Development of a Gradient Descent Algorithm for Pathway Routing Based on Functional-Voxel Modeling

Alexey Tolok [1] and Pavel Petukhov [2]

[1] *V.A. Trapeznikov Institute of Control Science of Russian Academy of Sciences, 65 Profsoyuznaya street, Moscow, 117997, Russia*
[2] *Moscow State University of Technology "STANKIN", 1 Vadkovsky street, Moscow, 127994, Russia*

Abstract

This paper considers a pathfinding algorithm using the gradient method for functional-voxel modeling problems. The basic principles of constructing gradient lines based on the functional voxel model are investigated. The tool to describe the scene with obstacles is the mathematical apparatus of R-functions. To solve the problem of pathfinding in a weakly deterministic environment, we propose an algorithm that is based on the use of: gradient method analyzing the color palette of images of local features of the function; mathematical apparatus of R-functions describing the topography of the solution surface at the current moment in time. An algorithm of target control that solves the problem of getting out of possible "trapped objects" is considered. For this, the principle of changing the position of the target was developed to control the gradient direction.

Keywords

Gradient descent, function-voxel modeling, R-function modeling, path finding with obstacles.

1. Introduction

To date, there are various approaches for solving problems associated with pathfinding in a weakly deterministic environment, where information about obstacles is partially defined. In most cases, they are reduced to the task of controlling the object, as well as the possibility of dynamic control when obstacles arise. Thus, to achieve the goal is the task of circumventing known obstacles, as well as obstacles arising in the process of movement. To solve the problem, different approaches to path construction are used. So, for example, using an algorithm based on the simulation of potential fields allows a dynamic response to obstacles that arise in the process of moving towards the goal. This approach is based on the application of "attractive" and "repulsive" potentials. As a result, at each step the potential field is calculated. This method is characterized by the fact that it is forced to constantly recalculate the dynamically changing environment. The disadvantage is that the execution time of the algorithm depends on the number of obstacles that fill the scene.

Functional-voxel method (FV-method), based on the representation of gradient values on a given area of a scalar function. Therefore, it is based on the gradient descent algorithm, which uses special M-images. The first studies of the principles of application of the gradient descent of the FV-method in the problem of laying the way were carried out by M.A. Loktev in papers [1, 2]. In the formulation of the pathfinding problem, the FV-method creates an imaginary representation of the gradient data, which are calculated based on the description of the scene in the form of the surface of the analytical function. At the same time, the scene must contain as objects the target and obstacles. The basis of the gradient motion is a target described by the surface of an inverted cone with the vertex at the target point. On such a surface, from any point of it, the gradient descent motion will tend to its top. The basis of the gradient motion is a target described by the surface of an inverted cone with the vertex at the target point. Such elements are constructed by analytically describing the flat figure $f(x, y) = 0$, and are

considered as a surface with a zero contour $z = f(x, y)$. This description principle is convenient to use in R-functional modeling of the final version of the scene [5, 6]. The aim of the research is to develop tools to simulate the dynamics of changes in the gradient data of the scene surface depending on simulated, dynamically changing targets and obstacles.

To achieve this goal, the following tasks were set:
- research of the gradient descent algorithm based on FV-modeling;
- development of tools to optimize the path with obstacles;
- developing tools for gradient descent control.

2. Research of the gradient descent algorithm based on FV-modeling

To understand the principles of the gradient descent algorithm, as part of the pathfinding problem, it is necessary to understand the basics of the FV-method. Let's consider basic concepts, such as basic and generated M-images.

