

Geometric Aspects of the Functional-Voxel Implementation of the ORCA Algorithm

Alexey Tolok¹ and Anastasiya Sycheva¹

¹ V.A. Trapeznikov Institute of Control Science of Russian Academy of Sciences, 65 Profsoyuznaya street, Moscow, 117997, Russia

Abstract

The problem of avoiding a collision between moving agents constantly arises in multi-agent systems with decentralized control. The various algorithms for solving this problem are accompanied by computational complexity and increasing computational power requirements as the number of agents in question increases. There are difficulties in adapting these algorithms to practical applications on mobile platforms. It is necessary to develop simpler computational schemes and to apply appropriate models.

The most computationally expensive step in the classical collision avoidance algorithm ORCA is to calculate the mutual half-planes of possible collision for each pair of robots and use linear programming to calculate the new velocity from them. The application of the functional-voxel method will simplify the necessary calculations by storing in graphical images the local geometric characteristics of the searched domain. Moreover, the application of such models will make it possible to perform most of the necessary calculations in advance, which will accelerate the work of the algorithm.

This paper proposes the construction of a functional-voxel model of a required geometric domain by interpolating the contour of the domain using Bézier curves. The local geometric modelling by means of local zeroing function is used as a tool for functional-voxel curve modelling.

The obtained functional-voxel model represents a static case of possible mutual positioning of two agents. A four-dimensional graphical model is proposed to solve the dynamic case. This model performs the distribution of the static case modelling results in the space-time characteristics.

Keywords

Multi-agent system, ORCA, Bézier curve, functional-voxel modelling

1. Introduction

Existing algorithms of robot motion in multi-agent systems are usually based on the geometric representation of the point-positions of agents and their velocity vectors [1]. In papers devoted to such algorithms [2-3], as a rule, only the general logic of behavior of each agent under threat of collision is described, and all geometrical aspects of such approaches are not presented.

The classical ORCA algorithm for determining the collision avoidance vector constructs a collision velocity region. The geometry of this region is formed by two tangents to a circle. The process of calculating this region is the most computationally intensive of all necessary calculations. This makes it difficult to use the algorithm on mobile platforms.

The considered software implementations of such algorithms do not follow the basic principles of the classical algorithm outlined in the theoretical part. They use simplified vector schemes, which reduces the accuracy of the search for the optimal solution.

GraphiCon 2021: 31st International Conference on Computer Graphics and Vision, September 27-30, 2021, Nizhny Novgorod, Russia

EMAIL: tolok_61@mail.ru (A. Tolok); a.a.sycheva@mail.ru (A. Sycheva)

ORCID: 0000-0002-7257-9029 (A. Tolok); 0000-0003-3230-4271 (A. Sycheva)



© 2021 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

The need arises to find accurate but simplified calculation schemes for the application of this algorithm in simulation systems, as calculations need to be carried out at each point in time in relation to each pair of robots under consideration. For example, approaches that allow the required parameters to be pre-calculated for the static problem, and then distribute the calculation results in space-time characteristics to solve the dynamic case.

Simplification of mathematical operations is a traditional advantage of functional-voxel modelling [4], due to the fact that part of calculations replaced by the generation of a graphic image containing all necessary local information about object geometry. The application of this method allows a significant part of calculations to be carried out in advance, before the robot movement process begins. The robot can perform calculations by referring to the required image of the functional-voxel model and obtaining the requested characteristic.

2. Local computer geometric interpolation of the collision velocity area

To construct a functional-voxel model requires the formulation of an analytical function describing the collision zone. For this purpose, it is necessary to obtain the R-functional modeling law [5] of the contour of its boundary on a given rectangular region. Based on the position of the half-planes organizing this contour, the inner region must contain a characteristic slope from the central ridge to the boundary of the collision zone. An example of such an image is shown in Figure 1.

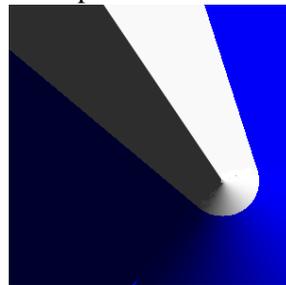


Figure 1: Constructing the collision velocity domain using FVM

There were earlier attempts to describe the required geometrical region by means of the apparatus of R-functions for further application in functional-voxel modeling [6]. A schematic illustration of the figures forming the collision velocity region is shown in Figure 2a. The proposed mathematical interpretation for contouring by R-functional constructions does not create the necessary interior geometry (Figure 2b). Other approaches for description of the required domain are needed.

