FUNCTIONAL VOXEL MODELING OF A PATH PLANNING ALGORITHM TO A TARGET BASED ON *R*-FUNCTIONS

A. V. Tolok* and N. B. Tolok**

*,**Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

*✉ tolok_61@mail.ru, **✉ nat_tolok@mail.ru

Abstract. This paper is devoted to analytical approaches to path planning with obstacles. Two analytical modeling principles are compared for obstacles in a scene: the methods of potentials and *R*-functional modeling. The functional voxel design principle of complex computational processes is presented on an illustrative example of modeling of the *R*-function for the union/intersection of the domains of two functions. The fundamentals of arithmetic operations over local geometrical characteristics describing the components of a homogeneous unit vector of a local function are discussed. The denormalization principle of such components is demonstrated for application in arithmetic operations constituting an *R*-function. The scene is modeled by the layout of concentric objects and a local function describing the target by a funnel surface at a given point. A dynamic formation algorithm is considered for the final local function of the union of the funnel and scene surfaces at a current point. The final local function is used to determine the components of the direction vector of gradient-based motion to the target.

Keywords: *R*-function, functional voxel modeling, local geometrical characteristic, local function, gradient, homogeneous vector, path planning with obstacles.

INTRODUCTION

The path planning problem statement already has many solutions for detecting spatial obstacles and constructing an avoiding route [1–3]. Most path planning algorithms are based on the "moving-until-stop" principle due to the lack of information necessary for analysis at each point. Among them, we mention heuristic algorithms, particularly the ant colony algorithm, the particle swarm method, and the evolutionary genetic algorithm [4], which often yield a path without smoothness and cannot be used in several applications requiring a smooth motion trajectory. This class of algorithms also includes the wave algorithm and the A* algorithm. Therefore, special attention of researchers is paid to analytical approaches to describing the environment to avoid obstacles. Such a method allows describing geometrical models with maximum accuracy and displaying the geometrical properties of environment objects for each point in a given domain of a scene.

The analytical approach to path planning with a complex configuration of obstacles is traditionally reduced to constructing and analyzing the surface relief of a function $f(x, y) = z$, where the set of points $f(x, y) = 0$ defines the boundary of an obstacle to be avoided. According to the studies of relief construction methods, there are two main approaches: the *R*-functional construction of the surface relief to design gradient-based motion (*R*-functional modeling) and the construction of the potential field as a sum of point sources of distributed potential (the method of potentials).

The method of potentials considers the sum of "hyperboloidal" functions of movement to a target and repulsion from separate points (obstacle objects). The method of *R*-functional modeling (RFM) [5–7] provides a unified functional description of a complex geometry of a scene with obstacles based on the *R*-functional union of "paraboloidal" functions. There is no need to perform complex calculations when constructing the definitional domain of the potentials: it suffices to sum their values sequentially. In *R*-functional modeling, union *R*-functions are used to include the next object in the scene. In comparison with the method of potentials, this fact appreciably increases the calculation time but allows describing objects of complex geometrical shape, which cannot be done with the method of potentials. Therefore, it is recommended to apply both methods: the *R*-functional model can be used to describe a static deterministic environment of complex geometry, whereas potentials can be dynamically added anywhere in the scene by summing them with the *R*-functional surface.

1. PROBLEM STATEMENT

In this paper, we study the possibility of using the functional voxel model (FM-model, in particular, its gradient properties) for the computer representation of the domain of an *R*-function in path planning with obstacle avoidance. Note that the optimal (shortest) path requirement is not intentionally imposed here: the principle of flowing around objects often makes the agent occluded, for a long time, with respect to the terminal point, and only the departure to the open space will require aligning its motion to the target. These are the properties of local gradient-based motion optimization (Fig. 1).

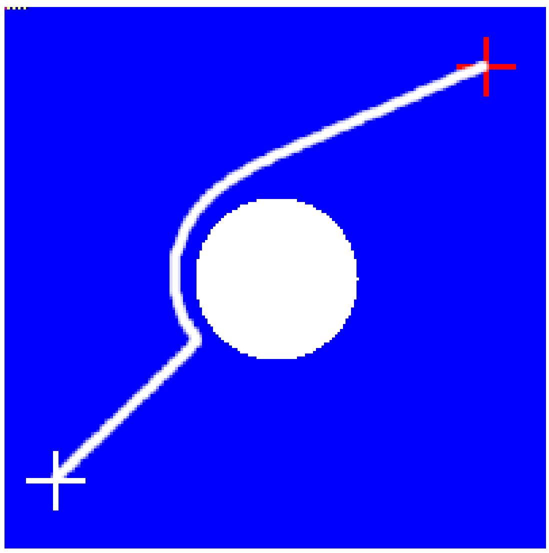

Fig. 1. Gradient-based motion to a target with obstacle avoidance (the red cross indicates the target and the white cross the agent).