2.1. Base and generated M-images

The FVM method [6] is based on the construction of basic M-images, which are linearly approximated representations of the function surface. The set of basic M-images itself is a mapping by local geometric characteristics of some scalar continuous function. For example, to obtain a set of two-dimensional basic M-images, the area of function $u = f(x, y)$ is linearly approximated, where each point of the scene space corresponds to the equation of the plane $Ax + By + Cu + D = 0$. Increasing the dimensionality of the normal to the area by one dimension, we obtain the equation $Ax + By + Cu + Dt = 0$. As a result, we have four coefficients (A, B, C, D) for each argument. After normalizing by $N = \sqrt{A^2 + B^2 + C^2 + D^2}$ we obtain the normal components or local geometric characteristics (n_1, n_2, n_3, n_4) for the local equation at each point of the functional domain u:

$$n_1 x + n_2 y + n_3 u + n_4 t = 0.$$

The voxel representation of the FVM-model assumes a monochrome palette representation for a single local characteristic, which can be represented by a correspondence to the value of the color palette gradation P in the range [0, 255]. Figure 1 shows the basic M-images $(C_1^{IV}, C_2^{IV}, C_3^{IV}, C_4^{IV})$ for local characteristics (n_1, n_2, n_3, n_4).

Generation of M-images (C_x^{II}, C_y^{II}) to organize the gradient descent algorithm is carried out by means of two-component renormalization of the corresponding components (n_1, n_2). Figure 2 shows the generated M-images (C_x^{II}, C_y^{II}).

In order to get the generated images it is necessary to perform the following sequence of calculations:

$$n_x^{IV} = \frac{2C_1^{IV} - P}{2}, \quad n_y^{IV} = \frac{2C_2^{IV} - P}{2},$$

$$n_x^{II} = \frac{n_x^{IV}}{\sqrt{(n_x^{IV})^2 + (n_y^{IV})^2}}, \quad n_y^{II} = \frac{n_y^{IV}}{\sqrt{(n_x^{IV})^2 + (n_y^{IV})^2}},$$

$$C_x^{II} = \frac{P(n_x^{II} + 1)}{2}, \quad C_y^{II} = \frac{P(n_y^{II} + 1)}{2}.$$

where P = 256 is the color palette gradation.

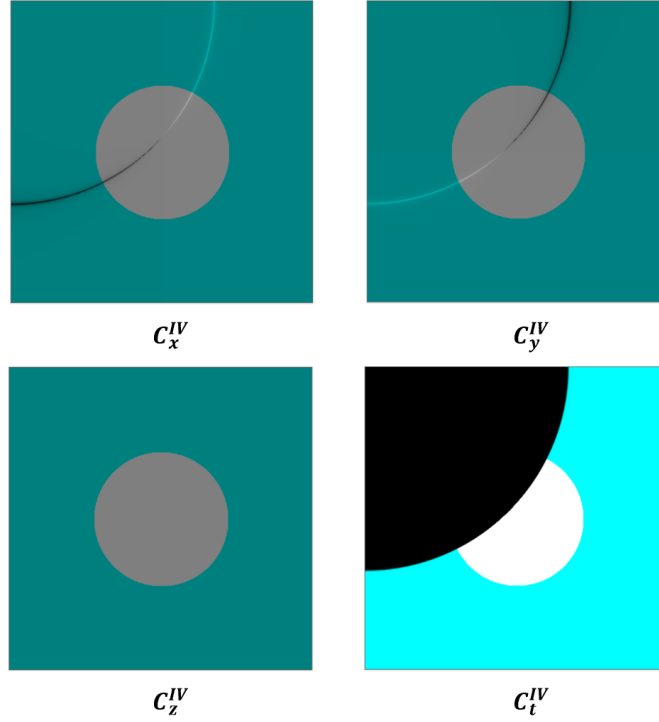$$c_x^{IV} \qquad c_y^{IV}$$

$$c_z^{IV} \qquad c_t^{IV}$$

Figure 1: Graphical representation of basic M-images
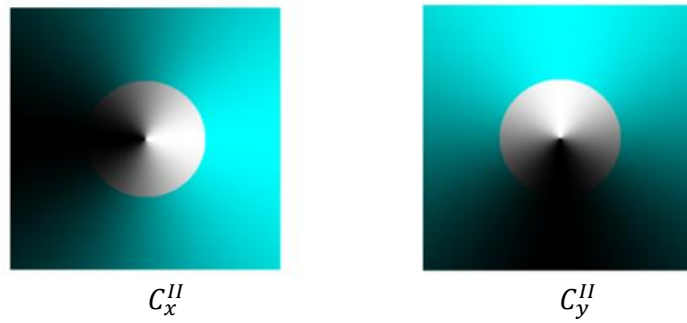

