At the beginning, points P_i are computed on the main curve in fixed intervals. Each element of P_i in the point set is the center of a circle with its radius approaching infinity (Figure 3a). For each segment of the auxiliary curve, a circle with a tangent point on this segment is found. As a result, a point set of tangent points E_j is created.

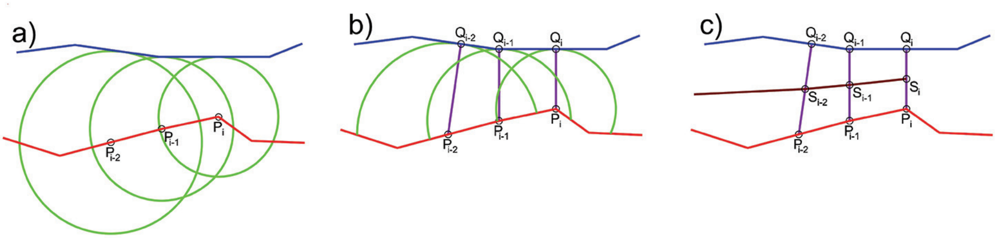


Fig. 3. a) determining circles b) computing minimal radii c) creating curve of minimal radii (red – MC, blue – AC, green – circle, purple – radius, brown – CMR)

Several geometrical conditions have to be respected during the calculation of a point E_j. The whole line segment P_iE_j must be contained within the object. If it intersects the boundary, then the point E_j is ignored. Then the algorithm searches for another point on this segment of the AC, which could create a qualifying line segment P_iE_j. If such a point does not exist, the algorithm continues with the calculations for the next segment of the AC. This topological condition guarantees that all of the obtained points of the CMR will be inside (covered by) the boundary of the input shape.

Figure 4 depicts the process of choosing points on AC. Segment connecting point P_i with closest point E_j doesn't meet the topological conditions described above. The same situation is with the next closest point E_{j+1}. Both segments intersect MC. Thus, the part of them that lays outside the figure and in some situations the segment's middle point, could also lay outside the object. The closest points E_{j+2} and E_{j+3} on the next two segments of AC meet the topological conditions. In the result, the shortest segment P_iE_{j+2} is the radius of the smallest circle (Figure 3a), and it is used in further computation.

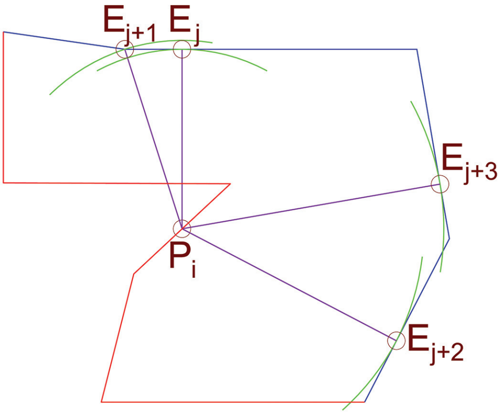
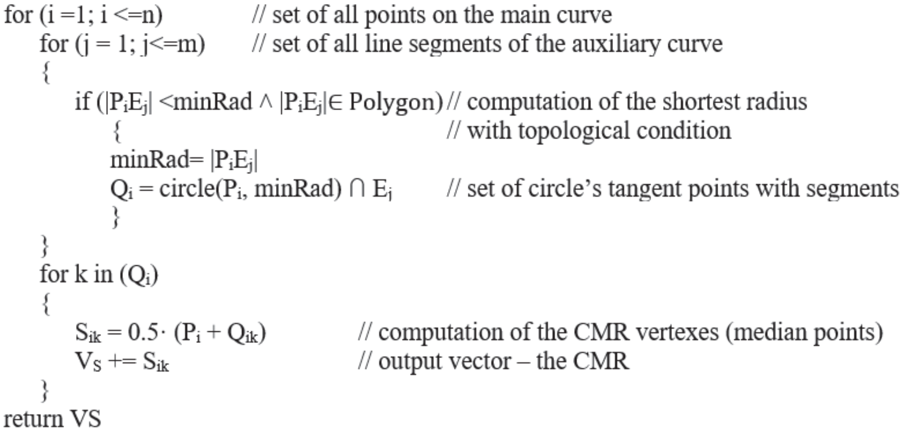


Fig. 4. Example of computing minimal radii to different segments from point P_i
(red – MC, blue – AC, green – part of circle, purple – radius)

The Point E_j of the shortest line segment P_iE_j becomes the end point Q_i of a radius connecting the main and the auxiliary curves (Figure 3b). The point S_i, the median point of the line segment P_iQ_j, is a new vertex of the CMR. The new line chain connecting consecutive points S_i is the final output curve (Figure 3c).