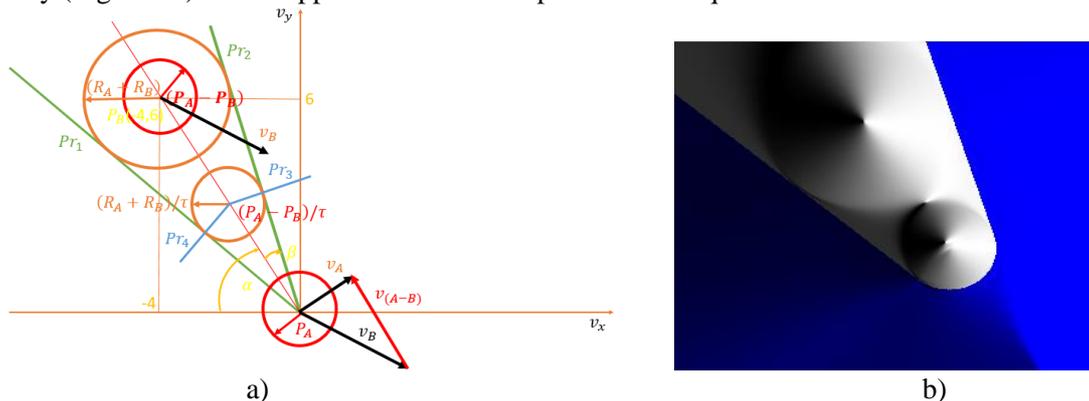


Figure 2: Schematic representation and resulting R-functional model of domain

By interpolating the robot collision velocity area with some curve, the properties of functional-voxel curve modelling can be applied. This requires the contour of the area to be defined, so an analytical representation of the curve describing this contour is needed

The task of constructing the required contour can be reduced to interpolating the contour of a curve that passes through the given points. Parametrically defined curves are a universal design tool for constructing smooth contours. There are various approaches to construct a parametrically defined curve from predetermined points [7, 8]. It seemed easy enough to choose the most suitable approach and

proceed to functional-voxel modelling. But the following problems arose in the process of solving this question. Let us consider a classical representative of parametric curves – the Bézier curve.

A smooth Bézier curve can be functional-voxel modelled in various ways. In particular, it can be modelled by applying the De Casteljto algorithm using the apparatus of R-functions [9]. However, the application of this approach cannot generate the required monotonic smooth surface. Moreover, creating complex curves with variable curvature is only possible with the discontinuity of the surface caused by the necessity of matching convex regions (Figure 3).

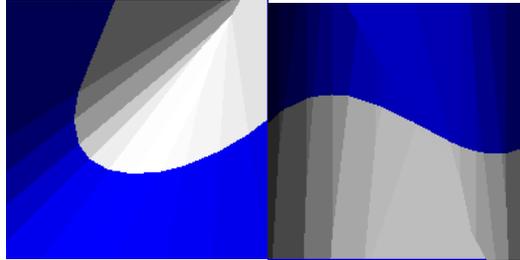


Figure 3: Examples of Bézier curves constructed by a combination of R-voxel modelling and De Casteljto's algorithm

Let us consider an approach to construct a functional-voxel model of a smooth curve that avoids the above disadvantages. This approach applies local geometric modelling by means of R-functional intersection. A smooth curve can be represented by a set of consecutive short segments. The end of one segment matches the beginning of the next segment. Given a small discretization step, such a segmentation will produce a sufficiently smooth curve.

The local zeroing function on the area of positive function values must be used as such a segment. When two such curve segments intersect on a smooth surface, the domain of values of each subsequent segment affects the already obtained domain. In this way, the smooth plastic surface forms the zero boundary as a given parametric curve (Figure 4).

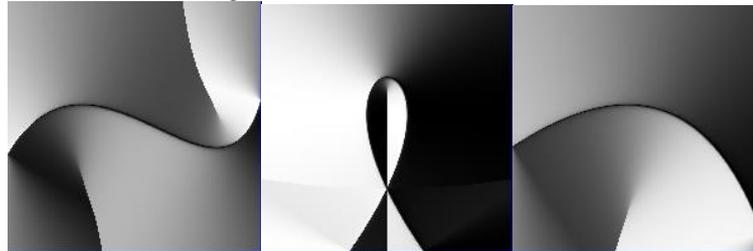


Figure 4: Examples of Bézier curves constructed via local zeroing function

The presented images show a high-quality image of the contour. The plastic smooth surface is accompanied by a characteristic ridge in the inner region of the Bézier curve, which are suitable characteristics of the model for further application in the given problem.

