UOIuBIH
ORSinBIH
Operations Research Society in
Bosnia and Herzegovina

Southeast Europe Journal of Soft Computing

Available online: www.scjournal.com.ba

COMPUTATIONAL GEOMETRY APPLICATIONS

Aybeyan Selimi[1], Muzafer Saračević[2]

[1]International Vision University, Gostivar, Macedonia; aybeyan@vizyon.edu.mk
[2]Department of computer sciences, University of Novi Pazar, Serbia, muzafers@uninp.edu.rs

Article Info

Article history:
Received 27 May, 2018
Received in revised form August 2,2018

Keywords:
Computer graphics,
triangulation of polygon,
linear optimization,
point location and motion planning.

Abstract

Computational geometry is an integral part of mathematics and computer science deals with the algorithmic solution of geometry problems. From the beginning to today, computer geometry links different areas of science and techniques, such as the theory of algorithms, combinatorial and Euclidean geometry, but including data structures and optimization. Today, computational geometry has a great deal of application in computer graphics, geometric modeling, computer vision, and geodesic path, motion planning and parallel computing. The complex calculations and theories in the field of geometry are long time studied and developed, but from the aspect of application in modern information technologies they still are in the beginning. In this research is given the applications of computational geometry in polygon triangulation, manufacturing of objects with molds, point location, and robot motion planning.

1. INTRODUCTION

The place and importance of mathematics in the development of science are very large. Arithmetic and geometry are two basic areas that form the basis of mathematics.

Geometry with considering some rules examines shapes that can be designed in the plane and in space. The etymological word "*geometry*" means the measurement of the earth. Geometry exists since ancient times. However, the name of the geometry began to be used by the Ancient Greeks, when geometry as a discipline became systematic. The original goal of geometry was to study shapes in a plane and in a space. Although the shapes in nature can be derived from specific objects, geometry has left the use of experimental methods very early. On the contrary, she tried to reduce real objects to the ideal form for testing.

Calculating the surface, determining the orbits of celestial bodies, the ratio in geographical maps, machine building, and architecture are some of the areas in which geometry is applied. The first geometries due to their nature were created intuitively. These geometries are mostly visual. The second phase in the development of geometry began with the measurement of the shape that developed in Egypt, Mesopotamia and Ancient Greece [32]. The development of geometry continues with Classical Indian geometry, Chinese geometry, Islamic golden age geometry, Analytic geometry, Non-Euclidean geometry to Modern geometry [33]. In the second half of the 20th century by the appearance of computers, a new discipline is developed, called computational geometry.

Computational Geometry is a discipline of computer science that deals with the research of algorithms from the aspect of geometry. The first use of the term "Computational Geometry" dates back to 1975 and was used by M. I. Shamos [29]. The subject of research of the computational geometry is to solve geometric problems with the help of computers whereas the resultant of these research geometric algorithms that have wide application in various fields are obtained. Although a computational geometry has made significant progress today, it represents one of the early geometric areas that has been used to solve practical problems of everyday life of people in the earliest times of humanity.

Algorithms of computational geometry contain a large number of points. These algorithms are encountered in very large datasets and are therefore of great importance in practical terms. The difference between these data sets is expressed as the difference $O(n^2)$ and $O(n\ logn)$. The goal of all researchs in computational geometry is to minimize the value of this difference.

At the center of development of computational geometry as a discipline, are the problems of geometry that were present in computer graphics, computer-aided design, and manufacturing (CAD / CAM). Problems computational geometry basically are classical geometric problems that have been added with mathematical visualization and modeling in the computer science. Other important applications of computational geometry are seen in robotics (motion planning and visibility problems), in geographic information systems (geometric locations and search, motion planning), integrated circuit design, computer-aided engineering (CAE) (Mesh generation), computer vision.

2. Disciplines of Computational Geometry

Computational geometry as a discipline with large domain today has the following sub-disciplines.

Combinatorial computational geometry, also called algorithmic geometry is the sub-disciple of computational geometry which considers geometric objects as separate objects.

Numerical computational geometry deals with the design of numerically robust algorithms for solving geometric problems [9], [17]. It has several different approaches to solve geometric problems. Edelsbrunner and Husk described the technique they designated as simulations of the simplicity of geometric algorithms [10].

