DOI: 10.4467/21995923GP.23.005.18604 GEOINFORMATICA POLONICA
22: 2023

Michał M. Buczek[1]
ORCID: 0000-0003-2584-8801

ITERATIVE APPROACH TO THE AREA COLLAPSE ALGORITHM FOR 2D GEOMETRIC OBJECTS REPRESENTING LONG ENGINEERING STRUCTURES

[1]AGH University of Krakow, Faculty of Mining Surveying and Environmental Engineering, Kraków, Poland
m.m.buczek.phd@gmail.com

Abstract

Nowadays the amount of gathered raw data emphasizes the importance of further data processing done by skilled engineers aided by computer algorithms. Researchers develop new algorithms for the automated determination of geometrical features, such as symmetry and main axes, skeleton lines, etc. This paper presented a new algorithm to compute an unbranched axis. It was based on the Curve of Minimal Radii (CMR) algorithm, and it overcomes its significant limitations depending on the shape of the input data. To define the accuracy of the results the threshold parameter was introduced. The described approach is more comprehensive than CMR in terms of the object shape. The tests were conducted on several planar objects, and the results were compared with the original CMR axes and Medial Axis.

Keywords: Main axis, polygon collapse, shape analysis, medial axis, geometry processing

ITERACYJNE PODEJŚCIE W ALGORTYMIE ŚCIENIANIA OBSZARÓW DLA OBIEKTÓW 2D REPREZENTUJĄCYCH WYDŁUŻONE BUDOWLE INŻYNIERSKIE

Abstrakt

Ze względu na rozmiar surowych danych pochodzących z pomiarów geodezyjnych szczególnego znaczenia nabiera ich dalsze przetwarzanie przez wyszkolonych inżynierów przy użyciu algorytmów komputerowych. Dlatego też naukowcy pracują nad kolejnymi algorytmami do automatyzacji procesów wykrywania cech geometrycznych takich jak symetria, osie główne, linie szkieletowe itp. W artykule przedstawiono autorski algorytm do wyznaczania nierozgałęzionej osi głównej. Opisywane rozwiązanie bazuje na algorytmie Krzywej Minimalnego Promienia (CMR) i eliminuje znaczące ograniczenia pierwowzoru dotyczące kształtu danych wejściowych. Użyty parametr progu iteracyjnego pozwala manipulować dokładnością wyników. Opisane rozwiązanie jest bardziej uniwersalne pod względem kształtu danych wejściowych niż algorytm CMR. Testy działania algorytmu przeprowadzono na obiektach dwuwymiarowych o zróżnicowanym kształcie, a otrzymane wyniki porównano z algorytmami CMR i Medial Axis.

Słowa kluczowe: oś główna, zapadanie wieloboku, analiza kształtu, medial axis, przetwarzanie geometryczne

1. INTRODUCTION

Nowadays gathering digital data for surveying purposes can be done in many ways. One can make measurements on-site using tacheometers, terrestrial and airborne laser scanning, or reuse existing analog documents after scanning, vectorizing, or digitalizing them. The introduction of more and more complex electronic instruments and more advanced software speeds up data acquisition. The application of total stations and LIDAR eliminated mundane, repetitive measurements, while they do not require as a high level of expertise and skills as in the past. The amount of gathered raw data emphasizes the importance of further data processing done by skilled engineers aided by computer algorithms.

Various measured objects, types of obtained data, and the final purposes of created products require a variety of computational approaches. In many cases, manual calculations are not reasonable, for example, in the process of fitting primitives into the point cloud. On the other hand, contemporary computer software is still imperfect, thus data processing still requires human assistance, and the generated results should be evaluated.

In some fields, like shape analysis, the empirical abilities of the human mind still outclass implemented methods and algorithms in the determination of geometrical and topological features and relations. The branch of science called shape analysis invents new solutions to solve such problems with more and more sophisticated algorithms. One of the fields of shape analysis research focuses on algorithms for the automated determination of geometrical features, such as symmetry and main axes, skeleton lines, edges, planes, and many others. A skeleton that captures the object's essential topology and shape information in a simple form is extremely useful in solving various problems [1, 2]. Surveyors might use a skeleton, or its subset called an axis, in the process of modeling point clouds of long constructions such as pipelines and chimneys.