The described computation schema can be presented with a pseudo-code.



```
for (i =1; i <=n)          // set of all points on the main curve
    for (j = 1; j<=m)      // set of all line segments of the auxiliary curve
    {
        if (|$P_iE_j$| <minRad ∧ |$P_iE_j$|∈ Polygon)// computation of the shortest radius
            {                                     // with topological condition
            minRad= |$P_iE_j$|
            $Q_i$ = circle($P_i$, minRad) ∩ $E_j$       // set of circle's tangent points with segments
            }
    }
    for k in ($Q_i$)
    {
        $S_{ik}$ = 0.5· ($P_i$ + $Q_{ik}$)          // computation of the CMR vertexes (median points)
        $V_S$ += $S_{ik}$                      // output vector – the CMR
    }
return VS
```

In the analytical approach, the process is based on the computation of distances between the point P_i and all of the segments of the AC. Each segment is tested for the fulfillment of the topological condition: *Is the new line segment contained or covered by the input shape*? If none of the segments fulfill this condition, then the point is omitted and the next point P_i is tested.

On the auxiliary curve, tangent points Q_i of the circle with the center in point P_i and the radius equal to the length P_iQ_i are computed. In some particular cases, there will be more than one such point. That is why in the next step the median points of minimal radius are computed from the set of points. In the end, a new vector is constructed from the ordered median points. This vector is the result output object called the Curve of Minimal Radii.

4. Results

For the first step of computations, one has to determine the position of points P_i on the main curve. The simplest approach is to build the point set only with the break points of the main curve and continue to the next steps. However, this approach is not recommended. The CMR would not properly represent the real shape of the object. It could be caused by the low density of the input's vertexes or their irregular distribution. If there are too many vertexes on the MC, computations may take considerably longer without providing noticeably more adequate results. The recommended approach is to choose points in the fixed interval on the MC. It makes the algorithm resistant to the problems presented above.

In the fixed intervals approach, the direction of the MC has an influence on the CMR plots. Points P_i marked in the intervals counted from the beginning of the MC, may not overlap with points P'_i marked in the same intervals counted from the end of the MC. To eliminate this problem, one should use the commensurate intervals approach. For example, the distance between points may be counted by using the Equation 1.

$$d_i = \frac{L_{KG}}{s_{il}} = L_{KG} * s_{proc} \tag{1}$$

where:
d_i – interval between the consecutive points P_i,
L_{KG} – length of the main curve,
s_{il} – number of points P_i to be obtained on the main curve,
s_{proc} – proportion between the searched interval and the length of the main curve in percent.

The best results can be obtained by using commensurate intervals that are complemented with the main curve's break points. To avoid the influence of the MC disturbances, one can consider an arbitrary modification of the point set P_i through eliminating and adding points.

The CMR plot is resistant to disturbances of the auxiliary curve, but is strongly related to the shape of the main curve and the chosen interval approach. The curve plot inside an object can be modified by moving segments between the main and the auxiliary curves, altering the point set P_i and the interval, and changing initial conditions.

By the initial conditions, the author means the methods of the determination of end segments of the CRM on the ends of an input object. It is achieved by choosing the end vertexes position. The choice is dependent on the input shape and the purpose of output objects. Several conditions are proposed below. Depending on the situation, end vertexes may:

1. Overlap end points of the main curve,
2. Overlap median points of the segments between corresponding ends of the main and the auxiliary curves (if the curves are disjointed),
3. Overlap median points of the radii of circles with center points in the ends of the main curve (if they are calculated for the segments of the AC disjointed with the MC),
4. Fulfill other conditions determined by the user.

Elements on the left of Figure 5 present the differences in the plot of the CMR ends depending on the shape of the MC. For each example the first assumption was made – the CMR end points have to overlap end points of the main curve. Depending on the shape of the main curve, the end segments of the CMR have different positions. The second result (left, in the middle) is the closest to the anticipations about the symmetry axis. In this case the MC and the AC of similar length and shape were used.