We assert that the FV-model allows performing arithmetic operations over the values of two different functions defined by a single domain [8–10]. In other words, the assertion is that functional voxel modeling (FVM) [10] allows implementing complex algebraic expressions based on arithmetic operations at the level of calculating local geometrical characteristics. As an illustrative example, let us consider the FVM of the intersection/union *R*-function $\omega(x, y)$ as a function whose arguments are also the values of functions $\omega_1(x, y)$ and $\omega_2(x, y)$ [5–7]:

$$\omega(x, y) = \omega_1(x, y) + \omega_2(x, y) \mp \sqrt{\omega_1^2(x, y) + \omega_2^2(x, y)}. \quad (1)$$

The expression (1) includes the following arithmetic constructs: sum, difference, raising to a power, and root extraction for function(s). We study each of these procedures in detail.

The FVM principle involves the computer representation of a given domain of an initial complex function $F(X_{m+1}) = 0$ by local functions $L(A_{m+1}, X_m) = 0$, where A_i is a local geometrical characteristic showing the deviation of the unit homogeneous normal vector at a current point. This characteristic is encoded by an appropriate color of the *m*-dimensional image. A local function is linear and decomposes into a polynomial:

$$L(A_{m+1}, X_m) = a_1x_1 + a_2x_2 + ... + a_mx_m + a_{m+1} = 0.$$

Consider the necessary arithmetic operations included in the *R*-function to calculate its local geometrical characteristics at the points of a given domain. As an example, we define a two-dimensional domain with two local functions at each point (x, y):

$$L_1(x, y, z) = a_1^{(1)}x + a_2^{(1)}y + a_3^{(1)}z + a_4^{(1)} = 0,$$
$$L_2(x, y, z) = a_1^{(2)}x + a_2^{(2)}y + a_3^{(2)}z + a_4^{(2)} = 0. \quad (2)$$

The superscript in the notation of a local geometrical characteristic, enclosed in brackets, describes the function number. To simplify the calculations, we divide all the equation coefficients, including the free one, by that at the argument *z*:

$$L_1(x, y, z) = \frac{a_1^{(1)}}{a_3^{(1)}}x + \frac{a_2^{(1)}}{a_3^{(1)}}y + \frac{a_3^{(1)}}{a_3^{(1)}}z + \frac{a_4^{(1)}}{a_3^{(1)}}$$
$$= l_1^{(1)}x + l_2^{(1)}y + z + l_4^{(1)} = 0,$$
$$L_2(x, y, z) = \frac{a_1^{(2)}}{a_3^{(2)}}x + \frac{a_2^{(2)}}{a_3^{(2)}}y + \frac{a_3^{(2)}}{a_3^{(2)}}z + \frac{a_4^{(2)}}{a_3^{(2)}}$$
$$= l_1^{(2)}x + l_2^{(2)}y + z + l_4^{(2)} = 0.$$

As a rule, the value of the components of the homogeneous vector grows, causing its denormalization (increasing the length of the normal vector).

To obtain the values of the local characteristics l_i for the sum of arguments $z = z^{(1)} + z^{(2)}$, it suffices to add all the equation characteristics in pairs. We prove this fact as follows:

$$z = z^{(1)} + z^{(2)}$$
$$= (-l_1^{(1)}x - l_2^{(1)}y - l_4^{(1)}) + (-l_1^{(2)}x - l_2^{(2)}y - l_4^{(2)})$$
$$= -(l_1^{(1)} + l_1^{(2)})x - (l_2^{(1)} + l_2^{(2)})y - (l_4^{(1)} + l_4^{(2)}).$$

By analogy, for the subtraction procedure,

$$z = z^{(1)} - z^{(2)}$$
$$= (-l_1^{(1)}x - l_2^{(1)}y - l_4^{(1)}) - (-l_1^{(2)}x - l_2^{(2)}y - l_4^{(2)})$$
$$= -(l_1^{(1)} - l_1^{(2)})x - (l_2^{(1)} - l_2^{(2)})y - (l_4^{(1)} - l_4^{(2)}).$$

In addition, $l_3^{(1)} = l_3^{(2)} = 1$.

Therefore, the local geometrical characteristics eliminating the coefficient l_3 at the computed argument *z* in the sum or difference produce the local function of addition or subtraction, respectively. In other words, to obtain the terms of operations over local functions, it suffices to add their coefficients.