In this paper, the author introduces a novel algorithm based on the previously presented algorithm *Curve of Minimal Radii* [3], shortly called *CMR*. The main goal of CMR is to calculate a polygonal chain approximating the mean axis of a 2D polygon. The described tests presented the worth of the CMR algorithm in the digital cartography and assessment of long engineering structures. The algorithm has significant limitations depending on the shape of the input data. In the initial research, the edges of all tested objects had similar shapes without asymmetry. If the edges differ a lot, or their shapes are changing quickly, the *CMR* algorithm may compute the axis closer to one of the edges. The goal of the presented new approach is to eliminate this problem. Tests were conducted on several theoretical cases of different shapes, and results were compared with the CMR and Medial Axis algorithms.

2. DIFFERENT APROACHES TO EXTRACTING THE AXIS

Skeletonization provides a representation of an input object by reducing its dimensionality to an axis or a skeleton while preserving the topologic and geometric properties of the object [4]. Such objects are widely used in computer sciences such as image analysis, graphics, computer-aided design, and others [1, 5–8].

Based on the overviewed curve-skeletonization algorithms Cornea, Silver [9] distilled a list of the skeleton's necessary properties such as homotopy, invariance under isometric transformations, centeredness, reliability, robustness, smoothness, efficiency, and others. Skeletons obtained with different methods may vary in terms of their properties such as shape or number of branches [1]. For some dedicated purposes, there might be a need to extract a single curve (axis) as an additional, final step.

Skeletonization algorithms might be grouped into three major categories: based on Voronoi diagrams and continuous geometric approaches of point clouds; based on the principle of the continuous evolution of object boundary curves; and based on the principle of digital morphological erosion or location of singularities on a digital distance transform [4]. Voronoi diagrams [10] were one of the first methods to compute the topological skeleton. The diagram divides a plane into regions based on the distance to points (seeds, Voronoi vertexes). Such regions (Voronoi cells) consist of all points of the plane closer to that seed than to any other. As a result, the diagram is approximating the topological skeleton. With infinite point density, it is equal to the Medial Axis [11].

Medial Axis (MA) is a widely recognized method of computing an object's skeleton. It finds points equidistant from at least two edges of a polygon [12]. The MA can represent a lossless shape descriptor, but it is difficult and expensive to compute accurately in 3D and higher dimensional spaces [13]. Blum used a comparison to the grass fire to present the idea behind this solution. If 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 [2, 12, 14].

Aichholzer, Aurenhammer [15] presented an alternative approach to computing skeleton. They have proposed the Straight Skeleton, where the polygon's boundary is divided into single lines called wavefronts. In the shrinking process, the waveforms are propagated inside the object. The skeleton edges are the bisector of two wavefronts. As a result, the output shape looks like a rooftop [16].

Eftekharian and Ilieş [17] proposed a novel approach to construct exact or approximate distance functions and the associated skeletal representations called C-skeleton. They used a closed subset of the semi-analytic domain Ω as an input. The subset is divided into primitive halfspaces described with R-functions [18]. Based on them they defined the skeleton's distance functions. The approximate distance function is transformed into an exact distance function by adding conical and trimming halfspaces at every concave vertex of the domain. The constructed C-skeleton is piecewise linear for polygons and closely resembles the medial axis of a planar domain but has a lower geometric complexity.

Aigner, Aurenhammer [19] proposed the method called *triangulation axis,* which has much fewer edges and branches than the Medial Axis or the Straight Skeleton. A simple polygon P is divided into triangles constructed out of P vertices. Then, all triangles are categorized into three types: ear triangles, link triangles, and branch triangles – having one, two, or three sides that are diagonals of *P*. They defined a specific line segment for each triangle type that contributes to the final axis. For example, if the triangle is a link triangle, then the new line segment connects the midpoints of the two bounding diagonals of *P*. The triangulation axis also allows for the reconstruction initial polygon *P* out of it.

