Mathematical models that coordinate the movement through obstacles of the dynamic systems endowed with artificial sight

OVIDIU ILIE ŞANDRU
Department of Mathematics II
The University "Politehnica" of Bucharest
Splaiul Independenţei 313, 060042 Bucharest
ROMANIA
oisandru@yahoo.com

Abstract: This paper proposes two kinds of geometrical models meant to coordinate the movement through obstacles of the automatons endowed with artificial sight. Besides the novelty of theoretical nature that accompanies them, it is important to underline the fact that these models have been adapted to the possibilities of computer based management.

Key–Words: Mathematical modeling, dynamical system, artificial sight, motion through obstacles, Euclidean distance, Riemann space, geodesics, Dirichlet problem, numerical discretization. ■

1 Introduction

The development of artificial intelligence forms greatly preoccupies many specialists from fields situated at the boundaries between engineering, informatics and mathematics. For them, the problem of creating some performant software applications that establish and coordinate the path through obstacles is a central one.

In order to illustrate, on the one hand, the multitude of applications, and on the other hand, the diversity of the methods that approach the problems discussed, we will mention, for instance, the papers [3, 4, 6, 8] recently published by the World Scientific and Engineering Academy and Society.

Due to the more and more exacting demands, the solutions provided until now are not thoroughly satisfactory and the problem still remains unsolved. That is the reason why any new method to approach it, is more than welcomed.

From the theoretical point of view, the main idea that we would like to highlight in this paper is of pure geometrical nature. It consists in associating the topography of the environment investigated by the system with a Riemannian differentiable manifold, related to which the access ways from one point to another should be chosen among its geodesics.

In this paper we will show that there are multiple ways of turning this idea into reality, ways that highlight two different categories of geometrical models, namely, one made up of geometrical models with singularities and another one made up of geometrical models without singularities.

Related to this fact, it is very important to highlight that each of the methods determined by both types of models has advantages and disadvantages that the other one does not have.

Therefore, it is recommended that software designers choose the work method only after a thorough analysis of both variants has been made.

The importance of the study presented in this paper, is not restricted only to building some very intuitive geometrical models that solve the problems related to the movement control. Actually, its importance is emphasized by the fact that the study also provides means that enable, indeed, the implementation of the models proposed according to the requirements implied by the operating capacity of a computer. More precisely, in this paper we will develop the mathematical instrument for each of the presented models, as well, based on which the chosen variant can be turned into practice.

This last task is even more difficult to solve than the first one because the system must be designed to work under very general conditions: the space where it operates is not a priori known, but on the contrary, it is due to be explored, expressing in this way the concrete situation from the location and from the moment of time when the automaton begins its activity. Thus, the final part of this paper is dedicated to presenting the numerical processing modalities of the information transmitted by the system to the theoretical model and vice versa.

To carry out of this desideratum, it is necessary on the one hand, to search some special techniques to construct some mathematical objects required by the theoretical models used, which are compatible with the increased degree of generality of the studied problem and with the computer operating capacity, and on the other hand, to extend the classical methods of the numerical discretizing of the boundary-value problems so that these can be used to approximate the extremals with fixed limits of the variational problems, too. Apart from the importance of enabling the practical use of the theoretical models developed by us in this paper, this last result has a pure mathematical relevance, too.

2 Geometrical models of the motion through obstacles

In order to simplify the explanation, the model that we intend to construct generally provides solutions for situations that can be expressed in 2 D, as well as images taken from high altitude. The main ideas that lie at the core of these theoretical models can be also extended, so that it can be applied in the case of tridimensional images, too.

The enuntiaton of the problem. An image $\mathfrak{I}$ that contains a certain number of obstacles, is being watched on a computer screen. After marking down the departure point of a robot on this image, as well as the arrival point where the robot is expected to arrive, we are interested in determining a method to establish a path that must be followed in order to avoid collisions with ambient obstacles.

From the study that we have carried out on this issue, two different ways of approaching it have resulted; they are presented in the following two subparagraphs.

2.1 Geometrical models with singularities

The first idea related to solving this problem consists: 1) in the mathematical modeling of the image $\mathfrak{I}$ through a reunion M of multiple connected domains from $\mathbb{R}^2$, whose frontiers are determined by the boundaries of the surface on which the image $\mathfrak{I}$ has been projected, as well as by the contour of the regions occupied by the obstacles that populate the image $\mathfrak{I}$, and 2) in the determination of a Riemannian metric $g_{ij},\ i,\ j = 1,\ 2,$ on M, whose geodesics would represent the possible paths to follow through the obstacles of the image $\mathfrak{I}$[2].

[2]It must not be expected that the path between two locations, established through our model is the shortest among possible alternatives from the point of view of an intuitively-Eucleedean approach. Indeed, let us not forget that this path is geodesic only in rapport with the metric of the model we are constructing. For us, the only practical feature of the path is that it helps us avoid collisions with obstacles within the environment.

In order to simplify the explanation on how to build the model, we will assume for the time being, that the set M is made up of a single component. In this way M limits to a multiple connected domain whose inner boundaries coincide with the contour of the obstacles that the image $\mathfrak{I}$ includes.