To preserve the linear property of a local function under the multiplication operation, it suffices to replace one of the functions by the value of the argument *z*, e.g.:

$$z = z^{(1)}z^{(2)}$$
$$= \left(-l_1^{(1)}x - l_2^{(1)}y - l_4^{(1)}\right)\left(-l_1^{(2)}x - l_2^{(2)}y - l_4^{(2)}\right)$$
$$= \left(-l_1^{(1)}x - l_2^{(1)}y - l_4^{(1)}\right)z^{(2)}$$
$$= -\left(l_1^{(1)}z^{(2)}\right)x - \left(l_2^{(1)}z^{(2)}\right)y - \left(l_4^{(1)}z^{(2)}\right).$$

The value of the sub-root expression can be obtained by utilizing the following property:

$$\sqrt{z^{(1)}} = \frac{z^{(1)}}{\sqrt{z^{(1)}}} = \frac{-l_1^{(1)}x - l_2^{(1)}y - l_4^{(1)}}{\sqrt{-l_1^{(1)}x - l_2^{(1)}y - l_4^{(1)}}}$$
$$= \frac{-l_1^{(1)}x - l_2^{(1)}y - l_4^{(1)}}{\sqrt{z^{(1)}}} = -\frac{l_1^{(1)}}{\sqrt{z^{(1)}}}x - \frac{l_2^{(1)}}{\sqrt{z^{(1)}}}y - \frac{l_4^{(1)}}{\sqrt{z^{(1)}}}.$$

2. *R*-FUNCTION MODELING FOR THE FV-MODEL

To illustrate the modeling process of the intersection *R*-function, we take as input functions $z^{(1)}(x, y)$ and $z^{(2)}(x, y)$.

Let $z^{(1)} = 1 - y^2$, which is the algebraic function of the geometrical object describing a symmetric band of positive values of x along the Ox axis (Fig. 2).

Figure 3 shows the FV model for the function $z^{(1)}$.

Let $z^{(2)} = 1 - x^2$, which is the algebraic function of the geometric object describing a symmetric band of positive values of y along the Oy axis (Fig. 4).

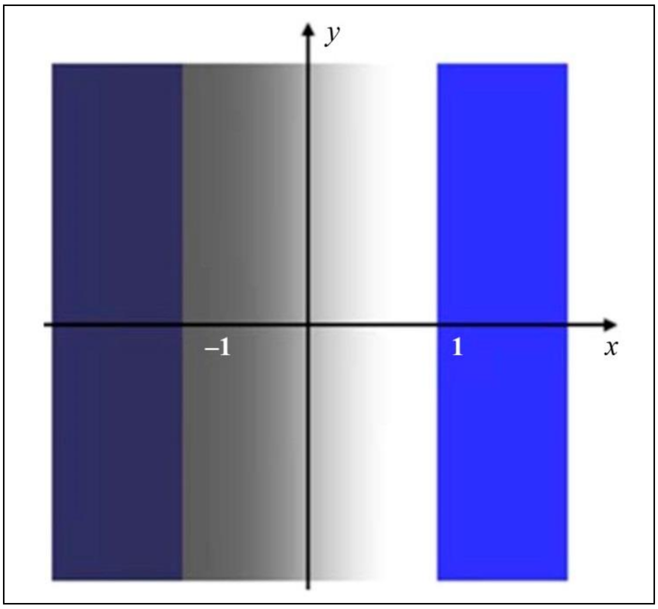

Fig. 2. The voxel representation of the positive domain of the function $z^{(1)}$ (the blue color indicates the negative domain of values).

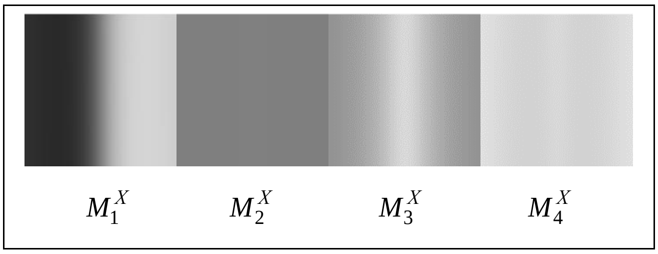

Fig. 3. The voxel representation of the normal components of the function $z^{(1)}$.

