Functional-voxel Modeling of Navigation Algorithm ORCA*

Mikhail Loktev[0000-0002-0465-1201], Alexey Tolok[0000-0002-7257-9029], and Vladimir Romakin1[0000-0002-9678-0130]

Laboratory of Computer Graphics, V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, 65 Profsoyuznaya street, Moscow, 117997, Russia

Abstract. The article considers an example of modeling the ORCA algorithm using the functional-voxel method. An analytical description of the permissible collision zone using the set-theoretic apparatus of Rvachev functions (R-functions) is proposed. Based on graphical image models, an approach to constructing the area of permissible robot speeds on the plane and in space has been developed.

Keywords: Functional-voxel Modeling, ORCA, Multi-agent Systems.

1 Introduction

The principles that solve navigation problems in multi-agent systems are comparable to the interaction of social public groups. They can also be divided into centralized and decentralized [2]. With a centralized approach, the actions of agents are often considered from the point of view of fulfilling a joint goal by joint efforts. The action strategy of such agents is described through global system states and joint actions of agents [3]. In this case, the agents are gathered in groups, often choosing a leader, and solve the problem by joint team interaction. However, there are situations in which a decentralized agent action planning system is required [3], including decentralized navigation systems. In such systems, each agent has its own goal (for example, delivery service agents). The algorithm for solving this problem involves taking into account the situation throughout the scene and makes an independent decision regarding the behavior of other agents. And this happens with every single agent. In such an algorithm, optimization of the solution is necessarily applied, taking into account the set goal and the many restrictions that arise in its path. One of the representatives of the algorithms of this class is the ORCA algorithm. It is based on the construction of linear half-spaces with respect to each agent, forming at the intersection an area of possible solutions for applying the optimization principles of linear mathematical programming.

* Supported by V.A. Trapeznikov Institute of Control Science of Russian.

The article discusses the approach to the construction of the ORCA algorithm by means of functional voxel modeling based on the analytical principles of the R-functional construction of a complex geometry domain [1].

At this stage, new challenges are opened for the effective implementation of such a model at the software and technical level.

2 Related Work

The ORCA algorithm was first introduced in 2011 [4], as the principle of decentralized navigation for large groups of agents, which provides a sufficient condition for avoiding collisions between agents with individual goals. Moreover, each agent takes at least half of the “responsibility” for avoiding a collision with another agent [2]. This principle is based on an earlier work [8] devoted to the formation of the collision region — the velocity obstacle (VO). A velocity obstacle is a region formed by velocity vectors and leading to a collision of agents over a certain period of time.

Subsequent scientific studies of other groups of authors used the ORCA algorithm to solve various applied problems. For example, the use of various navigation algorithms for mutual local evasion of virtual agents in video games is considered in [9]. Simulation of automobile traffic at the intersection based on the ORCA algorithm is considered in [10]. A number of modifications to the ORCA algorithm have also been developed. For example, a new accident-free approach to navigation of nonholonomic robots based on ORCA and taking into account the kinematics of the studied robots was presented in [11]. Also noteworthy is the work [12] which presents a unified collision-avoidance algorithm for the navigation of arbitrary agents, from pedestrians to various types of robots, including vehicles.

3 Urgency of the problem

The presented works are based on a significant number of complex geometric calculations at each stage of the operation of the ORCA algorithm. Moreover, for a large group of agents, it is required to carry out a mutual calculation of the possible velocity corrections for each pair of robots, followed by the calculation of their new speed on the basis of linear programming. Moreover, for a large group of agents, it is necessary to carry out a mutual calculation of possible speeds for each pair of robots, followed by calculating their new speed based on linear programming. Simplification of mathematical operations is traditionally considered an advantage of functional-voxel modeling, when part of the calculations is replaced by the formation of a graphic image with all the necessary local information it. Similar studies have been successfully carried out on the implementation of problems of local optimization of the path to the goal with obstacle avoidance [6] and in the development of functional-voxel modeling of mathematical programming problems [7]. Based on the proposed approach, various formulations of the path finding problem have already been implemented [6, 13-14].

The functional-voxel representation of the collision region can be calculated in advance, and during the operation of the algorithm only simple calculations will be applied to the obtained local geometric characteristics of the model, which greatly simplifies the computational process. Such a model can be used as a graphic template for modeling the relations of two robots defined in space.