Observations 1) *The general case earlier mentioned is imposed by the situations when the image $\mathfrak{I}$ contains obstacles that cross it from one end to the other, dividing it in disjunctive areas without any possibility to avoid.*

2) *The geometrical model, whose construction* will *be described for the special case when the set M is made up of one single connected component, can be easily extended to the case when the set M contains more such components.*■

In order to determine the metric $g_{ij},\ i,\ j = 1,$ 2, we propose the following construction: Considering the set $\mathfrak{F}$, given by the reunion of all interior frontiers of the multiple connected domain M (the reunion of the demarcation lines of the obstacles from the image $\mathfrak{I}$), then, the Monge surface $S:\ r(x,\ y) = (x,\ y,\ z(x,\ y)),(x,\ y) \in M,$ where $z = z(x,\ y)$ may be any differentiable function that has an unbounded behavior on $\mathfrak{F}$, will represent the model of the manifold $(M,\ g_{ij})$ in $\mathbb{R}^3$, the metric $g_{ij},\ i,\ i = 1,\ 2,$ being the Euclidian metric restriction of the arithmetical space $\mathbb{R}^3$ to S, namely $g_{11} = (r_x,\ r_x) = 1 + \left(\frac{\partial z}{\partial x}\right)^2$, $g_{12} = g_{21} = (r_x,\ r_y) = (r_y,\ r_x) = \frac{\partial z}{\partial x}\frac{\partial z}{\partial y}$, $g_{22} = (r_y,\ r_y) = 1 + \left(\frac{\partial z}{\partial y}\right)^2$, where $r_x = \left(1,\ 0,\ \frac{\partial z}{\partial x}\right)$, $r_y = \left(0,\ 1,\ \frac{\partial z}{\partial y}\right)$, and $(\cdot,\ \cdot)$ represent the usual Euclidian inner product in $\mathbb{R}^3$.

In order to use this model in practice it is necessary to describe a concrete method to determine the function $z = z(x,\ y)$ that could be also easily determined by means of the computer which the robot is endowed with, because the topography of the space where this is going to move, is not a priori determined, on the contrary, it is one that needs to be explored each moment. This method consists in choosing z of the form

$$z = f(d(x,\ y)),\ (x,\ y) \in M, \tag{1}$$

where f is a differentiable function from $(0,\ \infty)$ to $\mathbb{R}$ so that $\lim\limits_{x \searrow 0} |f(x)| = \infty$, and $d(x,\ y)$ is the Euclidian distance between the point $(x,\ y) \in M$ and the set $\mathfrak{F}$, for example: $z(x,\ y) = \frac{1}{d(x,\ y)}$, or $z(x,\ y) = \frac{1}{d^2(x,\ y)}$, or $z(x,\ y) = \ln d(x,\ y)$, etc.

Due to the way it has been defined, the function $z = z(x,\ y)$ tends in absolute value to $+\infty$ each time the point $A(x,\ y)$ (the automaton) gets close to an obstacle (to any of the interior frontiers of the domain M).

In comparison with the metric $E = g_{11}$, $F = g_{12} = g_{21}$, $G = g_{22}$, the line element on M will have the following expression $ds^2 = Edx^2 + 2Fdxdy + Gdy^2$, and the length of a curve $\gamma(t) = (x(t),\ y(t))$, $t \in [a,\ b]$, $a < b$, will be calculated using the formula

$$l_\gamma = \int_a^b \sqrt{Ex'^2 + 2Fx'y' + Gy'^2}dt. \tag{2}$$

In this way, in order to determine the geodesics of the manifold $(M,\ g_{ij})$, to the integral (2) we apply the Euler - Lagrange equations

$$\begin{cases} L_x - \frac{d}{dt}L_{x'} = 0, \\ L_y - \frac{d}{dt}L_{y'} = 0, \end{cases} \tag{3}$$

where $L(x,\ y,\ x',\ y') = \sqrt{Ex'^2 + 2Fx'y' + Gy'^2}$.

It is to be noticed that the above relations form an ordinal differential equations system of second order with x and y as unknown functions (of the parameter t) whose general solutions $x = x(t,\ C_1, C_2)$, $y = y(t,\ C_3, C_4)$, depend on 4 integration constants C_1, C_2, C_3, C_4, which can theoretically be determined from the condition that the geodesic $\gamma(t) = (x(t),\ y(t))$, $t \in [a,\ b]$, should pass through the point $A(x_a,\ y_a)$ (the robot's departure point) at the moment $t = a$, and through the point $B(x_b,\ y_b)$ (the robot's arrival or destination point) at the moment $t = b$. Thus, the problem of finding the path between obstacles for a robot that shifts between two predefined locations A and B is solved from the theoretical point of view.

In the general case, when M is not a connected set, this statement should be only referred to the case when the points A and B belong to the same connected component of the set M.

Of course, the problem of determining the distance function $d = d(x,\ y)$, that has been used to build the above mentioned Riemann model, remains open. This problem will be separately addressed in the next paragraph.

From the practical point of view, the implementation of this theoretical model into computer language will be made using the numerical integration of the boundary-value problem

$$\begin{cases} L_x - \frac{d}{dt}L_{x'} = 0, \\ L_y - \frac{d}{dt}L_{y'} = 0, \\ x(a) = x_a,\ x(b) = x_b,\ y(a) = y_a,\ y(b) = y_b, \\ x_a,\ x_b,\ y_a,\ y_b, \text{ given real values.} \end{cases} \tag{4}$$

Since this type of Dirichlet problem has not been studied in the specialized theory, a personal variant to solve it will be presented in paragraph 4 of this paper.

2.2 Geometrical models without singularities

The second idea to solve the problem, formulated at the beginning of this paragraph, relies on the mathematical modeling of the video image $\mathfrak{I}$ within a convex domain D from $\mathbb{R}^2$ and of a Riemannian metric g_{ij}, $i, j = 1, 2$, defined on D whose geodesics represent the robot's possible routes to move through (i.e. routes that avoid the obstacles of the given image).

In the present variant, the domain D is simply defined as a set from $\mathbb{R}^2$, homeomorphic with the set of the points of the surface on which image $\mathfrak{I}$ is projected (for example, a computer screen).

The definition of the Riemannian metric g_{ij}, i, $j = 1, 2$, requires some preparation.

In D the obstacles that populate the image $\mathfrak{I}$ are represented by their demarcation boundaries.