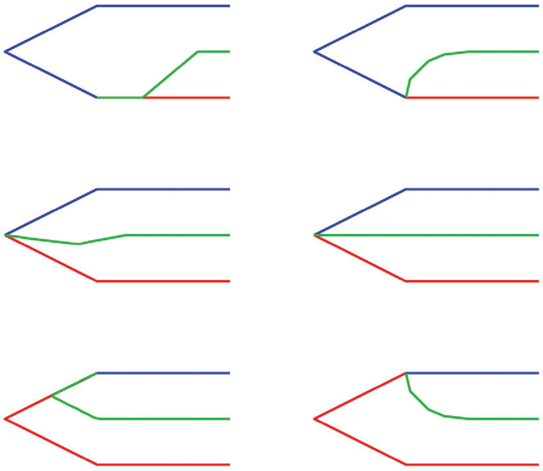


Fig. 5. Dependency between variants of the main curve and the shape of the CMR end segments for the pure algorithm (left side) and algorithm with initial conditions (red – MC, blue – AC, green – CMR)

If the input shape is a polygon with sharp ends, the first initial condition gives the best results. Additionally, for strongly irregular shapes, the influence of small disturbances decreases (Figure 6). The second and third conditions are designed to use with the input data containing two disjoined line chains.

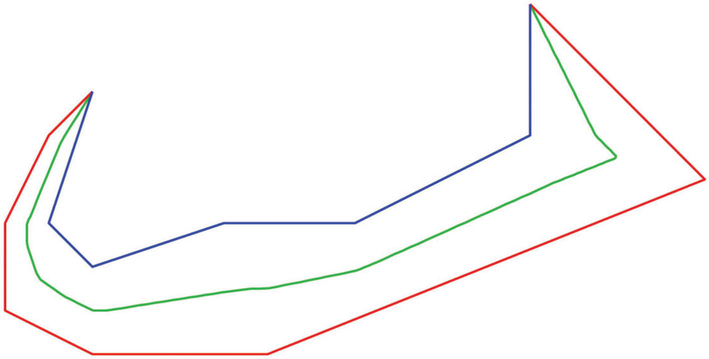


Fig. 6. The CMR computed for non-trivial object (red – MC, blue – AC, black – CMR)

Dealing with the end segments of the CMR is one of the biggest problems to solve. When the MC and initial conditions are selected incorrectly, the results would be far from perfect, as presented in Figure 5 (left). Methods of computing the first points of the CMR can be modified to improve results on their ends.

The problem with the end segment's disturbance was caused by the assumption that in order to compute the output object, the minimal radius should be used. For the points P_i on the first and on the last segment of the MC, the minimal radii overlap those segments. One of the recommended solutions to this problem is to choose points E_j on the MC, moved away from its beginning by the distance equal to the interval d_i, or proportionally to the length of the segments. Another way of solving this problem is to remove arbitrary first points P_i from the point set taken to calculations. Figure 5 depicts the results of an unmodified algorithm (left) and an algorithm using proportional intervals on the first segments of the MC and the AC (right).

Figure 7 depicts complicated objects with disturbances on both the MC and AC. The influence is visible on the CMR (black line). The altering of the point set P_i resulted in the local improvement of the results (green line). Due to the correlation between shapes of the CMR and the MC, the output object cannot be treated as a topological skeleton. However, the plot of the CMR for regular polygons is a sufficient approximation of a topological skeleton. Along with the increase of the distance from the ends, the influence of the initial conditions on the result decreases. In the middle part of an output object, the plot is the same for different initial conditions, and it is also similar to the results obtained from other algorithms. The final decision, which conditions should be used in computation, depends on the shape of the input object.

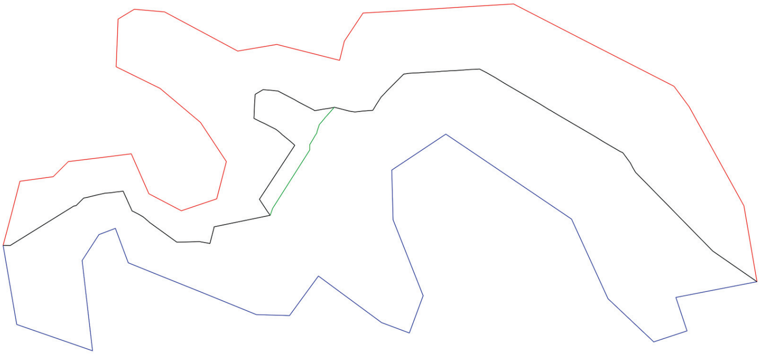


Fig. 7. CMR computer for complicated object (red – MC, blue – AC, black – CMR, green – CRM variant with Pi points elimination)

7. Discussion

The sample object (Figure 4) was used to compare results of the CMR algorithm with other methods. Figure 8 presents results of the CMR and three different algorithms: medial axis, straight skeleton, and Rhinoceros's mean curve. The medial axis and skeletonization results are branched objects, while the mean curve and CMR are single curve objects. If a single curve is required in further computation, the results of medial axis and skeletonization should be processed and reshaped. It would require assuming some additional conditions for an unambiguous output. On the other hand, single curve objects are useless in full description of an input object's shape.