4 Analytical presentation of Velocity Obstacle - $VO^T_{B|A}$

Consider a specific example of the mutual arrangement of robots A and B with their effective radii R_A and R_B equal to unity, where $P_B(-2{,}3)$and $P_A(2,-3)$ (Fig. 1). Bind the position of the robots to the speed coordinate system. To do this, transfer the origin to the point P_A, then P_B(-4,6) will change its coordinates. The initial speeds of both robots are determined by the vectors v_A ($v_{Ax} = 1.5$, v_{Ay} =1.0) and v_B($v_{Bx} = 3.0$, v_{By} =-1.5), and the difference of these vectors, which determines the vector of approach of the robots, is determined by the vector $v_{(A-B)}$($v_{(A-B)x}$ =-1.5, $v_{(A-B)y}$ =2.5) as shown in Figure 2.

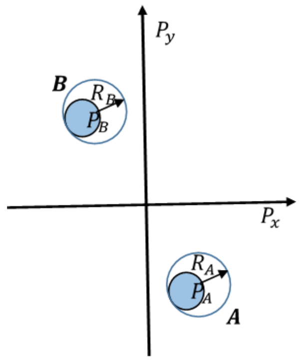


Fig. 1. The starting position of the robots

The difference in the coordinates of the centers of the agents will give a new point$(P_A - P_B)$, which in this case will coincide with the position of agent B. At this point, construct a new circle with a common radius of two robots $(R_A + R_B)$. Lines $Pr_1(x,y)$ and $Pr_2(x,y)$ define half-spaces for lines tangent to a circle with radius $(R_A + R_B)$ originating from the origin. To describe their location, it suffices to calculate the angles $\alpha = arctg\left(\frac{y_B}{x_B}\right)$ and $\beta = \arccos\left(\frac{\sqrt{(x_B^2+y_B^2)-(R_A+R_B)^2}}{\sqrt{x_B^2+y_B^2}}\right)$.

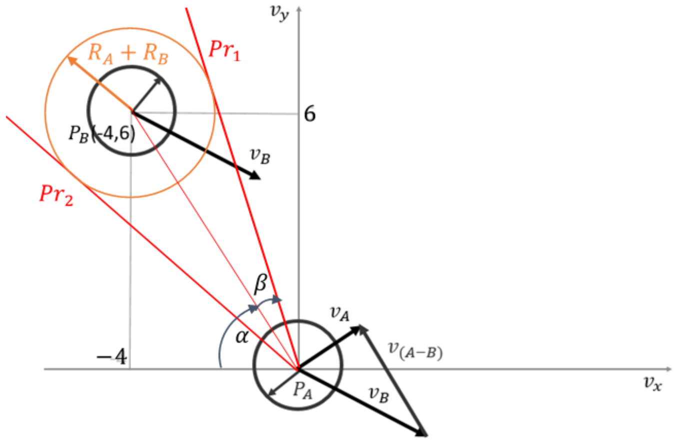


Fig. 2. Description of the scene with two agents A and B with given velocity vectors

The equations of the lines $Pr_1(x, y)$ and $Pr_2(x, y)$ take the form:

$$Pr_1(x, y) = y - tg(\alpha + \beta)x, \tag{1}$$

$$Pr_2(x, y) = tg(\alpha - \beta)x - y. \tag{2}$$

The area formed between them determines the zone of a possible collision when the velocity difference vector $v_{(A-B)}$ gets into it. However, the dynamics of the current process should be taken into account. For this, it is worth introducing the parameter of the time interval of the interaction of robots τ, which allows us to clarify the position of the critical zone taking into account the distance between two robots. The introduction of such a parameter makes it possible to clarify the collision avoidance zone. We introduce the value τ =2.0, as the interval of the known (spare) time for which it is required to determine the region of possible collisions.

The position of the new circle defining the beginning of the danger zone is set by the center point $(P_A - P_B)/\tau$ and radius $(R_A + R_B)/\tau$ (Fig. 3). The resulting circle will determine the border of the zone of collision with robot B closest to robot A over a time interval τ.