On the one hand, it is possible to construct the required contour as a single continuous Bézier curve. However, as can be seen in Figure 5, the resulting contour will be far from the required geometry and will be rather parabolic, while the branches of the resulting geometry should be tangent straight lines to the tip circle of the contour.



Figure 5: Schematic representation and functional-voxel model of a single curve contour

On the other hand, it is possible to construct a contour formed by several parametrically defined geometric shapes (a straight line segment, a circle segment) using this approach. Consider the construction of a contour consisting of such elements. Figure 6a shows a contour model constructed by means of two line segments and a circle segment localized by line segments. Figure 6b shows a variant

of the model in which, instead of two segments, linear Bezier curves are used, i.e., they are also localized by minimal segments.

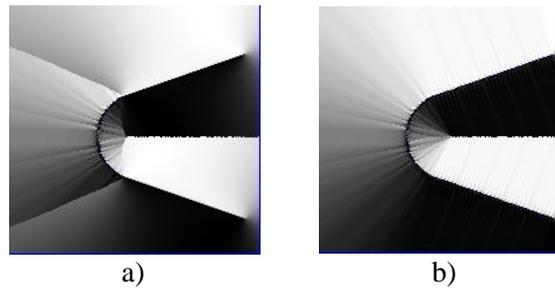


Figure 6: Contour construction by using 3 segments and a Bézier curve

As can be observed, the second construction method results in a more monotonous surface.

As a result, the functional-voxel model of the required region consists of a set of 4 image-models reflecting local geometric characteristics n_1 , n_2 , n_3 , n_4 for a local function $n_1x + n_2y + n_3z + n_4 = 0$ describing the inner and outer domain of the collision velocity zone contour at each point. An example of such image-models is shown in Figure 7.

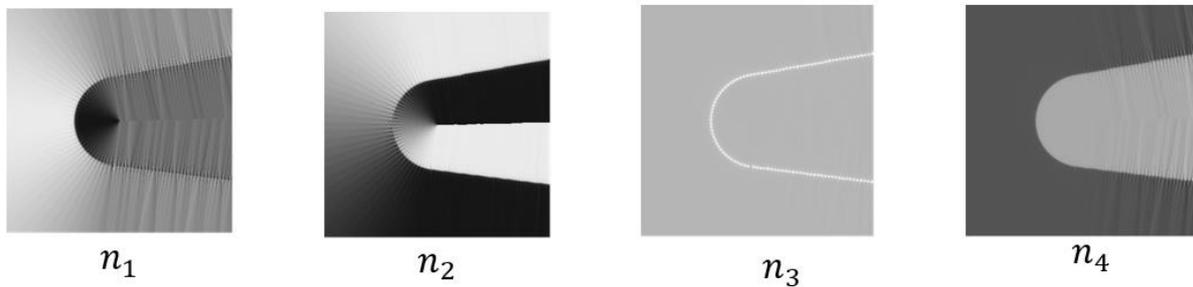


Figure 7: M-images of the functional-voxel model of the required domain

The accuracy of the proposed model is controlled by increasing the size of the graphic image and decreasing the discretization step. It is not possible to obtain a more accurate model using only one of these components. Decreasing the discretization step below some threshold at the same size of the graphical image will not increase the accuracy of the image in comparison with a larger discretization step, because the changes in the value of the calculated coordinates will be less than a pixel. Conversely, increasing the size of the graphical image with an unchanged discretization step will not give a more accurate image.

A visual improvement in the accuracy of the model is shown in Figure 8. Reducing the sampling step Δt from 0.1 to 0.01 with an image size of 800×800 pixels demonstrates a noticeable refinement of the resulting image.