Observations 1) *Within the first model, the variety M was built by means of the set of points belonging to the domain D minus the set of points occupied by the obstacles that populated the image $\mathfrak{I}$.*

2) *Within the present model, no point is removed from the domain D, the representation of the obstacles that populate the image $\mathfrak{I}$ being only realized in a symbolic manner.*■

Let $\mathfrak{F}$ be the reunion of all these boundaries that symbolize the placement of the obstacles of image $\mathfrak{I}$ and let $d : D \to \mathbb{R}_+$ be the function that defines the Euclidean distance from an arbitrary point $(x,\ y) \in D$ to $\mathfrak{F}$. With the help of $d = d(x,\ y)$ we want to build a function $z : D \to \mathbb{R}$ that would take values on $\mathfrak{F}$ that are larger (or smaller) than in the points of D, situated outside the surfaces that symbolize the obstacles of the image $\mathfrak{I}$. For example, we can consider

$z(x,y) = K\ e^{-\frac{d^2(x,y)}{2\sigma^2}}$, or $z(x,y) = \frac{K}{1+\frac{d^2(x,y)}{2\sigma^2}}$,

where K and σ are two real parameters among which K is a very large number in absolute value and σ is a strictly positive number. Also, for any function f $:(0,\ \infty) \to \mathbb{R}$ with the property $\lim\limits_{x \searrow 0} |f(x)| = \infty$, and for any real parameter $\varepsilon > 0$, small enough, we can as well consider

$$z = f(d(x,\ y) + \varepsilon). \tag{5}$$

Indeed, for well chosen values of parameters K or ε the corresponding z functions can take on $\mathfrak{F}$ values as large as wanted.

With the help of the function $z = z(x, y)$ defined previously, we consider the Monge surface S : $r(x, y) = (x, y, z(x, y))$, $(x, y) \in D$. On this surface we consider then the restriction g_{ij}, $i, j = 1, 2$, of the usual Euclidean metrics of the $\mathbb{R}^3$ space, namely

$g_{11} = E = (r_x, r_x) = 1 + \left(\frac{\partial z}{\partial x}\right)^2$, $g_{12} = g_{21} = F = (r_x, r_y) = (r_y, r_x) = \frac{\partial z}{\partial x}\frac{\partial z}{\partial y}$, $g_{22} = G = (r_y, r_y) = 1 + \left(\frac{\partial z}{\partial y}\right)^2$, where $r_x = (1, 0, \frac{\partial z}{\partial x})$, $r_y = (0, 1, \frac{\partial z}{\partial y})$, and $(\cdot, \cdot)$ represent the Euclidean inner product of the $\mathbb{R}^3$space.

Due to the elaborated construction we can consider the metric g_{ij}, $i, j = 1, 2$ of the surface S, that is defined on D and in this way we can regard the pair (D, g_{ij}) as a Rimannian space that it is associated to the problem related to obstacles avoidance.

Further on, we will present the arguments according to which we can consider that, in general, the geodesics of this variety avoid the obstacles of the image $\mathfrak{I}$.

Let us assume for the moment that the image $\mathfrak{I}$ only contains an obstacle $\mathfrak{O}$ which does not cross it from one end to the other and let us denote by $s_{\mathfrak{O}}$ the surface occupied by $\mathfrak{O}$ in the domain D. Now, let A and B be two points in $D \backslash s_{\mathfrak{O}}$ and $\gamma_{[AB]}(t) = (x(t), y(t))$, $t \in [a, b]$, $a < b$, be a simple curve (i.e. a curve that does not intersect itself) from D that joins them. If the curve $\gamma_{[AB]}$ has the representation $\widetilde{\gamma}_{[AB]}(t) = (x(t), y(t), z(x(t), y(t)))$, $t \in [a, b]$ on S, then, within the hypothesis we have earlier assumed, the length of the curve $\widetilde{\gamma}_{[AB]}$ is considerably larger (or smaller) if its projection $\gamma_{[AB]}$ on the plane of the domain D intersects the surface $s_{\mathfrak{O}}$ compared to the case when it does not intersect this surface. Indeed, in $\mathbb{R}^3$ the length of the curve $\widetilde{\gamma}_{[AB]}$ is calculated in connection with the usual Euclidean metric, and, as a result, the intuitive representation of the length notion works.

Thus, in comparison with the metrics g_{ij}, $i, j = 1, 2$, defined on D, the simple curves that avoid the surfaces that designate the obstacles of the image $\mathfrak{I}$ are shorter than those who intersect these surfaces. This result derives from the fact that the length of the curve $\widetilde{\gamma}_{[AB]}$ calculated by using the usual Euclidean metric of the space $\mathbb{R}^3$ coincides with the length of the curve calculated by using the metric g_{ij}, $i, j = 1, 2$, earlier built. Thus, the geodesics of the manifold (D, g_{ij}) avoid, in general, the obstacles of the image $\mathfrak{I}$.

Observations 1) *This remarkable property must be used with caution, as, unlike the situation presented in the previous subsection, we now can not assert that all geodesics of the manifold (D, g_{ij}) avoid the obstacles of the image $\mathfrak{I}$. Indeed, within a geometrical model without singularities, even when there is no access path between two given points, their union through a geodesic, remains possible due to the fact that the manifold (D, g_{ij}) is connected. Also, we can not assert that every path which avoids the obstacles of the image $\mathfrak{I}$ is to be found among the geodesic of the manyfold (D, g_{ij}), as paths of various lengths may exist, and can provide access ways between the two given points. Thus, in very accurate terms, the property presented earlier only allows us to restrict the search to the geodesics of the manifold (D, g_{ij}) when looking for paths which avoid the obstacles of the image $\mathfrak{I}$.*

2) *Obviously, all these precautions that we must take into consideration when we choose the model without singularities as a work method, diminish the importance of this method, yet we must not forget that this work variant has its own advantages that still make it of interest.*

3) *A way to decide whether a geodesic of the manifold (D, g_{ij})* is *or not the searched path, would be to set some acceptance length limits.*■

After having established the applicability limits of this remarkable property, all that we have to do is to indicate how the geodesics of the space (D, g_{ij}) can be found.

The technique to determine the geodesics of the manifold (D, g_{ij}) is similar to the one presented in the case of the manifolds (M, g_{ij}) where M is a domain or a reunion of multiple connected domains. Indeed, if we denote $g_{11} = E$, $g_{12} = g_{21} = F$, $g_{22} = G$, then the square of the arch element on D has the following expression $ds^2 = Edx^2 + 2Fdxdy + Gdy^2$ and the length of an arch of the curve $\gamma(t) = (x(t), y(t))$, $t \in [a, b]$ will be calculated using the formula $l_\gamma = \int_a^b \sqrt{Ex'^2 + 2Fx'y' + Gy'^2}dt$.