To construct an algorithm for determining the membership of the end point of the velocity difference vector $v_{(A-B)}$ in the region of the obtained collision region $VO^{\tau}_{A|B}$, it is necessary to divide this zone into two sections. The first section controls the entry of the vector $v_{(A-B)}$ into the $VO^{\tau}_{A|B}$ zone, bounded by the straight lines: Pr_1, Pr_2, Pr_3 and Pr_4, where Pr_3 and Pr_4- perpendiculars dropped to the lines $Pr_1$and Pr_2 from the center of the circle with radius $\frac{(R_A+R_B)}{\tau}$ (Fig. 3.). The equations of such lines are described as follows:

$$Pr_3(x, y) = \left(y - \frac{y_B}{\tau}\right) - \text{tg}\left(\frac{\pi}{2} + \alpha - \beta\right)\left(x - \frac{x_B}{\tau}\right), \tag{3}$$

$$Pr_4(x,y) = \left(y - \frac{y_B}{\tau}\right) - \mathrm{tg}\left(\frac{\pi}{2} + \alpha + \beta\right)\left(x - \frac{x_B}{\tau}\right). \quad (4)$$

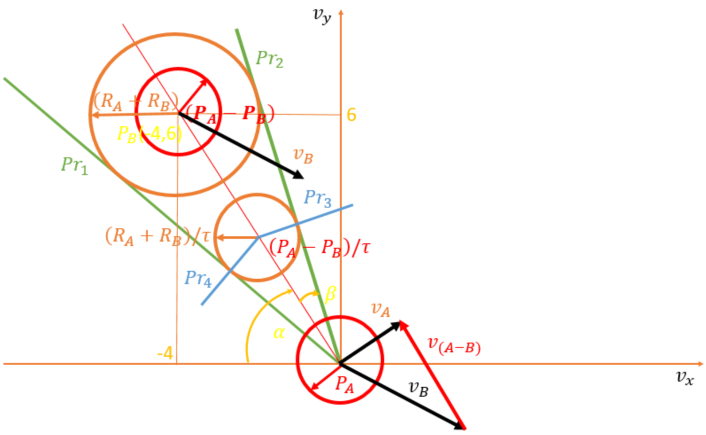


Fig. 3. Formation of collision region $VO^{\tau}_{A|B}$

At the initial stage of the algorithm (condition 1), we calculate the values of $Pr_1(v_{(A-B)x}, v_{(A-B)y})$ and $Pr_2(v_{(A-B)x}, v_{(A-B)y})$ using equations (1) and (2). If both values are positive, then the point with coordinates $(v_{(A-B)x}, v_{(A-B)y})$ is located between the lines Pr_1 and Pr_2, and therefore is for further study. Otherwise, this point does not fall into zone $VO^{\tau}_{A|B}$ and the robot A can move relative to robot B at the same speed and in the same direction.

If point $(v_{(A-B)x}, v_{(A-B)y})$ is between these lines, you should clarify its position by checking for a positive sign the values of any of the expressions (condition 2): $Pr_3(v_{(A-B)x}, v_{(A-B)y}) \geq 0$ or $Pr_4(v_{(A-B)x}, v_{(A-B)y}) \geq 0$. The fulfillment of the second condition is sufficient to begin to determine the direction of the normal $\vec{n}$ to the nearest boundary of the required region $ORCA_{A|B}$ for calculating speed correction vector of robot A to ensure its exit from zone $VO^{\tau}_{A|B}$. Denote such a correction vector $\vec{w}$. The direction $\vec{w}$ coincides with the direction $\vec{n}$, and its length is determined by the distance to the nearest boundary of zone $VO^{\tau}_{A|B}$. In this case, the definition of the nearest boundary is considered by the distance to the lines Pr_1 and Pr_2:

$$dx_1 = \frac{v_{(A-B)x} + tg(\alpha+\beta) v_{(A-B)y}}{\left(tg(\alpha+\beta)\right)^2 + 1}, \quad (5)$$

$$dy_1 = tg(\alpha+\beta)\frac{v_{(A-B)x} + tg(\alpha+\beta) v_{(A-B)y}}{(tg(\alpha+\beta))^2 + 1} - v_{(A-B)y}, \quad (6)$$