Geometric modeling refers to the process of creating geometric models of real objects or dynamic processes that can be stored in a computer for the purpose of designing (CAD / CAM) or for simulation processes. There are various sub-problems in geometric modeling [19]. One of the most important problems in geometric modeling is to automatically generate a mesh inside a polygon. This is also a basic and necessary tool for solving the system of partial differential equations with the finite element method.

A *computer vision* analyzes the scene in the real world using an input device that is usually a kind of transducer (digital camera) and to arrive at a description of a scene that is useful for accomplishing of some task [30], [16].

Motion planning, linkages, and automated assembly are three problems of *robotics* in which computational geometry is applied. The problem of motion planning involves modeling robots as a polygon in two dimensions or a polyhedron in three dimensions that can move in space between an obstacle collection [3], [4], [30].

Art Gallery Theorems and Algorithms is a discipline that deals with the problems of viewing the walls of a large and complex gallery that is not in the shape of a star. The solution to this problem was given by the Canadian-Czech mathematician Václav Chvátal, who is known as "Chvátal's Art Gallery Theorem", also known as "Watchman Theorem" [6]. Avis and Toussaint in 1981 developed an efficient algorithm for finding locations where cameras should be mounted in the gallery's walls [5].

Computer graphics is an area where computational geometry provides faster algorithms for solving problems that are associated with hidden lines and hidden surface removal problems [7], [13], [29].

Geodesic computational geometry is an area where instead of Euclidean measure is used geodesic measure, i.e., the length of the shortest path (geodesic path) between two points that avoids the obstacles [11], [12].

Dynamic computational geometry is an area that deals with calculations in the convex hull with n points and the additional point *p* in the plane. The determination of a new convex hull with $n + 1$ points is done by modifying the existing convex hull of *n* points. First, it is tested whether the new point *p* lies in a convex hull of *n* points. If point p belongs to the convex hull of *n* points, then the obtained convex hull with $n + 1$ points is again convex and other calculations are not required. But first of all, in this process is need what's possible faster determine whether new point *p* belongs to the interior of the convex polygon with *n* sides. Determining whether the point *p* is in the interior of the convex polygon with *n* sides is given with algorithm of O (*n*) computational complexity. Is possible to make this procedure much better? The answer is YES. This is done by storing the previous processing of the convex polygon in the corresponding data structure [29]. These data structures are called a dynamic data structure. The latest results on data structures for computational geometry are given in [15].

Parallel computational geometry is an area in which parallel computers are used. Parallel computers must communicate with each other in the parallel machine all time they use to find a solution to the geometric problem [2]. Instead of using one computer where primitive operations are performed one after the other, parallel computational geometry use a set of k computers, all of which work on the same problem at the same time. Each computer solves only part of the problem, and computers communicate with one another so as not to duplicate the work and "*stitch*" the partial solutions obtained from each computer into a complete solution. Algorithms used by parallel computers are different from those used by sequential computers. Such algorithms are called parallel algorithms. An algorithm of primitive operations for finding a convex hull of n points have complexity O (*n*3) unit of time. If O (*n*3) computers with simple processors are used as a massively parallel computer then the convex hull of n points can be calculated in one unit of time [1]. The neural networks and optical computer geometry are domains of application of the parallel computational geometry [18]. The analysis of the complexity of optical algorithms is done with the appropriate definition of optical primitives [14].

Isothetic computational geometry or rectilinear calculus geometry is an area of computational geometry that deals with input data such as line segments and polygons in which all edges are vertical or horizontal [31].

3. CONCRETE APPLICATION EXAMPLES OF COMPUTATIONAL GEOMETRY

Triangulation of polygon

Polygon triangulation is historically very old problem who led it to the discovery of Catalan numbers. When determining all triangulation of the polygon, must be considered the polygon shape (convexity or concavity). This makes the problem solution very difficult. The problem can be reduced by limiting ourselves to the convex polygons. In the convex polygons, all diagonals are internal diagonals. In this case, the number of triangulations of the convex polygon is independent of the form and can be uniquely characterized by the number of nodes *n*. Under the triangulation of a convex polygon, we mean decomposing the interior of the polygon to triangles by non-crossing interconnected diagonals. In this problem, the number of triangulations is actually considered, where is possible the maximum partition of the convex polygon into *n* - 2 triangles [21]. It is important to note that there is a correlation between the triangulation of the polygon and the decomposition of Catalan numbers see [27].