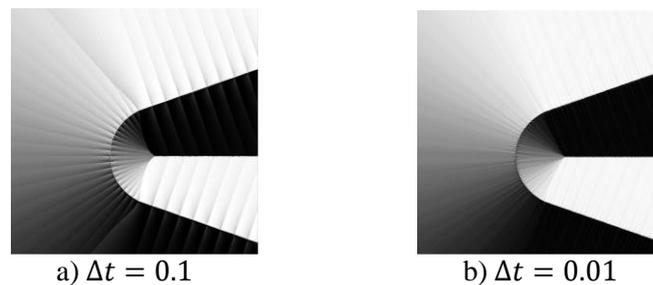


Figure 8: Effects of changing the sampling step

The smoothness of the image can also be improved by changing from a linear to a quadratic expression of the intersection operation. The R-function intersection operation of functions X and Y is given by the expression:

$$X \vee_{\alpha} Y = \frac{1}{1 + \alpha} \left(X + Y - \sqrt{X^2 + Y^2 - 2\alpha XY} \right),$$

where $\alpha \in (-1; 1]$.

If $\alpha = 1$ there is a linear expression of the intersection operation:

$$X \wedge_1 Y = 0.5(X + Y - |X - Y|)$$

This expression was used in generating all previous images.

If $\alpha = 0$ there is a quadratic expression of the intersection operation:

$$X \wedge_0 Y = X + Y - \sqrt{X^2 + Y^2}$$

Figure 9 shows a comparison of the images obtained in the linear case (a), quadratic with parametrized segments (b) and quadratic with non-parametrized segments (c).

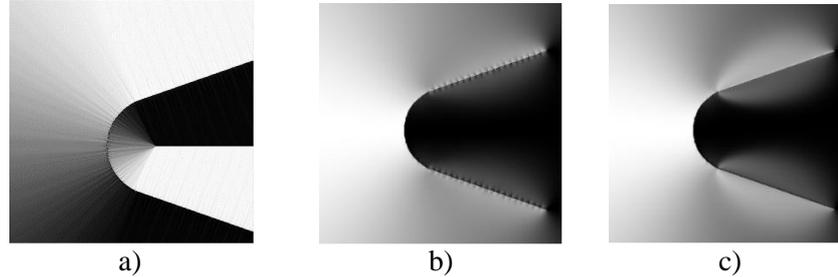


Figure 9: Obtaining curve images in different techniques

In the quadratic case, the smoothest surface is constructed. However, a blurring of the ridge required for further calculations is observed. It should be noted that in the quadratic case, when the segments are parametrized, their geometric influence on the overall shape of the resulting surface is particularly reduced, in contrast to the linear case.

As a result, it was found that the geometric model of the collision velocity zone which constructed by interpolating the contour of the region with linear R-voxel intersection of local zeroing function with parameterized tangent line segments is the most appropriate for the solution of the problem.

3. Four-dimensional model of the relative positioning of two robots

The resulting two-dimensional graphical images characterize the static situation of the relative position of the two robots relative to each other in x and y coordinates in the plane. In this case, the motion vectors of the robots lie on the same line. In practice, however, it is necessary to dynamically analyze the distance between the robots. Also, the motion vectors of the robots can be angled at different angles. Thus, it is necessary to consider the distance between the robots d and the angle of deviation α of the motion vector of the approaching robot from the motion vector of the current robot.

Initially it was supposed to perform necessary transformations of the obtained geometrical model via operations of spatial transformations [10] (parallel transfer, rotation and scaling). However, by using the functional-voxel model, it is possible to obtain the full variety of possible robot arrangements in a single model.

Reducing the distance between the robots leads to a change in the geometry of the resulting area. A gradual divergence of the tangent lines is observed (Figure 10).

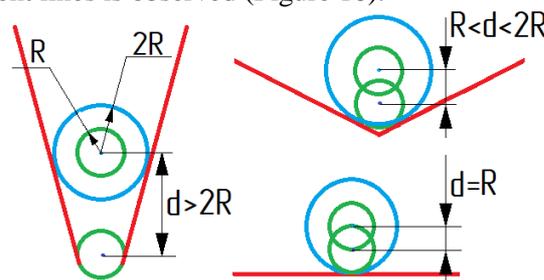


Figure 10: Changing the geometry of the collision area

The sequence of graphical models with increasing slope of the tangent branches to the circle segment is not difficult to implement. Figure 11 shows two images obtained by reducing the initial distance between the robots by a certain step. There is a divergence of the tangent lines of the generated area in different directions.