$$d_1 = \sqrt{dx_1^2 + dy_1^2}, \quad (7)$$

$$dx_2 = \frac{v_{(A-B)x} + tg(\alpha-\beta) v_{(A-B)y}}{\left(tg(\alpha-\beta)\right)^2 + 1}, \quad (8)$$

$$dy_2 = tg(\alpha - \beta)\frac{v_{(A-B)x} + tg(\alpha-\beta)v_{(A-B)y}}{(tg(\alpha-\beta))^2+1} - v_{(A-B)y}, \tag{9}$$

$$d_2 = \sqrt{dx_2^2 + dy_2^2}, \tag{10}$$

$$d(dx, dy) = \min\big(d_1(dx_1, dy_1), d_2(dx_2, dy_2)\big). \tag{11}$$

Region $ORCA_{A|B}$ here is determined by the equation of a line parallel to the selected boundary (for example, $Pr_1(x, y)$) transferred to a point with coordinates $\big(v_{Ax} + dx, v_{Ay} + dy\big)$:

$$ORCA_{A|B}(x, y) = (y - (v_{Ay} + dy)) - tg(\alpha + \beta)(x - (v_{Ax} + dx)). \tag{12}$$

In the event that after the fulfillment of the first condition the second is not fulfilled, we should continue to consider other conditions that allow us to clarify the relation of point $\big(v_{(A-B)x}, v_{(A-B)y}\big)$ to region $VO^{\tau}_{A|B}$.

Equation $Circle_\tau(x, y)$ is an equation of a circle with a radius $(R_A + R_B)/\tau$ a center at point $(P_A - P_B)/\tau$, which means it is described by the expression:

$$Circle_\tau(x, y) = (R_A/\tau + R_B/\tau)^2 - (x + x_B/\tau)^2 - (y - y_B/\tau)^2. \tag{13}$$

A part of this circle that is not included in the region $VO^{\tau}_{A|B}$ continues to describe its boundary. Therefore, the expression $Circle_\tau(v_{(A-B)x}, v_{(A-B)y}) \geq 0$ provides the third condition for belonging to the region $VO^{\tau}_{A|B}$ as confirmation of the belonging of the point $(v_{(A-B)x}, v_{(A-B)y})$ to the region of the circle $Circle_\tau(x, y)$.

Under the third condition, the closest point on the boundary for point $(v_{(A-B)x}, v_{(A-B)y})$ is a point on a circle with radius $(R_A + R_B)/\tau$. To determine the direction to such a point as the direction of the normal, it suffices to express a straight line orthogonal to a straight line passing through points $(v_{(A-B)x}, v_{(A-B)y})$ and $(x_B/\tau, y_B/\tau)$ and transfer to the point of vector v_A shifted by coordinates $(wcos\gamma, wsin\gamma)$, where $\gamma = arctg\frac{(\frac{y_B}{\tau} - (v_{(A-B)y})}{\left(\frac{x_B}{\tau} - (v_{(A-B)x}\right)}$, $w = \frac{(R_A + R_B)}{\tau} - \sqrt{\left(\frac{x_B}{\tau} - (v_{(A-B)x}\right)^2 + (\frac{y_B}{\tau} - (v_{(A-B)y})^2}$. As a result, we have:

$$ORCA_{A|B} = tg\left(\frac{\pi}{2} + \gamma\right)(x - (v_{Ax} + wcos\gamma) - \big(y - (v_{Ay} + wsin\gamma\big). \tag{14}$$

As can be seen from the reasoning, the presented algorithm allows us to describe the situation of mutual coordination of speed for the robot with respect to one of the oncoming robots. The full picture is determined by the intersection of all regions of the ORCA, built for each of the robots, affecting the adjustment of the final direction and speed. The obtained region of possible solutions requires the use of optimization tools to obtain the only true velocity vector v_A. Any optimization statement has an objective function, which is expressed by the main direction to the goal. Such a statement is completely solvable by linear mathematical programming [7].