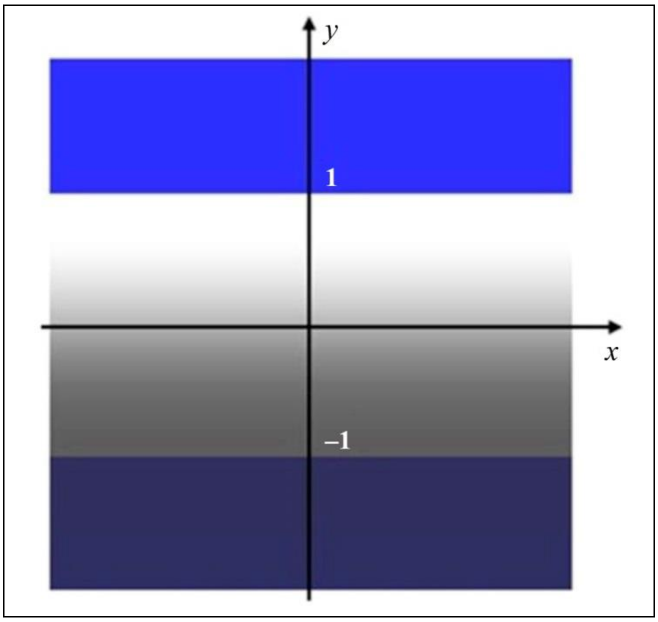

Fig. 4. The voxel representation of the positive domain of the function $z^{(2)}$ (the blue color indicates the negative domain of values).

Figure 5 shows the FV model for the function $z^{(2)}$.

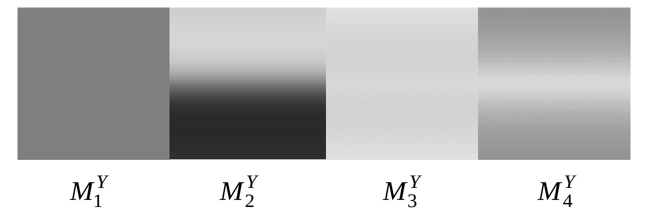

Fig. 5. The voxel representation of the normal components of the function $z^{(2)}$.

The graphical information in Figs. 3 and 5 allows obtaining four local geometrical characteristics at each point of the domain and forming two local functions $L^{(1)}$ and $L^{(2)}$ (2).

We simplify further calculations by dividing all the coefficients of the local equation by the coefficient a_3. In this case, the third coefficient takes a unit value:

$$l_{1,2,4}^{(1)} = \frac{a_{1,2,4}^{(1)}}{a_3^{(1)}},\ l_{1,2,4}^{(2)} = \frac{a_{1,2,4}^{(2)}}{a_3^{(2)}},\ l_3^{(1)} = l_3^{(2)} = 1.$$

The values of z for both functions are obtained based on these coefficients:

$$z^{(1)} = -l_1^{(1)}x - l_2^{(1)}y - l_4^{(1)},\ z^{(2)} = -l_1^{(2)}x - l_2^{(2)}y - l_4^{(2)}.$$

The final law for determining the local geometrical characteristics of the intersection/union *R*-function is given by

$$l_i^R = l_i^{(1)} + l_i^{(2)} \mp \left(\frac{l_i^{(1)} z^{(1)} + l_i^{(2)} z^{(2)}}{\sqrt{\left(z^{(1)}\right)^2 + \left(z^{(2)}\right)^2}} \right),\ l_3^R = 1,\ i = 1, 2, 4.$$

By normalizing the resulting coefficients l_i^R using the four-component norm $N_4 = \sqrt{\left(l_1^R\right)^2 + \left(l_2^R\right)^2 + \left(l_3^R\right)^2 + \left(l_4^R\right)^2}$ and matching them to the monochromatic gradient palette, we get the required four *M*-images of the intersection function (Fig. 6).

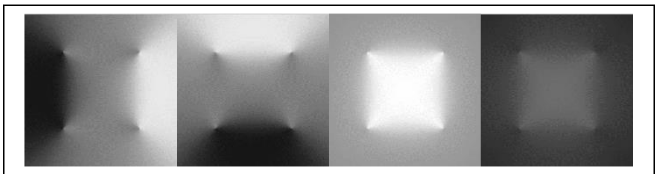

Fig. 6. The voxel representation of local geometrical characteristics on images M_i.

Figure 7 demonstrates the modeling result for the positive *z*-domain of a square with side 2.