For three-dimensional objects, there is a need to create a curvilinear (a one-dimensional construct) representation of a three-dimensional shape. This necessity came from the fact that the classical Medial Axis generally produces two-manifold elements when applied to three-dimensional shapes [20]. Dey and Sun [21] introduced a mathematical definition to approximate curve-skeletons of 3D shapes with connected manifold boundary. According to their definition, the curve-skeleton should be in the 'middle' of the shape it is natural to define it as some subset of the medial axis. They proposed a function called the medial geodesic function to transform medial axis planes into curves. It is based on the geodesic distances between points where the maximal balls defining the medial axis touch the shape boundary.

The new approaches to extracting axes are still presented due to the lack of one, universal method applicable for an object of any shape at any dimension. This paper presents an improvement of the CMR algorithm [3] designed for 2D geometrical objects representing long engineering structures. The goal of the presented new approach is to eliminate the limitations of the original algorithm and make its results more comprehensive and accurate.

3. THE CURVE OF MINIMAL RADII

The algorithm Curve of Minimal Radii is similar to the Medial Axis [3]. In both approaches, all vertexes of the resulting axis are equidistant to at least two elements of the input shape's boundary, but the method to determine their positions differs. The CMR algorithm

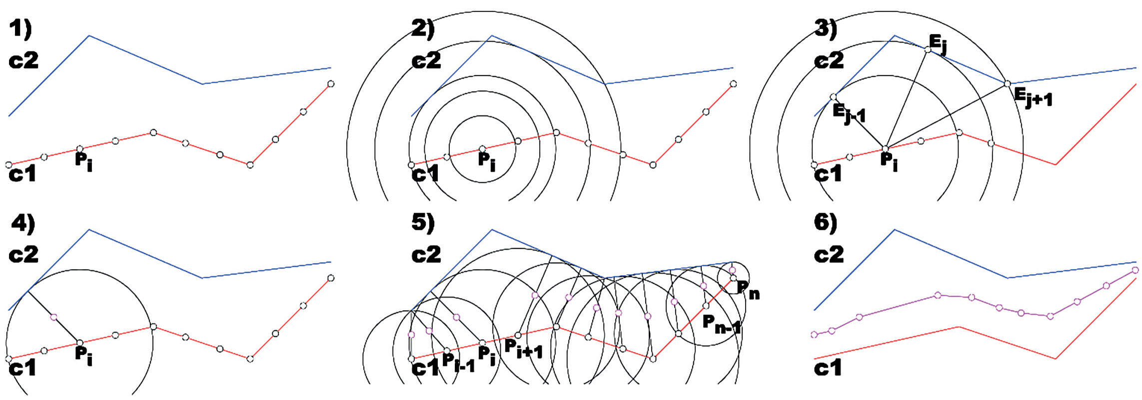

Fig. 1. Following steps to compute the CMR curve
Ryc. 1. Kolejne kroki w celu wyznaczenia krzywej CMR

requires the input data in the form of two approximately even and similar line chains. Consecutive CMR's vertices are computed using points on one of those polygon chains, called *c1*. The second polygon chain, called *c2*, is used to determine the position of the computed vertex. In the original paper [3] these two curves were also called a Main Curve, and an Auxiliary Curve.

Figure 1 depicts the next phases of the computation process. If the input data contains two disjoint polygon chains, one of them becomes *c1*, and the other becomes *c2* (Fig. 1.1). If the input shape is a closed polygon, it should be divided into two continuous parts. There cannot be a situation in which the boundary's segment belongs to none of these two curves, or belongs to both of them at the same time [3].

The algorithm works on the pointset *P* defined on the curve *c1* built out of the curve's vertices and optional equidistant points (in fixed intervals). A point P_i from such a pointset is considered as a center point of a circle of the unknown radius (Fig. 1.2). The points of tangency of these circles with curve *c2* create a pointset *E* (Fig. 1.3). The goal is to find the shortest segment P_i-E_j, which doesn't intersect curves *c1* and *c2*. One could interpret this segment as the shortest radius from the tangent circles to the curve *c2* (Fig. 1.4). The middle point of the found „minimal" radius is a vertex of the constructed axis. If the algorithm isn't able to find any segment fulfilling the condition of the lack of intersections, the point P_i is skipped, and computations start over for the next point in the set *P*. The procedure is repeated for all points from the set P (Fig. 1.5), and all computed vertices form an axis of the input object (Fig. 1.6).