Block method is the process of generating the triangulation of a polygon using a set of triangles. The general strategy used in the Block method is to break the main problem into fewer problems that are mutually dependent. Each problem is solved only once and is used several times, thus avoiding unnecessary repetition of the same calculations and is suitable for recursion with memoization. This allows convenience in the practical application of this method. Based on the experimental results of paper [28], from examining the Java application we see that the average execution time at one level in the Hurtado-Noy algorithm is 46.16 while for the block method is 10.74. All this significantly affects the load on the memory during execution, and the speed of generating triangulation.

In the paper [24] was processed, a triangulation from the aspect of notation and their storage. These methods have been presented with the aim of saving memory space. A connection has been made between polygon triangulation problems and combinatorial problems, such as the ballot problem and the lattice path. The first notation technique is called *ballot notation*. The second notation is called *alfa-numeric* (AN). The experimental results obtained in this paper show that the use of these notations achieves significant memory savings in the process of storage of triangulations. In this paper, has been introduced the possibility of using a stack structure in the storage of polygon triangulation.

Polygon triangulation also is a complex problem that requires complex classes for efficient object-oriented implementation. Object-Oriented Analysis and Design (OOAD) provide a comprehensive insight into the implementation of triangulation problems. The paper [25] gives approaches to the OOAD methodology for the polygon triangulation problem. The problem of triangulation in this paper is analyzed from three aspects, where each approach is defined by the appropriate type of engineering of this methodology: forward engineering, reverse engineering, and round-trip engineering. An important result of this paper is an improvement achieved by applying the reverse engineering and synchronization of the UML model and Java source code. There are also some advantages for all three approaches, with particular emphasis on improving implementation by applying synchronization in forwarding engineering and reverse engineering approaches.

Triangulation as an important aspect of computational geometry is also a concept that is applied in computer graphics. Basically, the polygon triangulation in computer graphics provides a three-dimensional view of images from a set of points. One of the basic goals of applying triangulation is what is used as a replacement for the storage of nodes and the distribution of internal polygon diagonals. Efficiency is achieved at the rate of triangulation recording in the Java buffer (cache) used to store interim results [22].

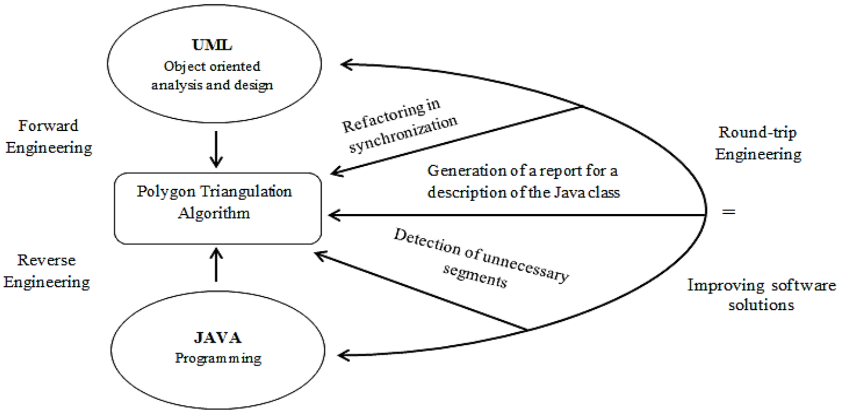
UML
Object oriented analysis and design
Forward Engineering
Polygon Triangulation Algorithm
Reverse Engineering
JAVA
Programming
Refactoring in synchronization
Generation of a report for a description of the Java class
Detection of unnecessary segments
Round-trip Engineering
=
Improving software solutions

Figure 1. Three object-oriented analysis and design approaches [21]

The process of recording convex polygonal triangulations is related to binary trees that in the computer sciences represent the structure for data storage. In the paper [22], a Java implementation of the Lukasiewicz algorithm for triangulation of polygon based on binary trees is shown. This application enables efficient graphic representation of triangulation. The result of the application is a binary notation of a triangulation of a convex polygon that can be used to detect some regularities in generating triangulations of larger polygons.

Triangulation as a procedure can be used in the presentation of a three-dimensional object from a set of points and provides a mechanism for the so-called "glazing" of these three-dimensional objects. As a procedure it is very important for speed in computer graphics, it provides quality and good resolution for the object. Triangulation of convex polygons is a question that arises in two-dimensional computational geometry [23].