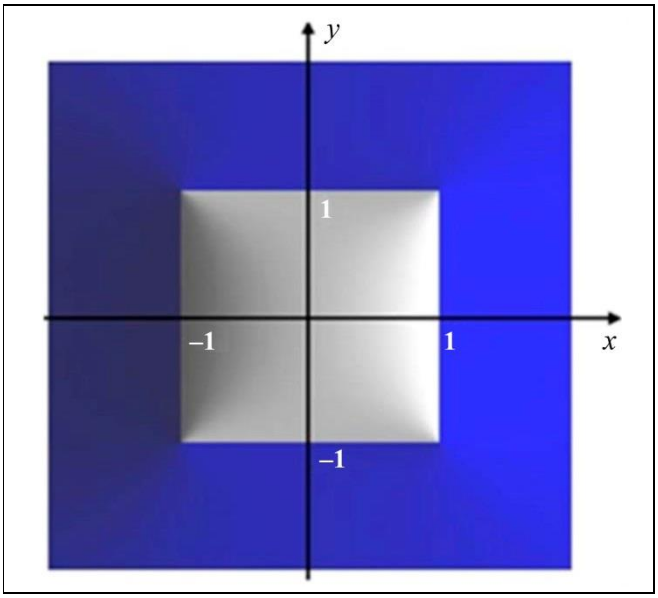

Fig. 7. The voxel representation of the positive domain of the function $L^{(1)} \wedge L^{(2)}$ (the blue color indicates the negative domain of values).

3. THE PATH PLANNING ALGORITHM TO A GIVEN TARGET IN A DETERMINISTIC ENVIRONMENT

To apply FVM principles, it is necessary to formalize the description of a scene in which the agent will act and describe the law of movement to a given point (target). The target is dynamically set during the operation of the future algorithm; in contrast, the scene may have various complex configurations and is prepared in advance in a special system by the algorithm presented in [11, 12].

As an example, we describe the scene by a function on a given domain of sizes 20 × 20, with the origin placed in the center. The scene function contains three concentric objects of unit radius; their arrangement is shown in Fig. 8. The formalization of the scene function contains a nested structure of the following functions:

$$z^S = z^{(3)} + z^{(12)} + \sqrt{\left(z^{(3)}\right)^2 + \left(z^{(12)}\right)^2},$$

$$z^{(12)} = z^{(1)} + z^{(2)} + \sqrt{\left(z^{(1)}\right)^2 + \left(z^{(2)}\right)^2},$$

$$z^{(1)} = 1 - (x+4)^2 - (y-2)^2,$$

$$z^{(2)} = 1 - (x-2)^2 - (y+4)^2,$$

$$z^{(3)} = 1 - (x-4)^2 - (y-4)^2.$$

To keep the previously accepted notation (the object number indicated by the superscript), we will put an element with a superscript in brackets when raising it to the second power.

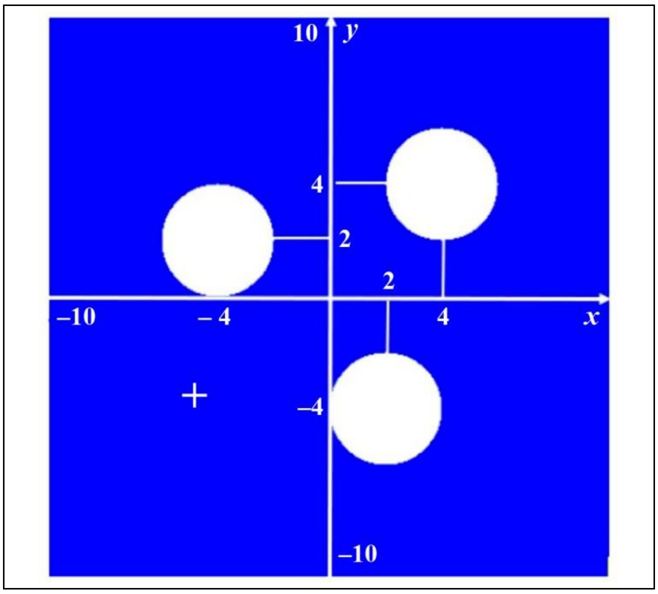

Fig. 8. The scene described by the function $z^S(x, y)$.

To set the target, it is necessary to specify in the scene a point with coordinates (x_C, y_C); see the cross in Fig. 8. Now any other point of the scene must lie on the surface of the function of the inverted cone with the vertex point (x_T, y_T). Uniting the function of this cone with the scene function z^S yields the surface function z^G of the problem solution (Fig. 9). Any point moving along the gradient of the surface will tend to the point (x_T, y_T), avoiding the concentric objects.