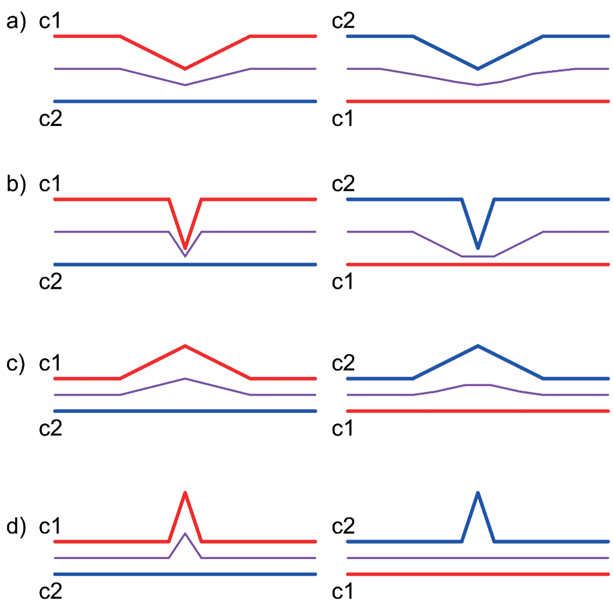

Fig. 2. Different shapes of axes depending on choosing the c1 curve

Ryc. 2. Różne kształty osi w zależności od wyboru krzywej *c1*

4. ITERATIVE APPROACH

The main weakness of the CMR algorithm is caused by using computation points picked only from one of the curves. The arbitrary definition of curves *c1* and *c2* determines the shape of the output axis. If there is a significant difference in shape between them, a computed axis may vary depending on the definition of a curve *c1*. This problem was mentioned in the original paper without providing any satisfying solution [3].

Figure 2 depicts four objects with two edges each, where one curve is a straight line, and a second is not. On the left side, axes are computed using upper edges, while on the right, axes are computed with straight lines. Differences in shape might be observed in each pair of axes around the sharp segment of the upper curve. One of the axes of the last object (Fig. 2d) is even a straight line. This should be interpreted as incorrect output due to the not-straight shape of the upper curve. The presented problem might be interpreted as a strong dependency on curve *c1* and a weak dependency on curve *c2*. Assuming each of the found axes lay closer to one of the edges, the 'real' axis is somewhere between them.

All CMR axes lay between curves *c1* and *c2*. This is guaranteed by the way the CMR algorithm finds the following axis's points. Based on this assumption an iterative approach was proposed. In the beginning, the object is divided into curves *c1* and *c2* (Fig. 3.1). Then two CMR axes are computed, the first one using *c1*, and the second using *c2*. These two axes then are treated as new object's edges and become new *c1* and *c2* (Fig. 3.2). In the following repetitions, two outputs become new edges used to compute the next CMR axes. Because each pair of axes lay between curves *c1* and *c2* used to compute them, after each iteration they approach closer to each other (Fig. 3.3).

At some point, a pair of axes is almost overlaying each other. To interpret its' similarity one may compute the maximal distance between axes. If the maximal distance between two axes is smaller than value *x*, one can assume that the final two axes are the same (Fig 3.4). Therefore, this value can be used as a threshold to stop the iterative process.

5. DISCUSSION

The described iterative approach leads to finding a single curve and eliminates the CMR's unequal dependency on the edges. The new algorithm is called the Iterative Curve of Minimal Radii (iCMR). Figure 4 presents axes computed with algorithms CMR (two variants), iCMR, and Medial Axis algorithms. A subset of Straight Skeleton is used as a reference axis for comparison. The Straight Skeleton algorithm was chosen because its results resemble the input edges. Both CMR and iCMR generated axes without branches, which is considered a feature of these algorithms. The Medial Axis generated a full skeleton. Therefore, the usage of