To verify the implementation of the calculation of the region on a specific example, the region $VO^{\tau}_{A|B}$ was simulated in the functional-voxel modeling system RANOK. For this, the R-functional apparatus of set-theoretic operations on functions [15] was applied to the description of regions, which made it possible to distinguish regions with a positive sign of values:

$$VO_{A|B} = Pr_1 \wedge Pr_2 \wedge (Pr_3 \vee Pr_4) \vee Circle_{\tau}. \quad (15)$$

Figure 4 demonstrates the union in the space of velocities of region $VO^{\tau}_{A|B}$ and region $ORCA_{A|B}$ visually demonstrating the result of calculations for the task.

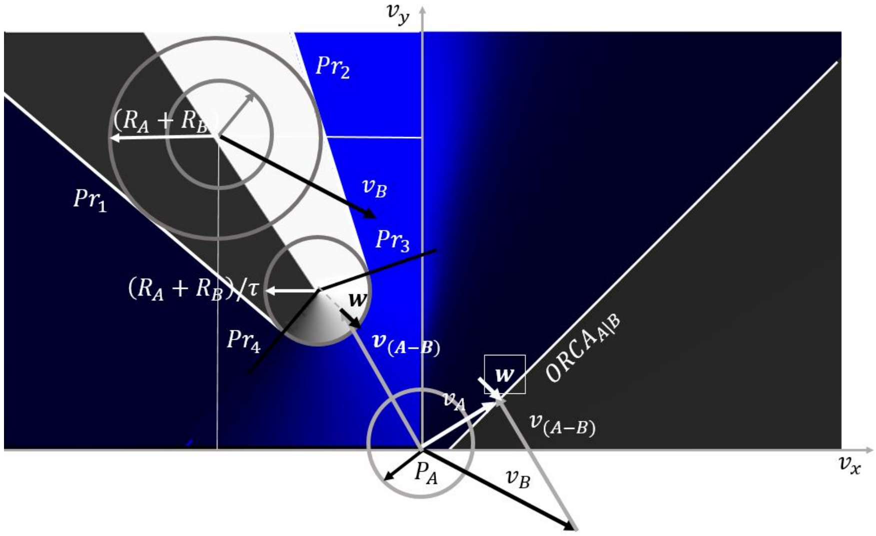


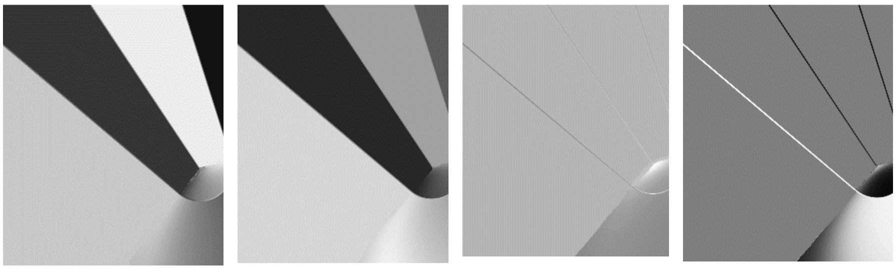
Fig. 4. Construction of a graphic M-image based on equation (15) and formation of a half-plane $ORCA^{\tau}_{A|B}$

Figure 5 presents graphical image models (M-images) describing the local equation at the points of the collision region. M-images are one of the key tools for graphical analysis of the functional-voxel modeling method [16]. Each of these images characterizes the normal component built in 4-dimensional space. To describe such a local equation, it is necessary to represent the normal vector at each point in the space in the 4th dimension.

To obtain the angle of inclination of region $ORCA^{\tau}_{A|B}$ it is sufficient to consider the component of the two-dimensional normal vector on the X axis at the point of application of the velocity difference vector, i.e. $cos\alpha\ _2$ renormalized by the formula:

$$cos\alpha\ _4 x + cos\beta\ _4 y + cos\gamma\ _4 z = cos\delta\ _4, \quad (16)$$

$$cos\alpha\ _2 = \frac{cos\alpha\ _4}{\sqrt{cos\alpha\ _4{}^2 + cos\beta\ _4{}^2}} = cos\alpha\ _{ORCA}. \quad (17)$$



$cos\alpha_4$ $cos\beta_4$ $cos\gamma_4$ $cos\delta_4$

Fig. 5. M-images describing the local equation at the points of the collision region