$$C_x^{II} \qquad C_y^{II}$$

Figure 2: Graphical representation of generated M-images

2.2.   A way of local optimization of gradient descent

Based on the information stored in each pixel of the M-image, the shift direction to one of the neighboring pixels is calculated. The gradient descent algorithm is described in detail in [3]. Here it is only important to note that in the control parameters of the algorithm there is a possibility of smoothing the resulting gradient line depending on the set step N [3, 4]. Figure 3 shows an example of such smoothing with different values of N. This parameter can be considered optimizing at the local level of the algorithm, because it affects.
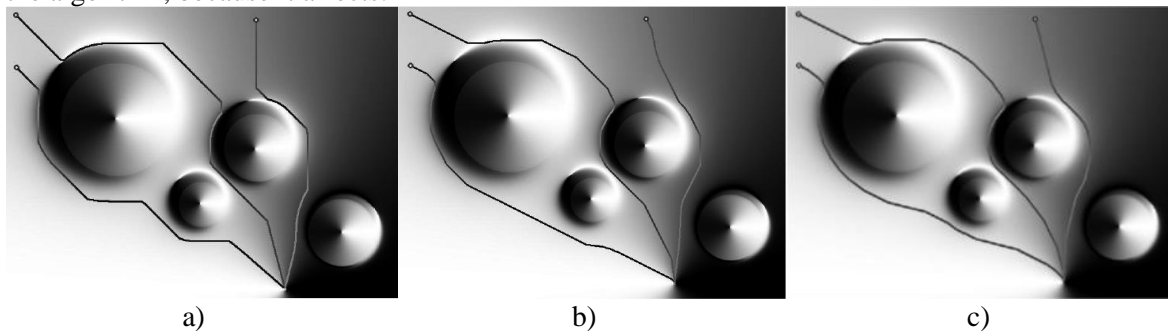

a)   b)   c)

Figure 3: Demonstration of gradient curve smoothing: N=1(a), N=2(b), N=3(c)

Figure 3 shows how the path for each of the presented gradient lines is reduced by smoothing them by the proposed method of increasing the step *N*.

2.3.   Global optimization of gradient descent considering an obstacle

Global optimization (as opposed to local optimization) in the course of solving the path reduction problem considers the properties of the path as a whole. If we consider the gradient line only for the cone surface without the presence of convex obstacles on it, it is obvious that any gradient line will descend to the point of its top along a straight line, i.e. optimally. If we consider the case with an obstacle (Fig.4), then in a straight line motion to the target, the gradient motion meets a change in relief in front of the obstacle, bypassing it. However, after reaching the point where the convexity of the obstacle no longer has any effect on the overall cone surface, the gradient descent again levels off toward the target.
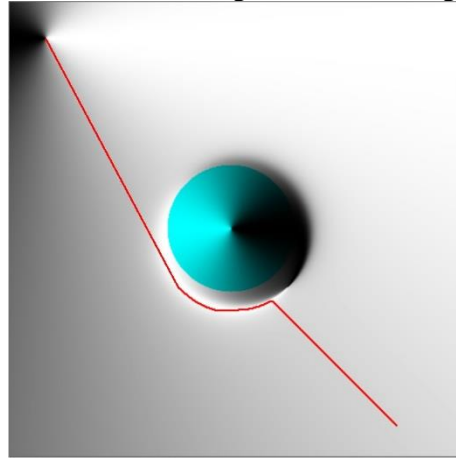

Figure 4: Movement of the gradient descent to the top of the cone with bypassing one obstacle