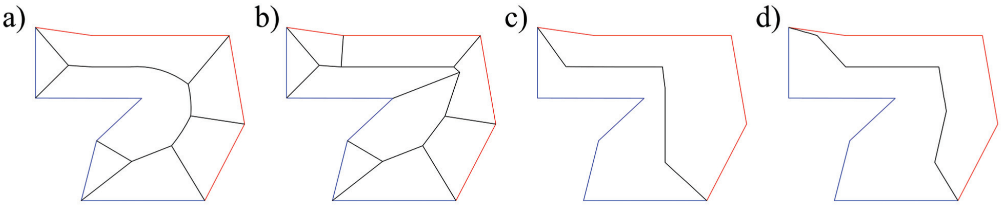


Fig. 8. Results of different methods: a) medial axis, b) straight skeleton, c) Rhinoceros's mean curve, d) CMR (red – MC, blue – AC, black – method's result)

Figure 9 compares the results of the used algorithms. In the middle part, the mean curve represents incorrectly the center of the input object, it lays too close to the AC. In the same part, medial axis and straight skeleton have different shapes. The CMR lays between them, and the regular part (upper part) is collinear or very closed to them. The ends of the medial axis and skeleton are dividing into a few segments and the CMR is collinear to one of them, and closed to the mean curve.

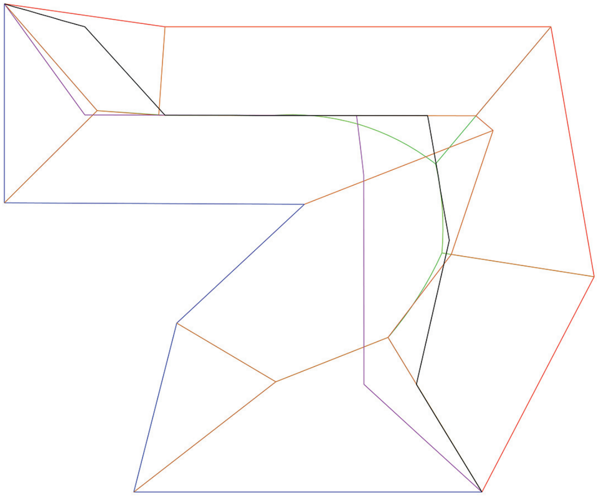


Fig. 9. Comparison of results of different methods (red – MC, blue – AC, black – CMR, green – medial axis, brown – straight skeleton, purple – Rhinoceros's mean curve)

The shape of single curve depends on the MC and AC. Picking them differently would change the general shape of the CMR and the mean curve. In addition, the CMR is influenced also by the initial conditions. From tested methods, the mean curve was the simplest one and the results don't represent the shape correctly. The CMR algorithm's results that are presented are similar to those from more advanced algorithms like medial axis and skeletonization. Thus, the algorithm should be chosen depending on the further purpose of the computed new curve.

The CMR algorithm was developed as a tool supporting hatching, defined in the amendment to the Polish law about the spatial database for topographical objects and utility infrastructure. For some objects on the map, a directional polyline (Polish: "polilinia kierunkowa") was introduced. It is defined as "an edge of an object parallel to the longitudinal symmetry axis of the object," and as "a direction of a conveyor or a movement direction of a gantry" (GESUT 2015, BDOT 2015). According to these definitions, the directional polyline can be:

- The main curve in the computation of the CMR, as long as it overlaps a part of a boundary,
- The CMR itself, if it lays within an object.

According to the law, the CAD programs have to offer a drawing tool for the directional polyline. The CMR algorithm could be an element of such a program's graphic engine, generating an automatic proposition of its plot. In this situation, the CAD user would only have to modify its plot in unambiguous situations.

The curve of minimal radii can be applied in the process of hatching of a map object, e.g. embankment. BDOT 2015 defines its directional polyline as one of the edges of this object. Modern spatial databases do not store hatch objects; hatching is generated automatically during the plotting of an object on the map. Except for the hatching pattern, the law does not specify any methods of computing its elements.

Nowadays, CAD programs use various methods in order to generate hatching. The biggest discrepancies appear with complicated and irregular geometrical objects. Graphical mistakes and mutual intersections may occur in the final result. Figure 10 depicts an improper hatching of a non-trivial embankment.

The author propounds the implementation of the presented algorithm as a solution to this problem. The CMR is always a curve without branches. Thus, it can be a basis for the generation of hatching elongated objects, such as embankments, gantries, or water gates. Additionally, the proper choice of the main curve improves the results for irregular objects. The main curve is defined during the drawing of an input object, or directly before the algorithm's calculations, as the object's directional polyline.