One of the good software solutions for triangulation is the optional Final Surface plug-in triangulation that offers the possibility of generating network triangles from cloud points whose application in the paper [23] has been shown in the reconstruction of the human brain. The alignment of the brain surface is determined by the correspondence of 3D points between pairs in the surface. This algorithm is based on geodetic distance combinations and curvature of the surface see Figure 2.

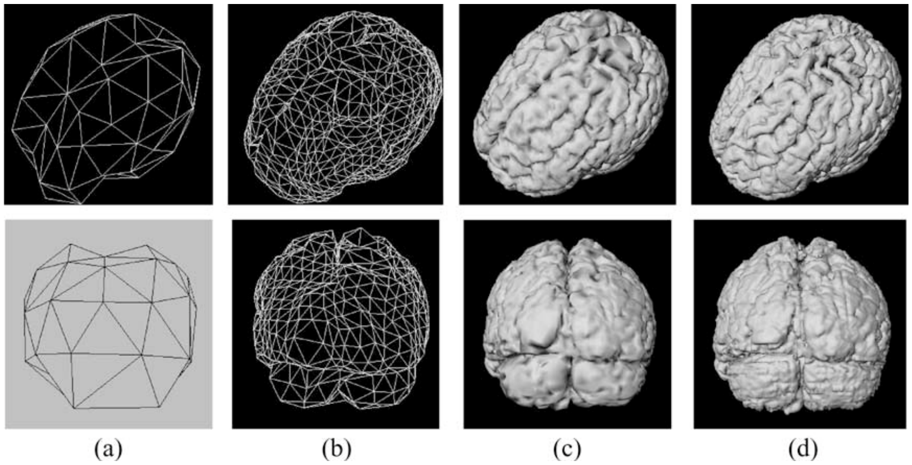

Figure 2. Brain surface reconstruction

Triangulation as a procedure also is applied in the duplex ultrasonic scanner for quantifying the speed of blood flow to two dimensions. In this procedure is applied in obtaining of surface triangulation from the three-dimensional image and in the triangular networks that are used to share the image in several non-overlapping regions and have similar characteristics.

Manufacturing of objects with molds

Most CAD / CAM objects today are made using some form of automatic production. Computers as modern multimedia appliances play an important role in the design and construction of these moving and non-moving objects. The process of constructing a particular object depends on the construction material, the shape and the quantity that is produced. The application of computational geometry in linear optimization is seen in the aspects of the production of plastic and metal objects. The process of producing metal objects is called casting. In Figure 3., the process of casting liquid metal into the mold is given.

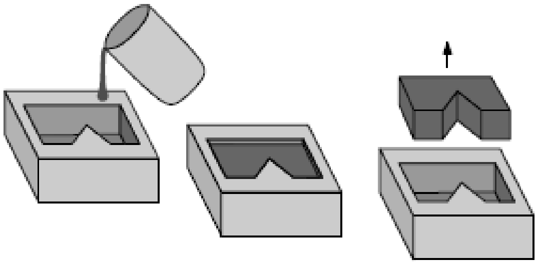

Figure 3. The casting process [8]

The casting process consists of three parts, the casting of liquid metal into the mold, the solidifies of liquid metal and the removal of the mold from the mold. In the process of manufacturing, removing the object from the mold is not always easy because the object can be stuck in the mold and may be required the breaking of mold. There are also objects for which it is very difficult to make a mold. These are generally objects that have winkled surfaces, such as spheres, ellipsoids, and so on. In order to avoid such complications in the manufacturing process, are making certain constraints. Due to this, in most of the cases in manufacturing are taken the objects which are polyhedral.

A greater number of casting uses a mold that consists of only one part and from which object can be removed only with one movement. For the production of objects by casting, one must first determine whether its shape is suitable for production by casting and how to make the appropriate mold to it. The shape of the object determines the cavity in the mold, from which it is understood that different molds must be made for the production of different objects.

Orientation is one of the most important traits in the process of removing objects from the mold. It is not possible to remove the object from the mold with the wrong orientation. Each object must have a horizontal upper side in the orientation, which is the only side that is not in contact with the mold. There are various potential orientations of objects, depending on how many facets the object has and for this reason, several different molds are being built in the process of production for these objects. The process of removing a particular object from the mold is a process that works with linear optimization. This is the process where we must find the solution to the problem where the object is castable. Today in the manufacturing are used the algorithms of incremental linear programming and randomized linear programming.