And hence, the optimization can be simulated using the obtained property. Temporarily swap the target and the starting point (Fig. 5a). Thus, we move the vertex of the cone to the starting point of motion, changing its surface. The path launched from the point of the former target follows the same principle, only in the opposite direction. Now, a section of straight line of movement from the obstacle to the former starting point is formed. Figure 5b demonstrates an overlay of the resulting traces with common intersection points, where we can observe the areas of the descendants for their further application as shown in Figure 6.
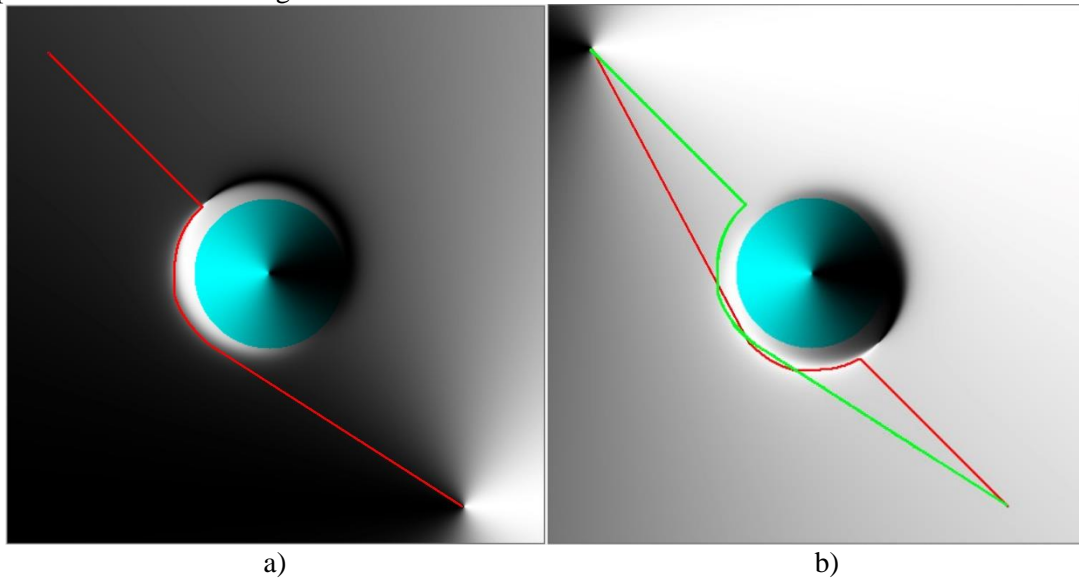

| a) | b) |

Figure 5: Demonstration of the global optimization algorithm: a) moving from the target to the starting point, b) superimposing both traces to determine the shortest
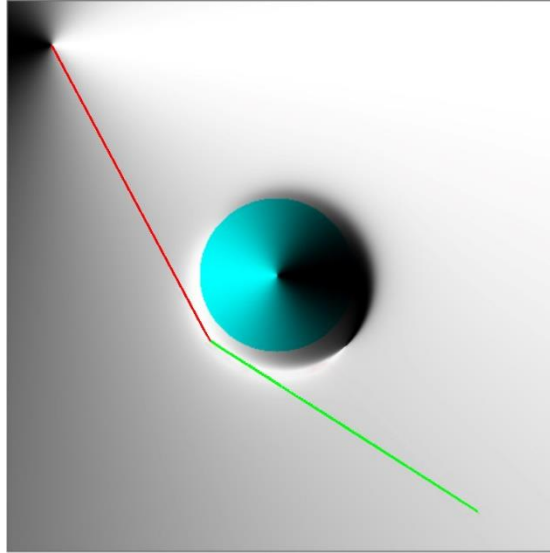
Figure 6: The result of the work of the global optimization algorithm

2.4. Gradient Descent Control Tools

In the set of tools that allow to control the gradient descent at the global level, we can distinguish two main ones: control of the accuracy of the gradient direction and directly control the area of gradient characteristics, i.e. the shape of the solution surface for the situation under consideration in the scene. If the accuracy of directional choice has already been considered earlier and is described in sufficient detail in the work [3], the dynamic change of the scene under the solution of the gradient descent problem is of certain interest, since it is connected with the analytical description of the surface by the function.

Changing the scene involves changing the position of its two main objects: the target and the obstacle. Both objects can change not only their position, but also their shape. The number of objects present (target and obstacle) is determined by the problem statement.