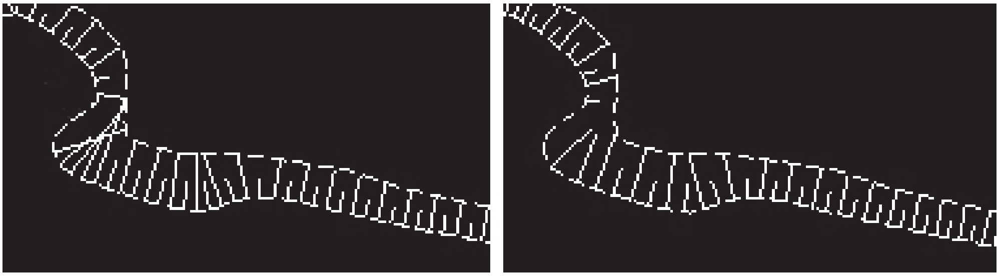


Fig. 10. Intersections of hatch lines (left), Hatch lines trimmed to the CMR (right)

Tests were conducted for the objects of a reinforced embankment (cartographic code BUBZ03_01). The symbol of embankment is constructed from four major elements: upper edge line, lower edge line, side edge line and hatching. Graphic representation defined by the law describes only regular cases with three types of edges. The side edges have lengths *a* and *c* (Figure 11). The distance between hatch lines *b* is equated from Equation 2:

$$b = \frac{a+c}{4} \tag{2}$$

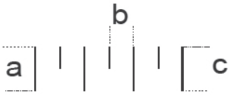


Fig. 11. Embankment hatching representation (BDOT 2015)

Additionally, every second hatch line length is as long as a half of the mean between *a* and *c*. Figure 6 depicts a case where the upper and lower edges are constrictive at the ends. According to Equation 2, lengths *a* and *c* should equal 0. It would make generating the hatching of such an object impossible.

The law only defines methods to compute regular cases. For various lengths *a* and *c*, determining the half lengths of hatch lines creates ambiguities. The author tested the CMR as a line chain to trim every second hatch line. Due to the possible occurrence of 0 length of *a* and *c*, distance *b* was counted separately for every hatch line from *a* and *c* determined locally. An additional condition of the minimal distance of *b* was made to preserve the readability of the map, and to avoid the infinite loop where length *b* approaches 0.

The CMR was used in the computation of embankment's hatching, according to the following convention. The object was divided into two line chains: the upper edge of embankment (the directional polyline) and the lower edge. The directional polyline becomes the MA. The second line chain becomes the AC. The CMR may be computed independently, or simultaneously with hatching lines. In the first option, the user has to define the interval between point P_i. In the second option, this interval can be equal to distance *b*. During the tests the first option was used. In order to calculate distance *b*, the previous value of *c* becomes *a*. If value *a* or *c* equals 0, or value *b* is smaller than the set constant, the minimal distance *b* is applied. The point of the MC moved from the previous point (or the first vertex of the MC) by interval *b,* is the first end of a hatch line. From this point the distance to the AC is measured. The line connecting the MC and the AC is a normal line and it becomes a hatch line. Every second line was trimmed to the CMR. The final result is presented in Figure 12. Topological conditions for the CMR ensure that the consecutive hatch lines have similar lengths.

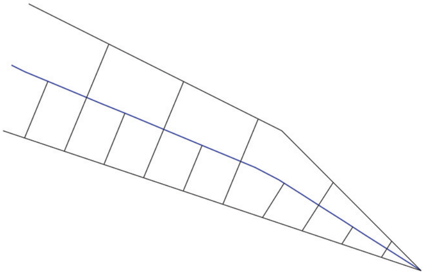


Fig. 12. Test result of hatching trimmed to CMR

The second proposed implementation is the usage of the CMR in hatching, depending on the symmetry axis of an object. The CMR may approximate this axis, especially for irregular objects, where the symmetry axes are not obvious. Due to the simplicity of the algorithm, it can successfully replace the symmetry axis for regular objects, such as gantries and conveyors. Figure 13 depicts the graphic representation of a conveyor (BDOT 2015). The center point of every point lays on the symmetry axis. Every second circle has to be tangent to both edges of a cartographic sign. The CMR algorithm fulfills the same criteria in the computation. Thus, with the interval between Pi approaching 0, it contains center points of all circles tangent to the MC. The definition of the CMR ensures that big circles never intersect the lines of the object, and the small circles lay in the middle between them.