Point Location

One other application of computational geometry can be seen in finding a point location in a map. Point location problem occurs in various situations of today. It is used in traffic on the water into determining the current position of a particular floating object. Point location problems also occur in interactive geographic systems that show the map on the screen [8]. These systems provide information about the location that is selected with the mouse. In the process, the point location terrain is divided into subdivision S with n edges and are looking for the answer to the following question is asked: Is the query point q in the face f of S. This can be arrived withdrawing the vertical lines through all vertices of the subdivision where the plane is divided into vertical slabs.

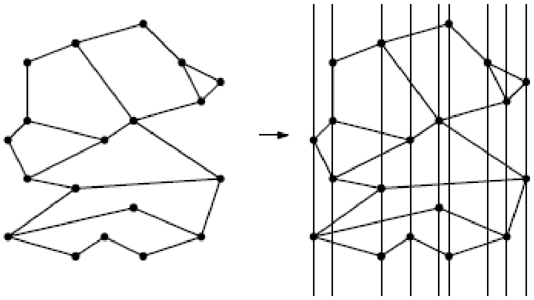

Figure 4. Partition into slabs [8]

By sorting in the array the x-coordinates of vertices is determined a slab that contains the query point q. This is achieved with a binary search and shows which segment is below q. If there is no segment below q, then q lies in the unbounded face of S.

A trapezoidal map is another decomposition that facilitates point location query. The time complexity of this decomposition is not much greater than the complexity of the original subdivision and fulfills several desirable properties in the point location problem. Trapezoidal maps are defined as sets of n non-crossing segments in the plane.

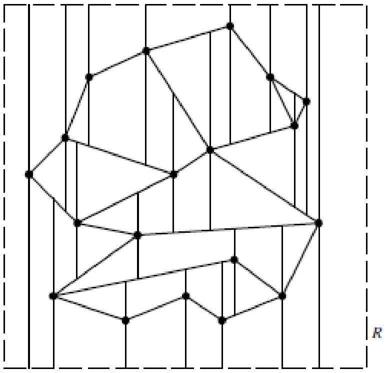

Figure 5. Trapezoidal maps [8]

The application of trapezoidal maps in a point location problem is realized by releasing of unbounded trapezoid faces which are at the boundary. This is done by introducing a large axis-parallel rectangle R that contains whole scenes and with the assumption that there are no two different points that have the same x-coordinates, i.e. there are no two points lies on the same vertical line. A set of these properties is a set of lines in a general position. Trapezoidal maps of planar division S is simply a subdivision of S, rectangle R, and lower and upper vertical extensions. For the construction of trapezoidal maps, is used the randomized incremental algorithm [8].

Robot Motion Planning

One of the main goals in robotics is the design of autonomous robots, which can plan their movements. For planning the movement, we must have information about the obstacles and the environment in which the robot will move. The construction of robots with such characteristics is very rare and in practice, more parts are used which are called robot arms. Each robot arms consists of several connected joints and is generally used to manipulate and assemble part of the items that are being manufactured.

In order to achieve this goal, robots in the process of moving a certain movement must move from one position to another. This robot movement is known as the motion planning problem. To define this problem, it is necessary to make several assumptions, for example, the environment in which the robot is moving, are the other objects static, can be represented the planar region with polygonal obstacles, on the road in which the robot is moving no people. Different movements that the robot can do depend on its mechanics. Some robots can move in any direction, while others are limited in their movements [8]. The movement of a robot into a 2-dimensional environment is done with translation and rotation in relation to the reference point *R*(0,0).