Dynamic modeling of additional objects that appear in the scene is carried out according to the following basic principles: the object must be simple in description and be included in the scene by means of R-functional modeling [7].

For a simple description of the object area, let's take the function $z = r^2 - (x - a)^2 - (y - b)^2$. In this case, to specify its location, it is sufficient to specify only the coordinates of the center (a, b) (e.g., with the mouse). To include an object in the scene, apply the R merge function:

$$x \vee_0 y = x + y + \sqrt{x^2 + y^2},$$

where x is the scene function and y is the function of the inserted object.

An example of constructing a pathfinder in a weakly deterministic environment for functional-voxel method problems is shown in Figure 7. Dynamic objects are visualized by flat circles for distinction and appear sequentially during gradient motion, changing the trajectory of the object to the target.

The movement of the gradient line can be controlled not only by dynamic substitution of object-obstacles, but also by changing the location of the target object. At the same time, the position of the surface of the cone changes, along which the path is laid to its top - the goal. Figure 8 shows the change in path to the new target indicated by the red dot.

One of the problems requiring such an approach is the problem of trajectory falling into a local extremum - a trap. Such a situation can occur even when going around a circular object, not to mention the case discussed in Figure 9. Such object was considered by M.A. Loktev in his works [1,2] and the method of introducing an additional "coefficient of pushing out" into the description of such an obstacle was applied (Fig.10).
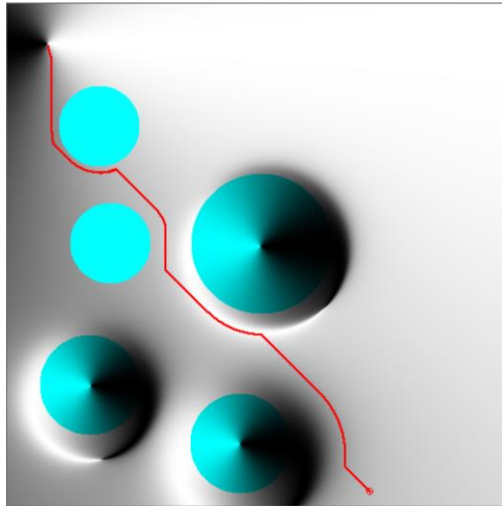
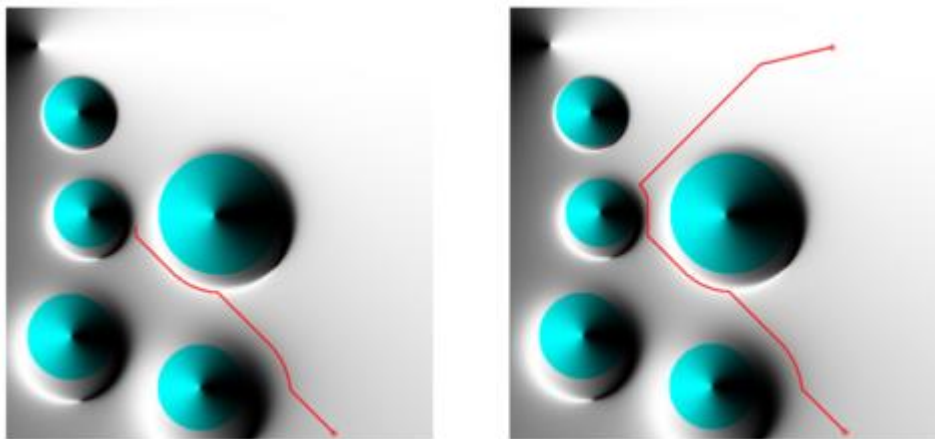**Figure 7:** Pathfinding in a weakly deterministic environment