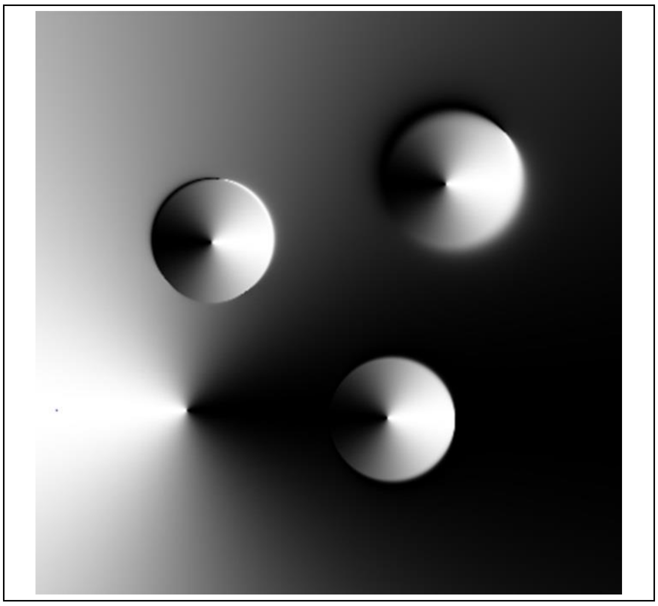

Fig. 9. The scene described by the function $z^G(x, y)$.

When constructing the FV-model, at each point of the domain, we obtain a local function L^S with local geometrical characteristics a_1^S, a_2^S, a_3^S, and a_4^S instead of the structurally nested function $z^S(x, y)$. Therefore, it makes sense to form the target function using local geometrical characteristics a_1^T, a_2^T, a_3^T, and a_4^T.

To this end, it is necessary to specify some current point (x_A, y_A) of the agent's position in the scene. For this point, we define the local geometrical characteristics of the target's local function $L^T(x_A, y_A)$. Let $z_T = -0.001$ be the level of the target point (i.e., the height of the inverted cone). In other words, the cone surface is maximally unfolded with respect to the plane xOy. Then the distance between the agent's current point and the target point is given by

$$d = \sqrt{(x_T - x_A)^2 + (y_T - y_A)^2 + (2z_T)^2},$$

and the local geometrical characteristics at the agent's point are expressed as follows:

$$a_1^T = \frac{x_T - x_A}{d},$$

$$a_2^T = \frac{y_T - y_A}{d},$$

$$a_3^T = \frac{2z_T}{d},$$

$$a_4^T = -a_1^T x_T - a_2^T y_T - a_3^T \cdot 10^{-3}.$$

We normalize the characteristics a_1^T, a_2^T, a_3^T, and a_4^T:

$$N_4^T = \sqrt{(a_1^T)^2 + (a_2^T)^2 + (a_3^T)^2 + (a_4^T)^2},$$

$$n_i^T = \frac{a_i^T}{N_4^T}, \quad i = 1, \ldots, 4.$$

These characteristics n_1^T, n_2^T, n_3^T, and n_4^T, as well as those a_1^S, a_2^S, a_3^S, and a_4^S of the local function $L^S(x_A, y_A)$ taken from the FV-model of the scene, have to be prepared to model the resulting local function $L^G(x_A, y_A)$ by the union construct:

$$l_{1,2,4}^T = \frac{n_{1,2,4}^T}{n_3^T}, \; l_{1,2,4}^S = \frac{a_{1,2,4}^S}{a_3^S}, \; l_3^T = l_3^S = 1.$$

We find the local geometrical characteristics of the local function $L^G(x_A, y_A)$:

$$z^T = -l_1^T x - l_2^T y - l_4^T, \; z^S = -l_1^S x - l_2^S y - l_4^S.$$

$$l_i^G = l_i^T + l_i^S + \left(\frac{l_i^T z^T + l_i^S z^S}{\sqrt{\left(z^T\right)^2 + \left(z^S\right)^2}} \right),$$

$$l_3^G = 1,\ i = 1,\ 2,\ 4.$$

The local geometrical characteristics at the point of the solution surface $z^G(x_A, y_A)$ being obtained, we determine the next point of the directed gradient-based motion segment with a certain step *S*. To find the cosines of the deviation of the normal vector in the plane xOy, it is necessary to reduce the two local geometrical characteristics to the two-dimensional norm $N_2^G = \sqrt{\left(l_1^G\right)^2 + \left(l_2^G\right)^2}$:

$$n_i^G = \frac{l_i^G}{N_2^G},\ i = 1,\ 2.$$

The next step is to calculate the coordinates of the new current point of the agent's position in the scene:

$$x'_A = x_A + Sn_1^G,\ y'_A = y_A + Sn_2^G.$$

Figure 10 demonstrates the operation of the gradient-based motion algorithm from the agent's position to the target point with avoiding the three concentric obstacles. The red cross indicates the target whereas the white cross the agent.