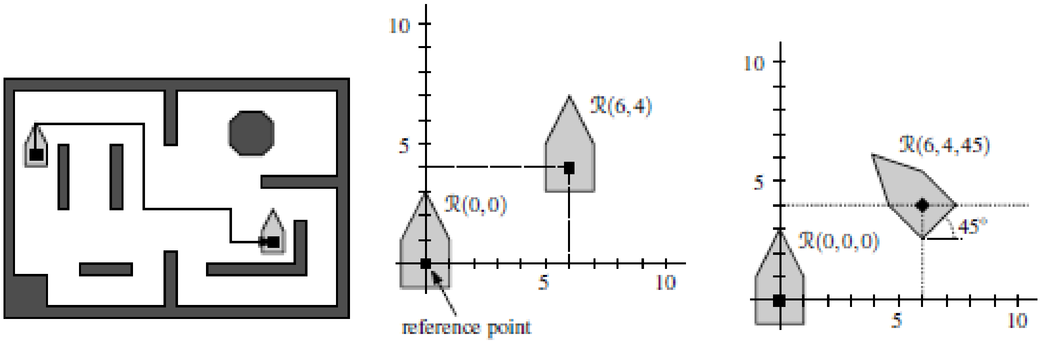

Figure 6. Robot motion planning [8]

For the robot movement also are used the trapezoidal maps. These maps are used to calculate the representation of the free space which finds the path for each starting point and goal position. The problem of robot motion planning can also be planned with the rotations defined by Minkowski sums [8].

4. Conclusion

Today, with the growing demand for skilled personnel for manufacturing and successful communication in the business process, companies and universities must be developed and restructured in parallel. With this new structurization, to use state-of-the-art technology and digital knowledge in solving geometric problems becomes more and more important. Solving various geometric problems both in the private and public sector today are the needs that are often seen in many business life situations. Naturally, this kind of business requires both the knowledge and skills of the employees of companies and institutions.

Unfortunately, in the Balkan region, because of inadequate training in universities, there are no strategies for education in the computational geometry of new generations and training of engineers who have acquired their education in the old way of education. The new education strategy must respond to the trends of the present. Young engineers should be ready to solve the geometric problems of the present and must be open to extending the application of computational geometry to various dynamically structured disciplines.

For current engineers knowledge of geographic information systems, computer graphics, computer design, spatial modeling, robot motion planning are just some of the areas where they must have the knowledge and which represent some of the application of the computational geometry. Therefore, one must recognize the great contribution of this discipline to give the importance it deserves in higher education.

The number of published papers and books is unsatisfactory and unfortunately shows a low level of development of this discipline in Macedonia and Serbia. In order to improve this situation, universities should engage in also this discipline as a compulsory subject at faculties where engineering staffs are trained. Since the importance of computer geometry cannot be ignored today, state institutions and private companies must invest in training their staff in this discipline.

REFERENCES

[1] Akl, S., “A constant-time parallel algorithm for computing convex hulls,” *BIT*, vol. 22, 1982, pp. 130-134.

[2] Akl, S., *The Design and Analysis of Parallel Algorithms*, Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[3] Alt, H. and Yap, C. K., “Algorithmic aspects of motion planning: a tutorial, part 1,” *Algorithms Review*, vol. 1, No. 1, pp. 43-60.

[4] Alt, H. and Yap, C. K., “Algorithmic aspects of motion planning: a tutorial, part 2,” *Algorithms Review*, vol. 1, No. 2, pp. 61-78.

[5] Avis, D. and Toussaint, G. T., “An efficient algorithm for decomposing a polygon into star-shaped pieces,” *Pattern Recognition,* vol. 13, 1981, pp. 295-298.

[6] Chvatal, V., “A combinatorial theorem in plane geometry,” *Journal of Combinatorial Theory, Series B,* vol. 18, 1975, pp. 39-41.

[7] De Berg M, *Efficient Algorithms for Ray Shooting and Hidden Surface Removal,* University of Utrecht, 1992.

[8] De Berg M, Cheong O, van Kreveld M and Overmars M, *Computational Geometry, Algorithms and Applications*, Third Edition, Springer-Verlag Berlin Heidelberg, 2008

[9] Dobkin, D. and Silver, D., “Recipes for geometry & numerical analysis - Part I: An empirical study,” *Proc. 4th Annual Symposium on Computational Geometry*, Urbana, June 1988, pp. 93-105.

[10] Edelsbrunner, H. and Mucke, E., “Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms,” *Proc. 4th Annual Symposium on Computational geometry*, Urbana, Illinois, June 1988, pp. 118-133.

[11] Guibas, L. and Hershberger, J., “Optimal shortest path queries in a simple polygon,” *Journal of Computer and System Sciences*, vol. 39, No. 2, 1989, pp. 126-152.