Therefore, in order to find the geodesics of (D, g_{ij}) we should only determine the extremals of the functional $\gamma \to l_\gamma$, $\gamma : [a, b] \to D$, with fixed limits $\gamma(a) = (x_a, y_a)$, $\gamma(b) = (x_b, y_b)$, namely use the well-known Euler-Lagrange equations system

$$\begin{cases} L_x - \frac{d}{dt}L_{x'} = 0, \\ L_y - \frac{d}{dt}L_{y'} = 0, \end{cases} \tag{6}$$

where $L(x, y, x', y') = \sqrt{Ex'^2 + 2Fx'y' + Gy'^2}$.

Observations 1) *The fact that the systems (3) and (6) are different is due to the metrics g_{ij}, $i, j = 1, 2$, used in each of the cases, and ultimately, due to the functions $z = z(x, y)$ used to build the models (M, g_{ij}), and respectively, (D, g_{ij}).*

2) *If we denote by* $g_{\varepsilon_{ij}}$, $i, \ \ j = 1, \ \ 2$, *the metrics defined on* D *with the help of functions* z, *having the form (5), then the models with singularities* $(M, \ g_{ij})$ *can be regarded as the (limiting) boundary cases of some models without singularities of the form* $\left(D, \ g_{\varepsilon_{ij}}\right)$. *This remarkable property that enhances the connection between the two mathematical models will be intensely used when we will describe the way these models can be used in practice.*

3 The determination of the function d

Because of the context we assume, the problem of determining the distance function $d = d\,(x, \ y)$ must supplementary satisfy the following two conditions:

1) the process of determining the function d must be applicable to an image $\mathfrak{I}$ which is beforehand unknown;

2) this process must be translatable into computer language.

Next, two ways of computer based approximations of this type of function will be presented; the first one belongs to M. Demi (see[1]) and the second one is a variant of the previous, that has been presented by us in [7].

Let us suppose that a certain given image is projected on a computer screen. The pixel gray levels by coordinates $(x, \ y)$ will be denoted by $I = I\,(x, \ y)$. Thus a function with real values defined on the screen surface is obtained. In the above mentioned paper, M. Demi analyzes the video images by using a class of Gaussian filters $H_\sigma\,(x, \ y) = \left(h_{\sigma_x} * I\right)^2 (x, \ y) + \left(h_{\sigma_y} * I\right)^2 (x, \ y)$, $\sigma > 0$, where $h_{\sigma_x}\,(x, \ y) = -\frac{x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$, $h_{\sigma_y}\,(x, \ y) = -\frac{y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$, are the partial derivatives of the function $h_\sigma\,(x, \ y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$, and "$*$" is the convolution product. According to M. Demi's theory [1], in the particular case when the gray level of the analyzed image would have the distribution $I\,(x, \ y) = \begin{cases} 0, y > mx \\ \alpha, y \leq mx \end{cases}$, $\alpha, \ \ m \in \mathbb{R}$, $\alpha \neq 0$, the Gaussian filter H_σ, would admit the following expression: $H_\sigma\,(x, \ y) = 2\pi\alpha^2\sigma^2 e^{-\frac{d^2(x, \ y)}{\sigma^2}}$, where $d = d\,(x, \ y)$ represents the Euclidean distance from the point $(x, \ y)$ to the line $y = mx$. Due to this result, in order to actually determine the function $d^2 = d^2(x, \ y)$, following formulas can be used $d^2\,(x, \ y) = -\sigma^2 \ln H_\sigma\,(x, \ y) + \sigma^2 \ln 2\pi\alpha^2\sigma^2$, or $d^2\,(x, \ y) = \sigma^4 \frac{H^2_{\sigma x}(x, \ y) + H^2_{\sigma y}(x, \ y)}{4H^2_\sigma(x, \ y)}$, (see [1]).

Observation *Obviously, in the case of an arbitrary image* $\mathfrak{I}$, *none of the previous formulas will no longer provide the square of an authentic Euclidean distance, except for a more or less exact estimation of it. There are two reasons why we suggest however the utilization of the earlier mentioned proceeding. The first reason is due to the fact that the results this proceeding provides are in most of the cases acceptable and the second reason is due to the computation method used to approximate the function* d. *Thus, except for some simple arithmetical operations, the computer is used only to calculate certain convolution products for whose evaluation one can use, for instance, the trapezoidal method or the Simpson's method (for details see, for example, [2]).*■

This paragraph ends by describing a second method to determine the function $d = d\,(x, \ y)$.

In the paper [7] it has been shown that by replacing the Gaussian function $h_\sigma\,(x, \ y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$, $\sigma > 0$, by a function of the form $h_\sigma\,(x, \ y) = \frac{1}{1+\frac{x^2+y^2}{2\sigma^2}}$, $\sigma > 0$, the corresponding filter $H_\sigma\,(x, \ y) = \left(h_{\sigma_x} * I\right)^2 (x, \ y) + \left(h_{\sigma_y} * I\right)^2 (x, \ y)$, $\sigma > 0$, will be expressed, in the particular case $I\,(x, \ y) = \begin{cases} 0, y > mx \\ \alpha, y \leq mx \end{cases}$, $\alpha, \ m \in \mathbb{R}$, $\alpha \neq 0$, by means of the same distance function d that has previously been mentioned. More precisely there is $H_\sigma\,(x, \ y) = \frac{2\pi^2\alpha^2\sigma^2}{1+\frac{d^2(x, \ y)}{2\sigma^2}}$, $\sigma > 0$, where $d = d\,(x, \ y)$ is the Euclidean distance from the point $(x, \ y)$ to the line $y = mx$.

From the previous formula one can deduce $d^2\,(x, \ y) = 2\sigma^2 \frac{2\pi^2\alpha^2\sigma^2 - H_\sigma(x, \ y)}{H_\sigma(x, \ y)}$.