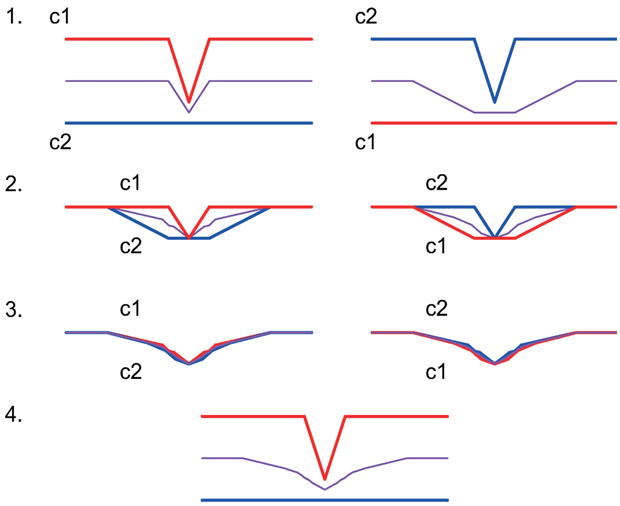

Fig. 3. Following steps of the iCMR algorithm: 1. first iteration, 2. second iteration, 3. final iteration, 4. iCMR axis
Ryc. 3. Kolejne kroki algorytmu iCMR: 1. pierwsza iteracja, 2. druga iteracja, 3. ostatnia iteracja, 4. oś iCMR

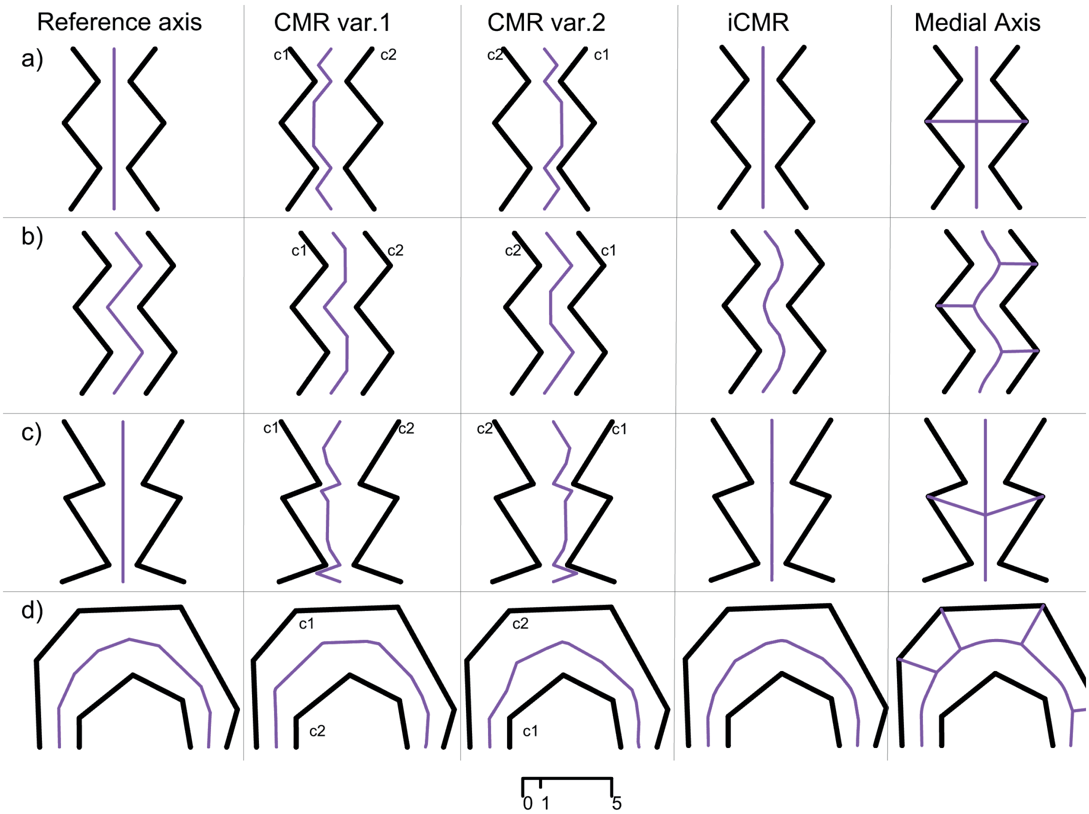

Fig. 4. The comparison of results of CMR, iCMR, Medial Axis algorithms, and reference axis generated with the Straight Skeleton algorithm in the dimensionless scale
Ryc. 4. Porównanie wyników algorytmów CMR, iCMR, Medial Axis i osi referencyjnych wygenerowanych algorytmem Straight Skeleton w skali niemianowanej

it in similar applications would require a method to extract a single curve out of it.