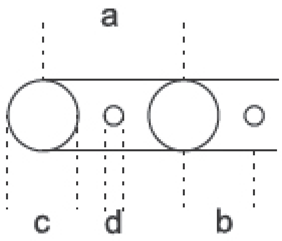


Fig. 13. Conveyor hatching representation (BDOT 2015)

Apart from the cartographic field of studies (2D objects), the presented algorithm was tested in the area of processing 3D data like point clouds. The survey evaluation of building health and geometrical features is made in the intersecting planes. Thus, the presented algorithm for 2D curve could be applied for objects like towers and chimneys.

To compute 2D curve, one has to intersect point cloud first, and generalize the ordered point sets into two planar curves (Figure 14a). Figure 14b presents points grouped into the MC and the AC with the tool called Average Depth, with a radius 0.01m from the CAD program Microstation. The computations were conducted in the coordinate system of the intersecting plane. As a result, the CMR approximated the symmetry axis of the object in the intersecting plane. To compute the end segments, the author chose the second presented initial condition – median points between the ends of two disjoined curves.

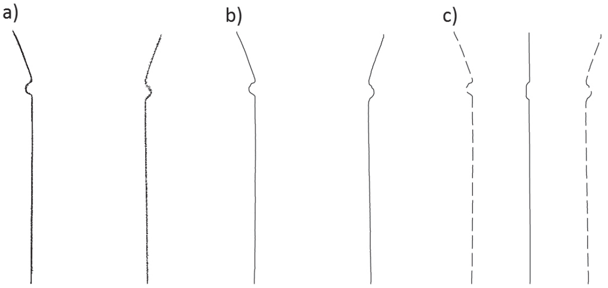


Fig. 14. a) Intersection of chimney point cloud, b) Curves generated from point cloud, c) computed CMR (solid line) (left dashed line – the MC, right dashed line – the AC)

The output object (Figure 14c) proves vulnerable to the disturbances of the main curve. In the bottom part of the point cloud, the plot of the CMR approximates the symmetry axis of the intersection, and the deviations are imperceptible. The disturbances in the top part, however, influence the plot of the CMR. The output curve converges to the MC in the disturbed segment.

To improve the results, the author propounds the arbitrary elimination of points P_i on the disturbed segment of the MC, or using generalizing and simplifying algorithms on the output object. Another way to resolve the problem of the disturbances, is to compute another CMR, replacing the MC and the AC with each other, and to make an averaged output from these two CMRs.

Figure 15 presents the top part of the chimney walls in two intersections, in directions X and Y. For each direction two CMR were computed, for each wall as the MC. In the lower part, the curves are almost the same. In the top part, CMR's shape is different. The two averaged outputs were computed with two methods: one as the Rhinoceros' mean curve and the second as curve connecting points computed as the middle points of two curves on the same height. As a result, new curves better approximate the real object's axis, but there are small differences between them.

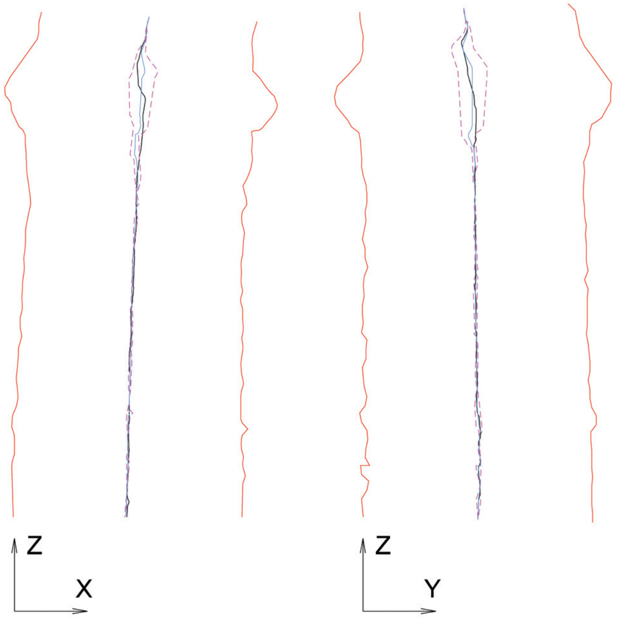


Fig. 15. The intersection of chimney's top (red line), with 2 CMR (dashed violet line), mean curve (black line) and curve from middle points (blue line)

Regardless of the choice of the presented methods, the final result is a better approximation of the symmetry axis in every part of intersection. But there is a need for further investigation and improvement of methods to average curves. More effort should be put on approximating the CMR in the end parts. It is also worth it to check the behavior of the algorithm for two curves in 3D.

Conclusions

The presented algorithm was designed to approximate the symmetry axis of geometric objects. It has a simple computation schema and the low number of steps needed to obtain results. As opposed to other algorithms, such as Medial Axis, there is no need to discretize the area of a polygon. It positively improves the efficiency of the automatic generation of axes for many cartographic objects at the same time.