5 Modeling collision region in three-dimensional space

To simulate such a case in three-dimensional space, it is necessary to describe the collision region in the form of a figure of rotation around a certain axis, for example, OX, the generators of which are two straight lines Pr_1 and Pr_2, combined with the region of the sphere completing this figure:

$$Pr_1 = tg\beta x' - \sqrt{y'^2 + z'^2}, \tag{18}$$

$$Pr_2 = tg\left(\frac{\pi}{2} + \beta\right)(x' - |P_A - P_B|) - \sqrt{y'^2 + z'^2}. \tag{19}$$

To determine the parameters of the spatial transfer of the figure to the required position, as shown in Figure 6, it is enough to determine the parameters of the spherical coordinates of the position point of the robot:

$$|P_A - P_B| = \sqrt{x^2_{(P_A-P_B)} + y^2_{(P_A-P_B)} + z^2_{(P_A-P_B)}}. \tag{20}$$

The rotation angles θ and φ are calculated as follows:

$$\theta = \begin{cases} arctg\left(\frac{y_{(P_A-P_B)}}{x_{(P_A-P_B)}}\right) & \to x_{(P_A-P_B)} > 0 \\ \pi + arctg\left(\frac{y_{(P_A-P_B)}}{x_{(P_A-P_B)}}\right) & \to x_{(P_A-P_B)} < 0 \\ \frac{\pi}{2} & \to x_{(P_A-P_B)} = 0, y_{(P_A-P_B)} \geq 0 \\ \frac{3\pi}{2} & \to x_{(P_A-P_B)} = 0, y_{(P_A-P_B)} < 0 \end{cases}, \tag{21}$$

$$\varphi = arccos\left(\frac{z(P_A-P_B)}{|P_A-P_B|}\right). \quad (22)$$

The matrix for recalculating the coordinates of the position of the region will take the form:

$$[x' \quad y' \quad z'] = [x \quad y \quad z]R_z R_y, \quad (23)$$

$$R_z = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_y = \begin{bmatrix} sin\varphi & 0 & cos\varphi \\ 0 & 1 & 0 \\ -cos\varphi & 0 & sin\varphi \end{bmatrix}. \quad (24)$$

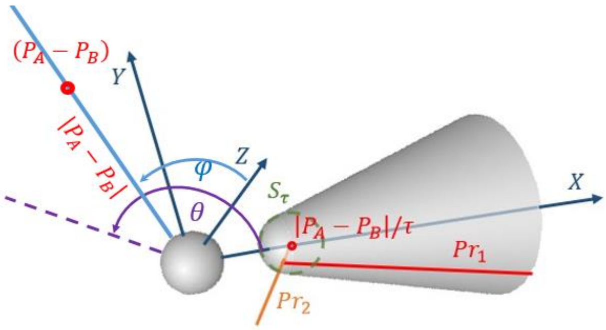


Fig. 6. Calculation of the displacement of collision region $VO_{A|B}$ in a 3D environment

If in a 2D environment the region of possible velocities $ORCA^{\tau}_{A|B}$ is half-plane, then in 3D such a region will be a half-space. Figure 7 shows a cross section of one of the components of the three-dimensional vector w, which determines the slope of the position of the half-space $ORCA^{\tau}_{A|B}$ in the 3D environment.

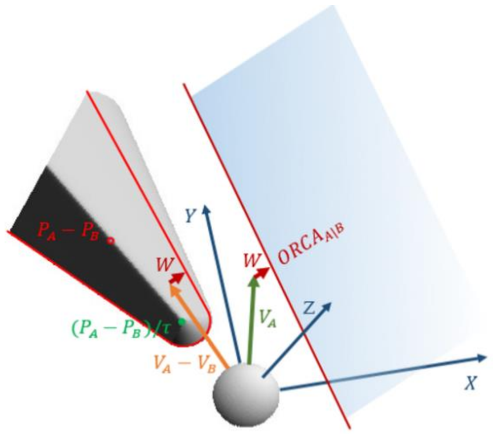


Fig. 7. Construction of the region of permissible speeds $ORCA^{\tau}_{A|B}$ in a 3D environment

6 Conclusion

The conducted studies proved the possibility of modeling the analytical tools of multi-agent motion algorithms using the functional-voxel method. The resulting functional-voxel model acts as a template that carries complete information about the local geometric characteristics at each point in space. This information allows one to easily determine the necessary direction of the region of admissible velocities $\boldsymbol{ORCA}^{\tau}_{\boldsymbol{A|B}}$. Using the presented approaches to the formation of the motion environment in multi-agent systems will allow us to abandon a number of numerical calculations in comparison with the classical description of the algorithm, and most importantly, it allows us to consider such problems without reference to the dimension of space (Fig. 7).