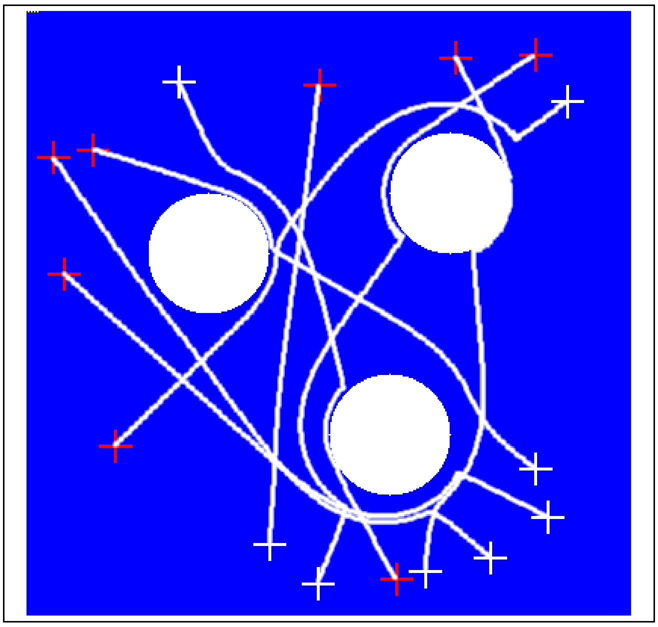

Fig. 10. Path planning for an agent in a scene with three obstacles.

Note that the FV-model of the scene is described and modeled separately, and the speed of the algorithm does not depend on the number and size of the objects placed in it.

Four images describing the scene function in the same format (a set of local functions) are the input of this algorithm, which makes the latter independent of the complexity of the scene. The speed of such an algorithm depends on the representation accuracy of the FV-model (image resolution) and the parameter *S* (the step of gradient-based motion).

Figure 11 shows the gradient-based motion to the target in a scene with 25 regularly distributed concentric objects.

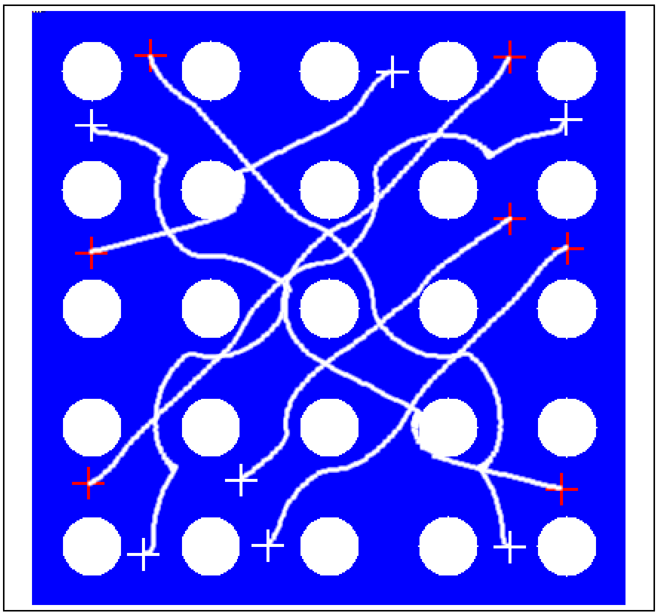

Fig. 11. Path planning for an agent in a scene with 25 obstacles.

Finally, Fig. 12 presents gradient-based motion to the target in a scene of complex geometry.

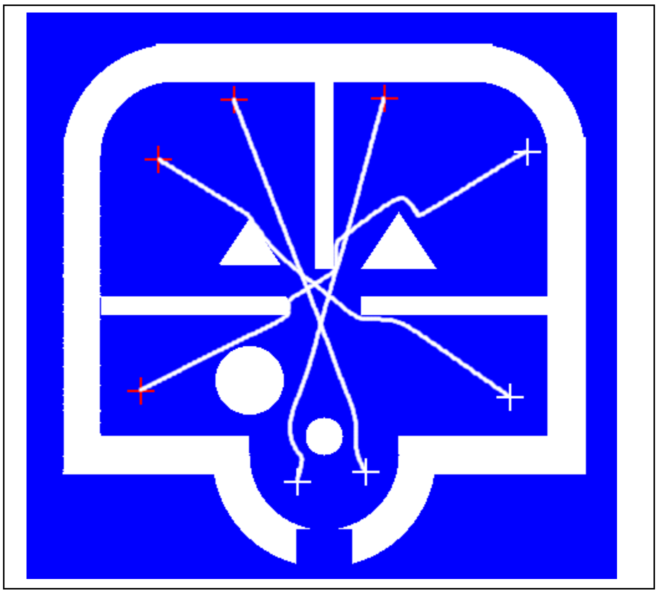

Fig. 12. Path planning for agents in a scene of complex geometry.