One of the most important features, is the requirement that it possesses a directional polyline called the main curve. It is the basis for computations of the minimal radii. The implementation of the algorithm requires that the curve should be modifiable so the user could define it before the computations. The algorithm may use a directional polyline defined for other purposes for specified objects of spatial databases. To its important features, one can add the method used to compute the interval, the choice between setting initial conditions and modifying the P_i point set. The proper assortment of these elements improves the final result.

The topological conditions of the CMR ensure, that the output object is completely contained by the input object. However, the result cannot be equated with the topological skeleton due to the lack of branches. The comparison with different methods proved, that the CMR algorithm can replace more complicated algorithms. The output object is similar to, or can even overlap, the main branch of result objects obtained by using medial axis or skeletonization.

The conducted tests present the usefulness of the described algorithm in digital cartography. The author also presented the practical implementation prescribed by the law in force (BDOT 2015). The ways to improve the CMR results with other algorithms, e.g. the ones serving generalization, simplification and averaging of geometric objects, are worth researching. It may especially improve the results for point clouds. It may lead to the creation of the algorithm's variant for 3D objects.

Acknowledgment

This work has been carried out within the scientific statutory research programs no. 11.11.150.005 (the use of integrated systems and surveying methods to assess the technical condition and safety of engineering structures) at AGH University of Science and Technology at the Faculty of Mining Surveying and Environmental Engineering.

References

Aggarwal, A., Guibas, L. J., Saxe, J., and Shor, P. W. (1989). A linear-time algorithm for computing the Voronoi diagram of a convex polygon. *Discrete & Computational Geometry*, 4(6), 591–604. DOI: 10.1007/BF02187749

Aichholzer, O., Aurenhammer, F., Alberts, D., and Gärtner, B. (1996). A novel type of skeleton for polygons. In J. UCS The Journal of Universal Computer Science (pp. 752–761). Springer Berlin Heidelberg. DOI: 10.3217/jucs-001-12-0752

Aichholzer, O., and Aurenhammer, F. (1996, June). Straight skeletons for general polygonal figures in the plane. In International Computing and Combinatorics Conference (pp. 117–126). Springer Berlin Heidelberg. DOI: 10.1007/3-540-61332-3_144

Amenta, N., Choi, S., and Kolluri, R. K. (2001). The power crust, unions of balls, and the medial axis transform. *Computational Geometry*, 19(2), 127–153. Doi:10.1016/S0925-7721(01)00017-7

Bader, M., and Weibel, R. (1997, June). Detecting and resolving size and proximity conflicts in the generalization of polygonal maps. In Proceedings 18th International Cartographic Conference (Vol. 23, p. 27).

Bharatkumar, A. G., Daigle, K. E., Pandy, M. G., Cai, Q., and Aggarwal, J. K. (1994, November). Lower limb kinematics of human walking with the medial axis transformation. In Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on (pp. 70–76). doi: 10.1109/MNRAO.1994.346252

Blum, H. (1967). A transformation for extracting new descriptors of shape. In: W. Wathen-Dunn (ed), Models for the Perceptionof Speech and Visual Form (pp. 362–380), MIT Press, Cambridge, MA.

Brandt, J. W., Jain, A. K., and Algazi, V. R. (1991). Medial axis representation and encoding of scanned documents. *Journal of Visual Communication and Image Representation*, 2(2), 151–165. DOI: 10.1016/1047-3203(91)90005-Z

Brandt, J. W. (1994). Convergence and continuity criteria for discrete approximations of the continuous planar skeleton. CVGIP: *Image Understanding*, 59(1), 116–124. DOI: 10.1006/ciun.1994.1007

Bucksch, A., Lindenberg, R. C., and Menenti, M. (2009). Applications for point cloud skeletonizations in forestry and agriculture. *Reports on Geodesy*, 65-75.

Cherin, N., Cordier, F., and Melkemi, M. (2014). Modeling piecewise helix curves from 2D sketches. *Computer-Aided Design*, 46, 258–262. DOI: 10.1016/j.cad.2013.08.042

Chin, F., Snoeyink, J., and Wang, C. A. (1999). Finding the medial axis of a simple polygon in linear time. *Discrete & Computational Geometry*, 21(3), 405–420. DOI: 10.1007/PL00009429

Cordier, F., Melkemi, M., and Seo, H. (2016). Reconstruction of helices from their orthogonal projection. *Computer Aided Geometric Design*, 46, 1–15. DOI: 10.1016/j.cagd.2016.04.004