References

1. Timofeev, A.V., Yusupov, R.M.: Principy postroeniya integrirovannyh sistem mul'tiagentnoj navigacii i intellektual'nogo upravleniya mekhatronnymi robotami [Principles of building integrated systems of multi-agent navigation and intelligent control of mechatronic robots]. Information Technologies & Knowledge, 5(3), 237-244 (2011) (in Russian)
2. Dergachev, S.A.: Eksperimental'noe issledovanie reaktivnogo algoritma navigacii dlya grupp agentov – ORCA [Experimental study of reactive navigation algorithm for groups of agents – ORCA]. 17th Russian Conference on Artificial Intelligence**,** Ulyanovsk, Russian Federation. P.102 (2019)
3. Yakovlev, K.S., Baskin, E.S, Andreychuk, A.A.: Metod avtomaticheskogo planirovaniya sovokupnosti traektorij dlya navigacii bespilotnyh transportnyh sredstv [Dynamics constraint-aware planning of multiple paths for unmanned vehicle]. Journal Large-Scale Systems Control 58, 306-342 (2015) (in Russian)
4. Van Den Berg, J., Guy, S. J., Lin, M., & Manocha, D.: Reciprocal n-body collision avoidance. In Robotics research (pp. 3-19). Springer, Berlin, Heidelberg. (2011)
5. Tolok, A. V. Funkcional'no-voksel'nyj metod v komp'yuternom modelirovanii [Functional voxel method in computer modeling]. Moscow, 112 p. (2016)
6. Vassilyev, S. N., Loktev, M. A., Tolok, A. V., Tolok, N. B., Ul'yanov, S. A.: Route planning in 3D environment with a multivariant model. Trudy SPIIRAN, *45*, 5-25 (2016)
7. Tolok, A. V., Tolok, N. B.: Mathematical programming problems solving by functional voxel method. Automation and Remote Control, *79*(9), 1703-1712. (2018)
8. P. Fiorini, Z. Shiller.Motion planning in dynamic environments using Velocity Obstacles. Int. Journal of Robotics Research 17(7), 760-772 (1998)
9. Lake, A. T., Snape, J., Guy, S. J., Vembar, D., Lake, A., Lin, M. C., Manocha, D.: Reciprocal collision avoidance and navigation for video games (2012)
10. Schaefer, M.: Planning and coordination in driving simulation. Acta Polytechnica CTU Proceedings, 2(2), 40-44 (2015)
11. Mao, R., Gao, H., Guo, L: A Novel Collision-Free Navigation Approach for Multiple Nonholonomic Robots Based on ORCA and Linear MPC. Mathematical Problems in Engineering (2020)
12. Wolinski, D., Lin, M. C.: Generalized WarpDriver: Unified Collision Avoidance for Multi-Robot Systems in Arbitrarily Complex Environments. In Robotics: Science and Systems. (2018)

13. Loktev M. A. Osobennosti primeneniya funkcional'no-voksel'nogo modelirovaniya v zadachah poiska puti s prepyatstviyami [Features of application of functional voxel modeling in problems of finding a path with obstacles]. Informacionnye tekhnologii v proektirovanii i proizvodstve [Information technologies in design and production] I. 1, 45-49 (2016) (in Russian)
14. Grigor'ev, S. N., Tolok, A. V., Tolok, N. B.: Local search gradient algorithm based on functional voxel modeling. Programming and Computer Software, 43(5), 300-306 (2017)
15. Rvachev V. L. Teoriya R-funkcij i nekotorye ee prilozheniya [Theory of R-functions and some of its applications]. Kiev, 552 p.(1982)
16. Tolok A. V. Graficheskie obrazy-modeli v informacionnyh tekhnologiyah [Graphic images-models in information technologies]. Prikladnaya informatika [Applied informatics] I. 4, 31-40 (2009) (in Russian).