All CMR axes were computed out of edges marked as *c1*. The pointsets used in computations contained vertices of *c1* and equidistant points on it (0.1 unit). In cases *a-c* the strong dependency on *c1* is visible. This problem does not occur in iCMR axes of the same objects. In variant *d,* both CMR and iCMR axes are placed approximately in between the edges. The biggest differences between CMR axes can be observed on the left side, where axes lay closer to *c1* edges. Once again, the iCMR curve minimalizes this unwanted dependency on *c1*. In addition, CMR axes don't resemble accurately `the shapes and sharp edges of the edges *c1*.

In all cases Medial Axes lay exactly in the middle between the edges, therefore it is considered as a desired outcome. Only in the last variant, CMR axes resemble the Medial Axis, in all other cases shapes of the axes are completely different. On the other hand, all iCMR axes are similar to the part of Medial Axes. In variants *b* and *d* iCMR, axes lay slightly closer to the sharp edges of the edges than the Medial Axis. These two axes don't accurately resemble all sharp edges on the case b, contrary to the reference axes.

The iCMR algorithm requires setting a threshold value. It may be defined by the further purpose of the axis. For example, if the axis is computed out of data with millimeter precision, a threshold value should be at least at the submillimeter level. Another way, used in this study, is to set a threshold value depending on the size of the initial object. The parameter was set as 0.25‰ of the minimal width between edges. All iCMR axes were computed within 5–10 iterations for so defined condition.

Table 1. The average distance between computed axes and reference axes, and between vertices of c1 and c2 in dimensionless scale

Tabela 1. Średnie odległości pomiędzy obliczonymi osiami i osiami referencyjnymi oraz pomiędzy wierzchołkami krawędzi c1 i c2 w skali niemianowanej

	CMR var.1	CMR var.2	iCMR	Medial Axis	Avg. vertices dist.
a)	0.57	0.57	0.00	0.00	3.69
b)	0.12	0.12	0.04	0.05	3.61
c)	0.58	0.58	0.00	0.00	4.74
d)	0.08	0.19	0.08	0.01	3.47

Picking a threshold value greatly impacts the algorithm's results and performance. Too small value will result in many unnecessary repetitions, where the maximal distance between axes is far smaller than needed. On the contrary, if the value is too big, the curve picked as an iCMR's result won't be similar enough to the second computed curve, therefore the algorithm couldn't be considered reliable.

Another limitation might be caused by the way of picking points on the edge to compute the next steps. In the CMR algorithm, pointset *P* is built out of the curve's vertices and the optional equidistant points might be added. A number of elements in *P* define a maximal number of CMR's vertices. In the iterative approach adding points in the fixed distances in each repetition might cause a meaningful increase of points to conduct computations, which would slow down the whole process. This problem requires deepened research.

The accuracy of tested algorithms was assessed in comparison with the reference axes. Table 1 presents the average distance between each pair of axes. In addition, the average distances between vertices of *c1* and *c2* are provided. All values are dimensionless because the tests were conducted in CAD software, and their main goal is to present the scale of differences. The distance between the edges of the input objects varies between 1.6 and 6.8 units.

In all cases, iCMR axes lay closer to the reference axes than CMR axes. The biggest improvement was obtained in cases *a* and *c* with symmetrical edges, where both results of iCMR and Medial Axis algorithms overlay the reference axes. In cases, *b* and *d* differences in values between them, caused by the different way of computing the axis, are negligible in comparison with the size of the input objects. The obtained average distances prove that the iCMR algorithm overcomes the CMR algorithm.

6. CONCLUSIONS

The presented algorithm was designed based on the Curve of the Minimal Radii. It computes axes from both edges, therefore it eliminates the CMR's main problem of strong dependency on only one of them. The iCMR algorithm is based on the CMR algorithm, thus all topological conditions defined for CMR are still met. Both computed axes are completely contained by the input object which allowed an iterative approach in the first place.