Culver, T., Keyser, J., and Manocha, D. (2004). Exact computation of the medial axis of a polyhedron. *Computer Aided Geometric Design*, 21(1), 65–98. DOI: 10.1016/j.cagd.2003.07.008

Dey, T. K., and Zhao, W. (2004). Approximate medial axis as a voronoi subcomplex. *Computer-Aided Design*, 36(2), 195-202. DOI: 10.1016/S0010-4485(03)00061-7

Djidjev, H., and Lingas, A. (1991, August). On computing the Voronoi diagram for restricted planar figures. In Workshop on Algorithms and Data Structures (pp. 54–64). Springer Berlin Heidelberg. DOI: 10.1007/BFb0028250

Dong, Z., Ting, Y., Xue, L., and Feng, A. (2015). Skeleton Extraction for Real Trees. *International Journal of Advancements in Computing Technology*, 7(4), 22.

Fogg, H. J., Armstrong, C. G., and Robinson, T. T. (2014). New techniques for enhanced medial axis based decompositions in 2-D. *Procedia Engineering*, 82, 162–174. DOI:10.1016/j.proeng.2014.10.381

Haunert, J. H., and Sester, M. (2008). Area collapse and road centerlines based on straight skeletons. *GeoInformatica*, 12(2), 169–191. DOI: 10.1007/s10707-007-0028-x

Kozioł, K., Szombara, S., and Knecht, J. (2012). Wyznaczenie punktów stałych obiektów przestrzennych na drodze automatycznej. *Archiwum Fotogrametrii, Kartografii i Teledetekcji*, 23.

Lähner, Z., Rodola, E., Schmidt, F. R., Bronstein, M. M., and Cremers, D. (2016). Efficient globally optimal 2d-to-3d deformable shape matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2185–2193). DOI: 10.1109/CVPR.2016.240

Lee, D. T. (1982). Medial axis transformation of a planar shape. IEEE Transactions on pattern analysis and machine intelligence, (4), 363–369. DOI: 10.1109/TPAMI.1982.4767267

Lee, I. K. (2000). Curve reconstruction from unorganized points. Computer aided geometric design, 17(2), 161-177. DOI: 10.1016/S0167-8396(99)00044-8

Lenda, G. (2006). Metody tworzenia i modyfikacji funkcji sklejanych na potrzeby opisu kształtu obiektów obserwowanych punktowo. Geodezja/Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, 12, 277–290.

Lenda, G., and Mirek, G. (2013). Parametrization of spline functions to describe the shape of shell structures. *Geomatics and Environmental Engineering*, 7(1), 65.

Loncaric, S. (1998). A survey of shape analysis techniques. *Pattern recognition*, 31(8), 983–1001. DOI: 10.1016/S0031-2023(97)00122-2

Pavlidis, T. (1980). Algorithms for shape analysis of contours and waveforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(4), 301–312. DOI:10.1109/TPAMI.1980.4767029

Piuze, E., Kry, P. G., and Siddiqi, K. (2011, April). Generalized helicoids for modeling hair geometry. *In Computer Graphics Forum* (Vol. 30, No. 2, pp. 247–256). Blackwell Publishing Ltd., DOI: 10.1111/j.1467-8659.2011.01856.x

GESUT 2015 – Rozporządzenie Ministra Administracji i Cyfryzacji z dnia 21 października 2015 r. w sprawie powiatowej bazy GESUT i krajowej bazy GESUT

BDOT 2015 – Rozporządzenie Ministra Administracji i Cyfryzacji z dnia 2 listopada 2015 r. w sprawie bazy danych obiektów topograficznych oraz mapy zasadniczej

Sarfraz, M., Samreen, S., and Hussain, M. Z. (2016, March). Modeling of 2D objects with weighted-Quadratic Trigonometric Spline. In Computer Graphics, Imaging and Visualization (CGiV), 2016 13th International Conference on (pp. 29–34). IEEE.

Shamos, M. I., and Hoey, D. (1975, October). Closest-point problems. In Foundations of Computer Science, 1975., 16th Annual Symposium on (pp. 151–162).

Szombara, S. (2013). Transformation of areal objects into linear objects, regarding the map scale. *Geoinformatica Polonica*, 12, 25–34, DOI: 10.2478/v10300-012-0010-5

Voronoï, G. (1908). Nouvelles applications des paramètres continus à la théorie des forms quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs. Journal für die reine und angewandte Mathematik, 134, 198–287.

Willcocks, C., Jackson, P., Nelson, C., and Obara, B. (2016). Extracting 3D Parametric Curves from 2D Images of Helical Objects. IEEE Transactions on Pattern Analysis and Machine Intelligence. DOI: 10.1109/TPAMI.2016.2613866