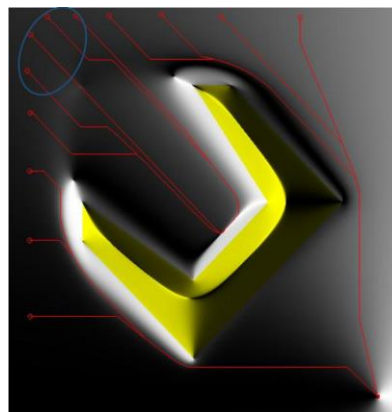**Figure 8:** Dynamic target positioning

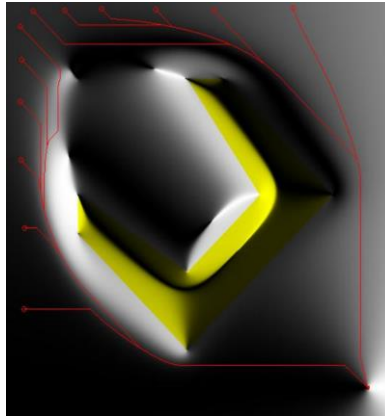**Figure 9:** Caught in a trap

Figure 10: Construction of a gradient descent for a U-shaped "trap" using a coefficient of pushing out

Figure 11 demonstrates a simulation of such a situation using sequential targeting to derive a trajectory from a local extremum to a target.
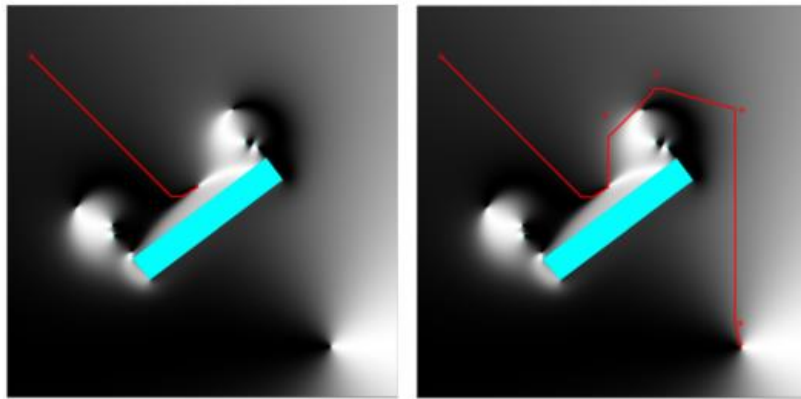

Figure 11: Constructing a gradient descent for a U-shaped "trap" using target reference points

Obviously, with obtaining such a tool, there is a new job of determining the sequence and position of the reference points of such targeting. In the future, it is planned to consider such a solution based on the principles of mathematical programming.

3. Conclusion

The proposed tool for gradient descent modeling in the solution of the obstacle pathway problem allows us to apply the mathematical apparatus of R-functional modeling to such a class of problems. The research has shown the possibility of applying such apparatus for modeling a dynamically changing scene, which allows putting it in line with other approaches that allow modeling pathway in nondeterministic environments.

4. References

[1] M.A. Loktev, A.V. Tolok, Functional principle of obstacle avoidance using functional-voxel modeling method, Bulletin of the MSTU Stankin 36 (2016) 75-80.
[2] M.A. Loktev, Peculiarities of Functional-Voxel Modeling Application in Problems of Pathfinding with Obstacles, Information technologies in design and production 1 (2016) 45-49. In Russian.
[3] E. A. Lotorevich, Principles of spatial visual layout of analytical models displayed in voxel graphical space, Mechanical Engineering Technology 11 (2013) 59-63. In Russian.
[4] A. M. Plaksin, S. A. Pushkarev, Geometric modeling of objects thermal characteristics by the functional-voxel method, Geometry and Graphics 1 (2020) 25-32. In Russian.

[5] K. V. Maksimenko-Sheiko R-functions in mathematical modeling of geometric objects and physical fields. — Kharkov: IPMash of the National Academy of Sciences of Ukraine, 2009.

[6] A.V. Tolok, Functional-Voxel Method in Computer Modeling, Under the editorship of S.N. Vasiliev, Academician of the Russian Academy of Sciences, FIZMATLIT, Moscow, 2016.

[7] V.L. Rvachev, The theory of R-functions and some of its applications, Naukova Dumka, Kiev, 1982.