The relation obtained satisfies the requirements formulated at the beginning of the paragraph as well, namely it provides a compatible alternative to a computer operating capacity in order to obtain a satisfactory approximation of the distance function, required by the theoretical models presented in the previous paragraph.

4 The numerical discretization of the Dirichlet problem (4)

Since the boundary-value problem (4) which the practical utilization of the theoretical models to determine the path between obstacles depends on, does not fit into the typology of the problems solved by means of the numerical analysis, we shall have to find new methods in order to solve it.

Thus, in order to numerically solve the problem (4), we shall propose a special variational approximation method, in this paragraph. The method used in the field literature for the discretization of the elliptical boundaries problems, see for example [5], has served as inspiration source for it. Let $(x(t),\ y(t))$, $t \in [a,\ b]$, $a < b$, be a solution of the Dirichlet problem (4). After performing the substitutions $x(t) = \widetilde{x}(t) + \frac{t-a}{b-a}x_b + \frac{b-t}{b-a}x_a$, $y(t) = \widetilde{y}(t) + \frac{t-a}{b-a}y_b + \frac{b-t}{b-a}y_a$, $t \in [a,\ b]$, the initial Dirichlet problem (4) transforms into the equivalent Dirichlet problem:

$$\begin{cases} \widetilde{L}_{\widetilde{x}} - \frac{d}{dt}\widetilde{L}_{\widetilde{x}'} = 0, \\ \widetilde{L}_{\widetilde{y}} - \frac{d}{dt}\widetilde{L}_{\widetilde{y}'} = 0, \\ \widetilde{x}(a) = \widetilde{y}(a) = \widetilde{x}(b) = \widetilde{y}(b) = 0, \end{cases} , \quad (7)$$

where $\widetilde{L}(t, \widetilde{x}, \widetilde{y}, \widetilde{x}', \widetilde{y}') = L(\widetilde{x}(t) + \frac{t-a}{b-a}x_b + \frac{b-t}{b-a}x_a, \widetilde{y}(t) + \frac{t-a}{b-a}y_b + \frac{b-t}{b-a}y_a, \widetilde{x}'(t) + \frac{x_b - x_a}{b-a}, \widetilde{y}'(t) + \frac{y_b - y_a}{b-a})$. Next, the preparations required by a variational formulation of the problem (7) are being made. Let V be the space of functions continuous almost everywhere on the interval $[a,\ b]$, $(a < b)$ that vanish in a and in b. Any function from V is derivable, at least with respect to the distribution theory. Therefore, the determination of two functions $\widetilde{x}$, and $\widetilde{y}$ from V so that

$$\begin{cases} \int\limits_a^b \{\widetilde{L}_{\widetilde{x}}(t,\ \widetilde{x}(t),\ \widetilde{y}(t),\ \widetilde{x}'(t),\ \widetilde{y}'(t))\, v(t) + \\ +\widetilde{L}_{\widetilde{x}'}(t,\ \widetilde{x}(t),\ \widetilde{y}(t),\ \widetilde{x}'(t),\ \widetilde{y}'(t))\, v'(t)\}dt = 0, \\ \int\limits_a^b \{\widetilde{L}_{\widetilde{y}}(t,\ \widetilde{x}(t),\ \widetilde{y}(t),\ \widetilde{x}'(t),\ \widetilde{y}'(t))\, v(t) + \\ +\widetilde{L}_{\widetilde{y}'}(t,\ \widetilde{x}(t),\ \widetilde{y}(t),\ \widetilde{x}'(t),\ \widetilde{y}'(t))\, v'(t)\}dt = 0, \end{cases} \quad (8)$$

for any $v \in V$, makes sense. This problem represents the variational formulation of the problem (7). Indeed, because $\int\limits_a^b \{\widetilde{L}_{\widetilde{x}}v + \widetilde{L}_{\widetilde{x}'}v'\}dt = \int\limits_a^b \{\widetilde{L}_{\widetilde{x}} - \frac{d}{dt}\widetilde{L}_{\widetilde{x}'}\}vdt$, and $\int\limits_a^b \{\widetilde{L}_{\widetilde{y}}v + \widetilde{L}_{\widetilde{y}'}v'\}dt = \int\limits_a^b (\widetilde{L}_{\widetilde{y}} - \frac{d}{dt}\widetilde{L}_{\widetilde{y}'})vdt$, then the problem (8) is equivalent to the problem of determining two functions $\widetilde{x}$, and $\widetilde{y}$ from V so that

$$\begin{cases} \int\limits_a^b \{\widetilde{L}_{\widetilde{x}}(t,\ \widetilde{x}(t),\ \widetilde{y}(t),\ \widetilde{x}'(t),\ \widetilde{y}'(t)) - \frac{d}{dt}\widetilde{L}_{\widetilde{x}'}(t, \\ \widetilde{x}(t),\ \widetilde{y}(t),\ \widetilde{x}'(t),\ \widetilde{y}'(t))\}v(t)\,dt = 0 \\ \int\limits_a^b \{\widetilde{L}_{\widetilde{y}}(t,\ \widetilde{x}(t),\ \widetilde{y}(t),\ \widetilde{x}'(t),\ \widetilde{y}'(t)) - \frac{d}{dt}\widetilde{L}_{\widetilde{y}'}(t, \\ \widetilde{x}(t),\ \widetilde{y}(t),\ \widetilde{x}'(t),\ \widetilde{y}'(t))\}v(t)\,dt = 0 \end{cases} \quad (9)$$