[12] Hershberger, J., “Optimal parallel algorithms for triangulated simple polygons,” *Proc. 8th ACM Symposium on Computational Geometry,* Berlin, June 10-12, 1992, pp. 33-42

[13] Imai, H. and Iri, M., “Polygonal approximations of a curve - formulations and algorithms,” in *Computational Morphology,* G. T. Toussaint, ed., North-Holland, 1988, pp. 71-86.

[14] Karasik, Y. B. and Sharir, M., “Optical computational geometry,” *Proc. 8th ACM Symposium on Computational Geometry,* Berlin, June 10-12, 1992, pp. 232-241.

[15] Kreveld, M. van, *New Results on Data Structures in Computational Geometry*, University of Utrecht, 1992.

[16] Melter, R. A., Rosenfeld, A., and Bhattacharya, P., eds., *Vision Geometry*, American Mathematical Society, 1991.

[17] Milenkovic, V., “Verifiable implementations of geometric algorithms using finite precision arithmetic,” Tech. Rept. CMU-CS-88-168, Carnegie Mellon University, July 1988.

[18] Minsky, M. and Papert, S., *Perceptrons: An Introduction to Computational Geometry*, M.I.T. Press, 1969.

[19] Mortenson, M. E., *Geometric Modeling*, John Wiley & Sons, 1985.

[20] Preparata, F. P. and Shamos, M. I., *Computational Geometry*, Springer-Verlag, NewYork, 1985.

[21] Saračević, M. Methods for solving the polygon triangulation problem and their implementation (*Ph.D. thesis), Faculty of Science and Mathematics*, University of Niš, 2013.

[22] Saračević M., Mašovic S., Miloševic D. (2013), Java implementation for triangulation of convex polygon based on Lukasiewicz's algorithm and binary trees, *Southeast European Journal of Soft Computing*, ISSN: 2223 -1859, Vol.2, No.2.

[23] Saračević M., Mašović S., Milošević D., Kudumović M. (2013), Proposal for applying the optimal triangulation method in 3D medical image processing: Software solution based on Java Net Beans environments, *Balkan Journal of Health Science* (BJHS), ISSN: 2303 -4092, Vol.1, No.1, pp. 27-34.

[24] Saračević M., Stanimirović P., Krtolica P., Mašović S. (2014), Construction and Notation of Convex Polygon Triangulation based on ballot problem, *ROMJIST- Journal of Information Science and Technology*, ISSN: 1453–8245, Vol.17, No.3, pp. 237-251.

[25] Saračević M., Stanimirović P., Mašović S. (2013), Object-oriented analysis and design for one algorithm of computational geometry: Forward, reverse and round-trip engineering, *Journal of Information Technology and Applications*, ISSN: 2232-9625, DOI: 10.7251/JIT1302096S, Vol.3, No.2, pp. 96-106.

[26] Saračević M., Mašović S., Kamberović H. (2012), Procedure division of convex polygon triangulation and application in computer graphics, *6th International quality conf., Faculty of Engineering*, ISBN: 978-86-86663-82-5, Center for Quality, Vol.2, pp. 991.

[27] Stanimirović P., Krtolica P., Saračević M., Mašović S. (2014), Decomposition of Catalan numbers and Convex Polygon Triangulations, *International Journal of Computer Mathematics*, Vol. 91, No. 6, pp. 1315–1328, ISSN: 0020-7160, publisher: Taylor and Francis: UK.

[28] Stanimirović P., Krtolica P., Saračević M., Mašović S.(2012), Block Method for Convex Polygon Triangulation, *ROMJIST- Journal of Information Science and Technology*, ISSN: 1453–8245, Vol.15, No.4, pp. 344–354

[29] Toussaint, G. T., "Efficient triangulation of simple polygons," *The Visual Computer*, vol. 7, No. 5-5, September 1991, pp. 280-295.

[30] Toussaint, G. T., "What is Computational Geometry," *Computational Geometry Laboratory*", School of Computer Science McGill University. Montreal, Quebec, Canada

[31] Wood, D., "An isothetic view of computational geometry," in *Computational Geometry*, G. T. Toussaint, ed., North-Holland, 1985, pp. 429-459.

[32] Tabak J., *The History of Mathematics Geometry: The Language of Space and Form*, Facts On File, Inc. NewYork, 2004.

[33] Tabak J., *The History of Mathematics and the Laws of Nature: Developing the Language of Science*, Facts On File, Inc. NewYork, 2004.