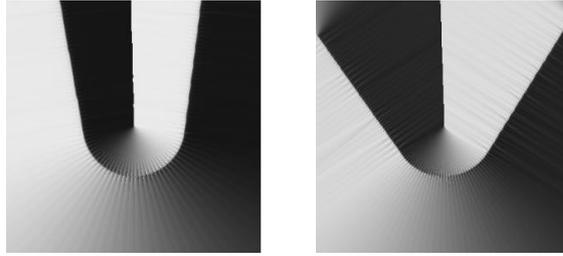


Figure 11: Graphic images of different distances between robots

The set of such images with different angles between tangent lines simulates various situations of mutual positioning of robots in case the motion vectors of robots are located on the same line. In other words, the local coordinate system of the approaching robot does not deviate from the local coordinate system of the robot in question.

Considering the angle of rotation of the local coordinate system as the robot approaches the current robot is possible through a standard rotation procedure. Initially, the origin of coordinates is in the centre of the robot in question and the approaching robot moves towards it in the opposite direction to the axis OY. It is possible to simulate the approach of the robot from any other position by rotating the local coordinate system of the robot in question by an angle α (Figure 12).

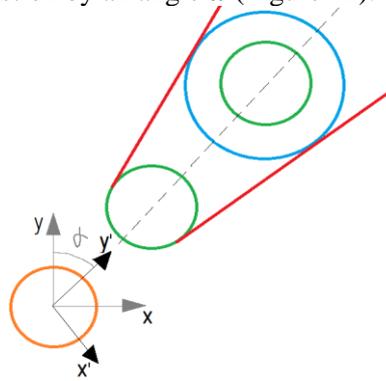


Figure 12: Rotation of the local coordinate system

In functional-voxel modelling, the rotation around the zero of coordinate system occurs by transforming the local geometric characteristics at each point of the local function area:

$$\begin{aligned} n'_1 &= n_1 \cos \alpha + n_2 \sin \alpha \\ n'_2 &= -n_1 \sin \alpha + n_2 \cos \alpha \\ n'_3 &= n_3 \\ n'_4 &= n_4 \end{aligned}$$

It is also worth noting that visually the imaged area will not rotate (only the colour of the palette at each point changes), because the local function area rotates with the object. For example, rotating the coordinate system by 90 degrees essentially means swapping the A and B images of the functional-voxel model. Therefore, for clarity, coordinate axes have been added to further images. An example of a 45-degree rotation around the coordinate axis is shown in Figure 13.

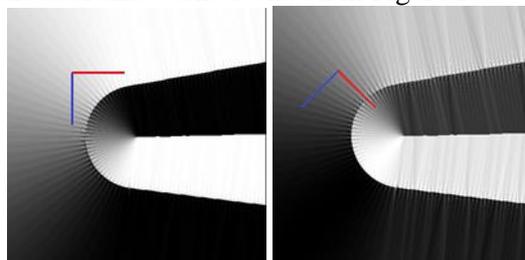


Figure 13: Example of rotating the coordinate axis by 45 degrees

By rotating around the origin, various positions of the approaching robot in relation to the current robot are simulated with a fixed distance between them.

A full dynamic representation of the convergence of the robots requires the construction of images with different distances between the robots and the construction of images with different angles of rotation around the origin for each of the them.

As a result, the mutual location of the two robots is defined by four variables - x and y coordinates, contained in a two-dimensional graphical image, the distance between the robots d and the angle of inclination α . Thus, the dynamics of motion of the two robots will be displayed in a four-dimensional model, a visual representation of which is shown in Figure 14.