for all $v \in V$. Thus if the applications $t \to \widetilde{L}_{\widetilde{x}} - \frac{d}{dt}\widetilde{L}_{\widetilde{x}'}$, and $t \to \widetilde{L}_{\widetilde{y}} - \frac{d}{dt}\widetilde{L}_{\widetilde{y}'}$, are continuous, the problem (9) is equivalent to the problem (7). For those cases when this problem admits a unique solution, one of its approximations can be made in the following way: A natural number n is chosen and we calculate $\delta = \frac{b-a}{n+1}$. By denoting $t_i = a + i\delta$, $i = 0,\ 1, .., \ n+1$, we obtain a division of the interval $[a,\ b]$ of spacing δ. We consider the set $V_\delta = \left\{v \in V \mid v\big|_{[t_i,\ t_{i+1}]} \in \mathbb{R}_1[t], 0 \le i \le n\right\}$, where $\mathbb{R}_1[t]$ is the space of the polynomial functions of degree 1 or less, and $v\big|_{[t_i,\ t_{i+1}]}$, is the restriction of the function v on the interval $[t_i,\ t_{i+1}]$. Together with the usual functions addition operation and the usual functions multiplication operation with scalars, V_δ is a vectorial subspace of the space V. The functions

$$\varphi_i(t) = \begin{cases} 1 - \frac{|t-t_i|}{\delta}, & t \in [t_{i-1},\ t_{i+1}] \\ 0, & t \notin [t_{i-1},\ t_{i+1}] \end{cases},$$

$i = 1,\ 2, .., \ n$, constitute a base of the space V_δ. If we want to approximate the solution of the problem (8) by using functions from V_δ, we replace in the equation (8) the unknown functions $\widetilde{x} = \widetilde{x}(t)$ and $\widetilde{y} = \widetilde{y}(t)$ with $\widetilde{x}_\delta(t) = \sum\limits_{i=1}^{n} x_i\varphi_i(t)$, respectively with $\widetilde{y}_\delta(t) = \sum\limits_{i=1}^{n} y_i\varphi_i(t)$, and we gradually replace the function $v = v(t)$ with each of the functions $\varphi_j = \varphi_j(t)$, $j = 1,\ 2,\ ..,\ n$, the determination of the coefficients x_i, y_i, $i = 1,\ 2,\ ..,\ n$, of the definition expressions of the functions $\widetilde{x}_\delta$ and $\widetilde{y}_\delta$ being deduced from the following conditions

$$\begin{cases} \sum\limits_{i=1}^{n+1} \int\limits_{t_{i-1}}^{t_i} \{\widetilde{L}_{\widetilde{x}}(t,\ \widetilde{x}_\delta(t),\ \widetilde{y}_\delta(t),\ \widetilde{x}'_\delta(t),\ \widetilde{y}'_\delta(t)) \cdot \\ \cdot\varphi_j(t) + \widetilde{L}_{\widetilde{x}'}(t,\ \widetilde{x}_\delta(t),\ \widetilde{y}_\delta(t),\ \widetilde{x}'_\delta(t),\ \widetilde{y}'_\delta(t)) \cdot \\ \cdot\varphi'_j(t)\}dt = 0, \\ \sum\limits_{i=1}^{n+1} \int\limits_{t_{i-1}}^{t_i} \{\widetilde{L}_{\widetilde{y}}(t,\ \widetilde{x}_\delta(t),\ \widetilde{y}_\delta(t),\ \widetilde{x}'_\delta(t),\ \widetilde{y}'_\delta(t)) \cdot \\ \cdot\varphi_j(t) + \widetilde{L}_{\widetilde{y}'}(t,\ \widetilde{x}_\delta(t),\ \widetilde{y}_\delta(t),\ \widetilde{x}'_\delta(t),\ \widetilde{y}'_\delta(t)) \cdot \\ \cdot\varphi'_j(t)\}dt = 0, \\ j = 1,\ 2,\ ..,\ n. \end{cases} \quad (10)$$

Indeed, these conditions form a system of $2n$ equations with $2n$ unknown that under certain conditions, yet not the restrictive one, admit solutions. The numerical approximation of the system solutions (10), if it is a compatible one, can performed by using the well-known Newton's method, for instance (for more details see [2]).

5 Comments on the possibilities regarding the practical use of the models proposed in this paper

The practical use of the models proposed in this paper, implies afterwards the numerical solving of some algebraic systems of the form (10). Before raising the issue of how to approximate the solutions of the systems of the form, (10) the way how the construction of such systems is achieved by computer needs firstly, an analysis, since in the hypotheses we operate with, their description depends on factors we do not know before having analyzed the video information representing the field situation, situation that, as it is normal, may differ from one case to another. In this respect, let us observe that the reconstitution of the systems of the form (10) depends on knowing components $\widetilde{L}_{\widetilde{x}}$, $\widetilde{L}_{\widetilde{x}'}$, $\widetilde{L}_{\widetilde{y}}$, $\widetilde{L}_{\widetilde{y}'}$ which these are built of. By explicitating the components $\widetilde{L}_{\widetilde{x}}$, $\widetilde{L}_{\widetilde{x}'}$, $\widetilde{L}_{\widetilde{y}}$, $\widetilde{L}_{\widetilde{y}'}$, we observe that these in their turn depend, on the coefficients of the first fundamental form E, F, G, of the surface S that provides the geometrical model used, on the functions $\widetilde{x}_\delta$, $\widetilde{y}_\delta$, as well as on their derivatives E_x, F_x, G_x, E_y, F_y, G_y, $\widetilde{x}'_\delta$, $\widetilde{y}'_\delta$. Indeed, $\widetilde{L}_{\widetilde{x}}$ is, for instance, expressed by means of an expression of the form $\frac{1}{2}\{\widetilde{E}_{\widetilde{x}}\left(\widetilde{x}'_\delta(t)+\frac{x_b-x_a}{b-a}\right)^2 + 2\widetilde{F}_{\widetilde{x}}\left(\widetilde{x}'_\delta(t)+\frac{x_b-x_a}{b-a}\right)\left(\widetilde{y}'_\delta(t)+\frac{y_b-y_a}{b-a}\right) + \widetilde{G}_{\widetilde{x}}\left(\widetilde{y}'_\delta(t)+\frac{y_b-y_a}{b-a}\right)^2\} \cdot \{\widetilde{E}\left(\widetilde{x}'_\delta(t)+\frac{x_b-x_a}{b-a}\right)^2 + 2\widetilde{F}\left(\widetilde{x}'_\delta(t)+\frac{x_b-x_a}{b-a}\right)\left(\widetilde{y}'_\delta(t)+\frac{y_b-y_a}{b-a}\right) + \widetilde{G}\left(\widetilde{y}'_\delta(t)+\frac{y_b-y_a}{b-a}\right)^2\}^{-\frac{1}{2}}$, where the coefficients $\widetilde{E}$, $\widetilde{F}$, $\widetilde{G}$, $\widetilde{E}_{\widetilde{x}}$, $\widetilde{F}_{\widetilde{x}}$, $\widetilde{G}_{\widetilde{x}}$, are obtained from the coefficients E, F, G, E_x, F_x, G_x, by substituting $x(t) = \widetilde{x}_\delta(t) + \frac{t-a}{b-a}x_b + \frac{b-t}{b-a}x_a$, $y(t) = \widetilde{y}_\delta(t) + \frac{t-a}{b-a}y_b + \frac{b-t}{b-a}y_a$, $x'(t) = \widetilde{x}'_\delta(t) + \frac{x_b-x_a}{b-a}$, $y'(t) = \widetilde{y}'_\delta(t) + \frac{y_b-y_a}{b-a}$, $t \in [a,\, b]$, etc.