Picking the threshold value allows a user to obtain an axis on the desired level of accuracy. Computations are done in the recursions until the required level of accuracy is obtained. On the contrary to the Medial Axis, iCMR computes an axis without branches, which eliminates the need to extract the axis out of the skeleton. The CMR algorithm was designed to generate axes only of long engineering structures, but the new iterative approach is more accurate and comprehensive in terms of the object shape. It might be used in any field, where 2D axes without branches are required.

REFERENCES

1. Lakshmi, J.K.; Punithavalli, M. A Survey on Skeletons in Digital Image Processing. in 2009 International Conference on Digital Image Processing, 2009.
2. Tagliasacchi, A.; Delame T.; Spagnuolo, M.; Amenta, N.; Telea, A. 3D skeletons: A state-of-the-art report. in Computer Graphics Forum. Wiley Online Library, 2016.
3. Buczek, M.M. Area collapse algorithm computing new curve of 2D geometric objects. Geodesy and Cartography, 66(1), 2017, pp. 23–43.
4. Saha, P.K.; Borgefors G.; di Baja, G.S. A survey on skeletonization algorithms and their applications. Pattern recognition letters, 76, 2016, pp. 3–12.
5. Siddiqi, K.; Pizer, S. Medial representations: mathematics, algorithms and applications. Vol. 37. Springer Science & Business Media, 2008.
6. Whited, B.; Rossignac J. Ball-Morph: Definition, Implementation, and Comparative Evaluation. IEEE Transactions on Visualization and Computer Graphics, 17(6), 2011, pp. 757–769.
7. Yuan, J., Wang D.; Wu B.; Yan L.; Li R. Automatic road extraction from satellite imagery using LEGION networks. in 2009 International Joint Conference on Neural Networks, 2009.
8. Makem, J.E.; Fogg, H.J.; Mukherjee, N. Automatic feature recognition using the Medial Axis for structured meshing of automotive body panels. Computer-Aided Design, 124, 2020, pp. 102845.
9. Cornea, N.D.; Silver, D.; Min, P. Curve-skeleton properties, applications, and algorithms. IEEE Transactions on visualization and computer graphics, 13(3), 2007, pp. 530–548.
10. Voronoi, G. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites. Journal für die reine und angewandte Mathematik, 133, 1908, pp. 97–178.
11. Brandt, J.W. Convergence and continuity criteria for discrete approximations of the continuous planar skeleton. CVGIP: Image Understanding, 59(1), 1994, pp. 116–124.
12. Blum, H. A transformation for extracting descriptors of shape, 1967.
13. Culver, T.; Keyser, J.; Manocha, D. Exact computation of the medial axis of a polyhedron. Computer Aided Geometric Design, 21(1), 2004, pp. 65–98.
14. Chin, F., Snoeyink, J., Wang. C.A. Finding the medial axis of a simple polygon in linear time. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995.
15. Aichholzer, O., Aurenhammer, F.; Alberts D.; Gärtner, B. A Novel Type of Skeleton for Polygons, in J.UCS The Journal of Universal Computer Science: Annual Print and CD-ROM Archive Edition Volume 1, Springer Berlin Heidelberg: Berlin, Heidelberg, 1995, pp. 752–761.
16. Aichholzer, O.; Aurenhammer F. Straight skeletons for general polygonal figures in the plane. in International computing and combinatorics conference, Springer, 1996.
17. Eftekharian, A.A.; Ilieş, H.T. Distance functions and skeletal representations of rigid and non-rigid planar shapes. Computer-Aided Design, 41(12), 2009, pp. 865–876.
18. Shapiro, V. Semi-analytic geometry with R-functions. Acta Numerica, 16, 2007, pp. 239–303.
19. Aigner, W.; Aurenhammer F.; Jüttler, B. On triangulation axes of polygons. Information Processing Letters, 115(1), 2015, pp. 45–51.
20. Tagliasacchi, A. Skeletal representations and applications. arXiv preprint arXiv:1301.6809, 2013.
21. Dey, T.K.; Sun, J. Defining and computing curve-skeletons with medial geodesic function. in Symposium on geometry processing, 2006.