CONCLUSIONS

In this paper, we have presented a new FVM tool that provides set-theoretic operations over local functions constituting a rectangular domain. It allows operating the data of object's local geometry when implementing the description of complex analytical constructs on a PC; these data are further applied in dynamic algorithms of analytical optimization schemes. With the gradient-based motion algorithm, the modeling parameters of the cone surface of the motion are chosen to change the agent's path by introducing objects of complex shape; as a result, the agent surely arrives at the target. Using the FV-model, it is possible to introduce a new object in the scene, such as the target's cone surface, for dynamic control of the agent's motion in a complex geometrical environment.

REFERENCES

1. Yushchenko, A.S., Intelligent Planning in Robot Activity, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2005, no. 3, pp. 5–18. (In Russian.)
2. Liu, W., Path Planning Methods in an Environment with Obstacles (A Review), *Mathematics & Mathematical Modeling*, 2018, no. 01, pp. 15–58. (In Russian.)
3. Vassilyev, S.N., Loktev, M.A., Tolok, A.V., and Tolok, N.B., Route Planning in 3D Environment with Multivariant Model, *SPIIRAS Proceedings*, 2016, vol. 2, no. 45, pp. 5–25. (In Russian.)
4. *Intellektual'noe planirovanie traektorii podvizhnykh ob"ektov v sredakh s prepyatstviyami* (Intelligent Path Planning for Moving Objects in Environments with Obstacles), Pshikhopov, V.Kh., Ed., Moscow: FIZMATLIT, 2014. (In Russian.)
5. Maksimenko-Sheiko, K.V., *R-funktsii v matematicheskom modelirovanii geometricheskikh ob"ektov i fizicheskikh polei* (R-functions in Mathematical Modeling of Geometric Objects and Physical Fields), Kharkiv: Podgorny Institute for Problems in Mechanical Engineering, National Academy of Sciences of Ukraine, 2009. (In Russian.)
6. Rvachov, V.L., *Geometricheskie prilozheniya algebry logiki* (Geometric Applications of Logic Algebra), Kiev: Tekhnika, 1967. (In Russian.)
7. Rvachov, V.L. and Sheiko, T.I., An Introduction to the Theory of R-functions, *Problemy Mashinostroeniya*, 2001, vol. 4, no. 1-2, pp. 46–58. (In Russian.)
8. Tolok, A.V., *Lokal'naya komp'yuternaya geometriya* (Local Computer Geometry), Moscow: IPR-Media, 2022. (In Russian.)
9. Tolok, A.V., *Funktsional'no-voksel'nyi metod v komp'yuternom modelirovanii* (The Functional Voxel Method in Computer Modeling), Moscow: Fizmatlit, 2016. (In Russian.)
10. Tolok, A.V. and Tolok, N.B., Arithmetic in Functional-Voxel Modeling, *Scientific Visualization*, 2022, vol. 14, no. 3, pp. 107–121. URL: http://sv-journal.org/2022-3/08/en.pdf.
11. Masłowski, A., Training in Military Robotics and EOD Unmanned Systems, *NATO EOD Demonstrations and Trials 2014*, Trenčín, 2014. URL: https://www.eodcoe.org/files/en/events/nato-eod-demonstrations-trials-2014/3-nato_eod_trencin-09_2014-maslowski-opt.pdf.
12. Zhong, M., Yang, Y., Dessouky, Y., and Postolache, O., Multi-AGV Scheduling for Conflict-Free Path Planning in Automated Container Terminals, *Comput. Ind. Eng.*, 2020, vol. 142. art. no. 106371.

This paper was recommended for publication by E.Ya. Rubinovich, a member of the Editorial Board.

Received September 16, 2024, and revised October 8, 2024.
Accepted October 24, 2024.

Author information

Tolok, Alexey Vyacheslavovich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ tolok_61@mail.ru
ORCID iD: https://orcid.org/0000-0002-7257-9029

Tolok, Nataliya Borisovna. Cand. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ nat_tolok@mail.ru
ORCID iD: https://orcid.org/0000-0002-5511-4852

Cite this paper

Tolok, A.V. and Tolok, N.B., Functional Voxel Modeling of a Path Planning Algorithm to a Target Based on *R*-Functions. *Control Sciences* **5,** 41–47 (2024).

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com