At first glance, the determination of coefficients E, F, G, E_x, F_x, G_x, E_y, F_y, G_y, seems very difficult to achieve, taking into account the limited operating possibilities of a computer. But in reality, this situation is different. Indeed, the coefficients E, F, G, E_x, F_x, G_x, E_y, F_y, G_y, can be calculated using the partial derivatives of the the function $z = z(x,\, y)$. Since in our theory, $z = z(x,\, y)$ is expressed by means of H_σ, and eventually of its derivatives; let us observe that any of the derivatives of the function $z = z(x,\, y)$ is an expression made up of the function H_σ and of its derivatives, up to a certain order.

As a result of this situation the degree of difficulty of the evaluation of the coefficients E, F, G, E_x, F_x, G_x, E_y, F_y, G_y, depends on how difficult or how easy it is to calculate the function H_σ and its derivatives of different orders. Because $H_\sigma = (h_{\sigma_x} * I)^2 + \left(h_{\sigma_y} * I\right)^2$, one can immediately notice that both the function H_σ and its derivatives are obtained from the convolution product between different derivatives of the function h_σ and the function I. Since in our theory the function h_σ is known a priori, its derivatives can be calculated beforehand. Thus, the actual determination of the coefficients E, F, G, E_x, F_x, G_x, E_y, F_y, G_y, reduces to the calculation of the convolution product between some functions known a priori and the function I that the computer builds when the analyzed image is projected on the screen. In addition, this thing enables us not only to numerically approximate the value of the convolution products that we refer in different fixed points of the definition domain to, but also to determine some analytical expressions that will approximate these convolution products. Therefore, the approximation of expressions of the form $\widetilde{E}\left(\widetilde{x}'_\delta(t)+\frac{x_b-x_a}{b-a}\right)^2 + 2\widetilde{F}\left(\widetilde{x}'_\delta(t)+\frac{x_b-x_a}{b-a}\right)\left(\widetilde{y}'_\delta(t)+\frac{y_b-y_a}{b-a}\right) + \widetilde{G}\left(\widetilde{y}'_\delta(t)+\frac{y_b-y_a}{b-a}\right)^2$, $\widetilde{E}_{\widetilde{x}}\left(\widetilde{x}'_\delta(t)+\frac{x_b-x_a}{b-a}\right)^2 + 2\widetilde{F}_{\widetilde{x}}\left(\widetilde{x}'_\delta(t)+\frac{x_b-x_a}{b-a}\right)\left(\widetilde{y}'_\delta(t)+\frac{y_b-y_a}{b-a}\right) + \widetilde{G}_{\widetilde{x}}\left(\widetilde{y}'_\delta(t)+\frac{y_b-y_a}{b-a}\right)^2$, $\widetilde{E}_{\widetilde{y}}\left(\widetilde{x}'_\delta(t)+\frac{x_b-x_a}{b-a}\right)^2 + 2\widetilde{F}_{\widetilde{y}}\left(\widetilde{x}'_\delta(t)+\frac{x_b-x_a}{b-a}\right)\left(\widetilde{y}'_\delta(t)+\frac{y_b-y_a}{b-a}\right) + \widetilde{G}_{\widetilde{y}}\left(\widetilde{y}'_\delta(t)+\frac{y_b-y_a}{b-a}\right)^2$, etc., does not raise any problems because by knowing the analytical expressions of the functions $\widetilde{x}_\delta$, $\widetilde{y}_\delta$, their derivatives $\widetilde{x}'_\delta$, $\widetilde{y}'_\delta$ can be calculated beforehand. Consequently, the determination of the systems (10) can be organized in a way that is compatible with the operating capacity of a computer.

To conclude, the only thing left is to prove that it is possible to numerically approximate the solutions of the system (10) by using Newton's method even when the function $I = I(x,\, y)$ is unknown at the moment when the algorithm is created.

If $\mathbf{f}_1(\mathbf{x},\, \mathbf{y}) = (f_{1j}(\mathbf{x},\, \mathbf{y}))$, $\mathbf{f}_2(\mathbf{x},\, \mathbf{y}) = (f_{2j}(\mathbf{x},\, \mathbf{y}))$, $j = 1,\, 2,\, ..,n$, where $\mathbf{x} = (x_1, .., x_n)$, $\mathbf{y} = (y_1, .., y_n)$, $f_{1j}(\mathbf{x},\, \mathbf{y}) = \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} \{\widetilde{L}_{\widetilde{x}}(t,\ \widetilde{x}_\delta(t),\ \widetilde{y}_\delta(t),\ \widetilde{x}'_\delta(t),\ \widetilde{y}'_\delta(t))\varphi_j(t) + \widetilde{L}_{\widetilde{x}'}(t,\ \widetilde{x}_\delta(t),\ \widetilde{y}_\delta(t),\ \widetilde{x}'_\delta(t),\ \widetilde{y}'_\delta(t))\varphi'_j(t)\}dt$,

$$f_{2j}(\mathbf{x},\, \mathbf{y}) = \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} \{\widetilde{L}_{\widetilde{y}}(t, \widetilde{x}_\delta(t), \widetilde{y}_\delta(t), \widetilde{x}'_\delta(t),$$