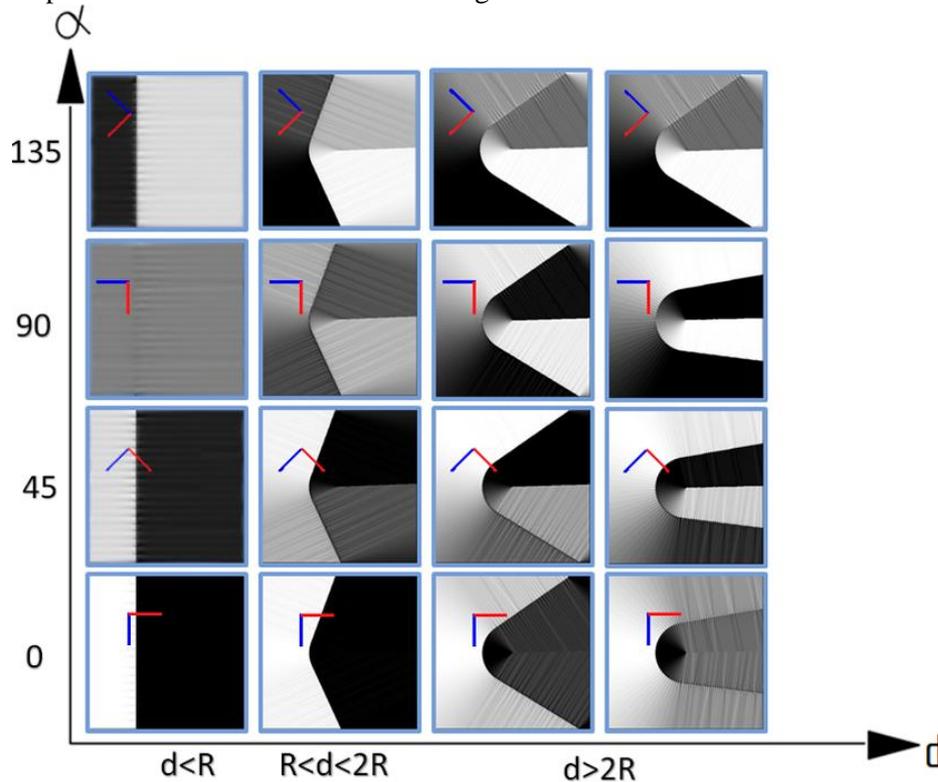


Figure 14: Four-dimensional robots placement model

Each graphic image of this model shows a specific plane for the situation in the positioning of two robots. At the same time, the model describes all possible variants of the mutual arrangement of the robots.

It is possible to refine this model both by increasing the number of graphic images (decreasing the discretization step of distance d and angle α) and by increasing the accuracy of the images. However, even with a small number of images and their low accuracy, it is always possible to determine the direction of the collision avoidance vector. This allows to use models, which do not require high computational power to create.

4. Conclusion

The obtained 4-dimensional functional-voxel model displays the dynamics of collision velocity zone representation for basic situations of relative position of two robots and allows to eliminate from the procedure of ORCA region determination the main part of calculations, involved for determination of deviation of velocity direction of the considered robot. This approach will justify its application in various multicriteria calculations in mathematical modeling problems represented geometrically.

5. References

- [1] A. V. Timofeev, R. M. Yusupov, Principles of Integrated Multi-Agent Navigation and Intelligent Control Systems for Mechatronic Robots, Information Technologies & Knowledge 3 (2011) 327. in Russian.
- [2] Jur van den Berg, Stephen J Guy, Ming C Lin, Dinesh Manocha, Reciprocal n-Body Collision Avoidance, Robotics Research (2011) 3-19. doi: 10.1007/978-3-642-19457-3_1.

- [3] D. Alejo, J. A. Cobano, G. Heredia and A. Ollero, Optimal Reciprocal Collision Avoidance with Mobile and Static Obstacles for Multi-UAV Systems, International Conference on Unmanned Aircraft Systems (2014) 1259-1266. doi: 10.1109/ICUAS.2014.6842383
- [4] A. V. Tolok, Functional voxel method in computer modeling, Moscow, Fizmatlit, 2016. in Russian.
- [5] V. L. Rvachev, Theory of R-functions and Some Applications, Kiev, Naukova Dumka, 1982. in Russian;
- [6] A. V. Tolok, P. M. Harlanova, L. N. Sizova, M. A. Loktev, A. A. Sycheva, Modeling of control algorithms for groups of mobile robots by means of the functional-voxel method, Stankoinstrument 4 (2020) 76-81. in Russian.
- [7] E. V. Konopatsky, Geometric modeling of multifactor processes based on point calculus, Ph.D. thesis, Nizhny Novgorod State University Of Architecture And Civil Engineering (NNGASU), Nizhny Novgorod, 2020. In Russian.
- [8] V. V. Borisenko, Construction of an Optimal Bezier Spline, Fundamental and Applied Mathematics 3 (2016) 57 –72. in Russian.
- [9] A. V. Tolok, A. A. Sycheva, N. B. Tolok, Construction of the Functional Voxel Model for a Spline Curve, in: Proceedings of the 30th International Conference on Computer Graphics and Machine Vision, GraphiCon 2020, CEUR Workshop Proceeding, St.Petersburg, 2020. URL: <http://ceur-ws.org/Vol-2744/paper52.pdf>.
- [10] E. V. Lotorevich, Geometric transformations of the functional voxel model space, Master`s thesis, Nizhny Novgorod State University Of Architecture And Civil Engineering (NNGASU), Nizhny Novgorod, 2016. In Russian.