$\widetilde{y}'_{\delta}(t))\varphi_j(t) + \widetilde{L}_{\widetilde{y}'}(t, \widetilde{x}_{\delta}(t), \widetilde{y}_{\delta}(t), \widetilde{x}'_{\delta}(t), \widetilde{y}'_{\delta}(t))$ $\varphi'_j(t)\}dt$, then by knowing a certain approximation $(\mathbf{x}^{(p)},\ \mathbf{y}^{(p)})$, $p \geq 0$, of an system solution

$$\begin{cases} \mathbf{f}_1(\mathbf{x},\ \mathbf{y}) = 0 \\ \mathbf{f}_2(\mathbf{x},\ \mathbf{y}) = 0 \end{cases}, \tag{11}$$

the next approximation of this solution may be determined using the following formula $\begin{pmatrix} \mathbf{x}^{(p+1)} \\ \mathbf{y}^{(p+1)} \end{pmatrix} = \begin{pmatrix} \mathbf{x}^{(p)} \\ \mathbf{y}^{(p)} \end{pmatrix} - W^{-1}(\mathbf{x}^{(p)},\ \mathbf{y}^{(p)})$ $\begin{pmatrix} \mathbf{f}_1(\mathbf{x}^{(p)},\ \mathbf{y}^{(p)}) \\ \mathbf{f}_2(\mathbf{x}^{(p)},\ \mathbf{y}^{(p)}) \end{pmatrix}$, known as the formula of Newton. In this formula $W^{-1}(\mathbf{x}^{(p)},\ \mathbf{y}^{(p)})$ represents the inverse of the Jacobi matrix $\frac{\partial(\mathbf{f}_1,\ \mathbf{f}_2)}{\partial(\mathbf{x},\ \mathbf{y})}$ calculated in the point $(\mathbf{x}^{(p)},\ \mathbf{y}^{(p)})$. More details on the conditions in which the sequence $(\mathbf{x}^{(p)},\ \mathbf{y}^{(p)})$, $p \geq 0$ of successive approximations converges to the targeted solution are available, for example, in [2].

The difficulty that can be pointed out when using this method and that we try to solve next, is related to the fact that the applicability of the method described depends on whether an approximation of the targeted solution is previously known or not.

In the case of manifolds of the form $(D,\ g_{ij})$, $\mathbf{x}^{(0)} = (0,\ 0, .., \ 0)$, $\mathbf{x}^{(0)} = (0,\ 0, .., \ 0)$ can always be considered as an initial approximation of one of the solutions of the system (11). Indeed, finding an approximation of one of the solutions of the system (11) is equivalent with finding an approximation of one of the solutions of the system (8) through functions from V_{δ}. We remind that $\delta = \frac{b-a}{n+1}$ is a parameter that is chosen according to the accuracy degree we want the functions $\widetilde{x}_{\delta}$ and $\widetilde{y}_{\delta}$ from V_{δ} to approximate the (exact) solution $\widetilde{x}$, $\widetilde{y}$, of the system (8) with. Since the domains of the varieties $(D,\ g_{ij})$ are convex, the points A and B (the robot's departure and arrival location) can be joined (by making abstraction of the obstacles that populate the image $\mathfrak{I}$) through the line segment of the parametric equations $x(t) = \frac{t-a}{b-a}x_b + \frac{b-t}{b-a}x_a$, $y(t) = \frac{t-a}{b-a}y_b + \frac{b-t}{b-a}y_a$, $t \in [a,\ b]$. With respect to the approximation of the path between A and B, it is obvious that the functions $\widetilde{x}$ and $\widetilde{y}$ defined by the following relations $x(t) = \widetilde{x}(t) + \frac{t-a}{b-a}x_b + \frac{b-t}{b-a}x_a$, $y(t) = \widetilde{y}(t) + \frac{t-a}{b-a}y_b + \frac{b-t}{b-a}y_a$, $t \in [a,\ b]$, are identically null. Thus, for the approximation of the functions $\widetilde{x}(t) \equiv 0$, $\widetilde{y}(t) \equiv 0$, through functions from V_{δ}, we can consider the functions $\widetilde{x}_{\delta}(t) \equiv 0$, and respectively $\widetilde{y}_{\delta}(t) \equiv 0$.

Since the curve $\gamma(t) = \left(\frac{t-a}{b-a}x_b + \frac{b-t}{b-a}x_a,\ \frac{t-a}{b-a}y_b + \frac{b-t}{b-a}y_a\right)$, $t \in [a,\ b]$, can precisely intersect a part of the surfaces from D that symbolize the obstacles from the image $\mathfrak{I}$, two important remarks need consequently to be made:

1) If, between the points A and B there is no access path, the presented method leads to false results[1].

2) The method presented can not be generally applied within the first model.

Due to the restriction enunciated at point 2), in the case of manifolds having the form $(M,\ g_{ij})$, finding an initial approximation of one of the solutions of the system (11) is not that simple anymore, but it implies going over some previous steps. As we will see, this thing can be done in two different ways: one has a theoretical nature and is based on the connection between the two mathematical models and the other one has an empirical nature and is based on the use of a special program of analyzing image $\mathfrak{I}$. Due to the intrinsic character of the empirical method, its presentation will be done in a separate paragraph. From here on we will focus on the first of the two variants mentioned above.

Let us fix a real number $\varepsilon > 0$. Replacing d with $d + \varepsilon$ in the relation (1), we get an extension of the function $z = z(x,\ y)$ from M to D. Due to this extension the model with singularities $(M,\ g_{ij})$ is replaced with the model without singularities $(D,\ g_{\varepsilon_{ij}})$ where the metric $g_{\varepsilon_{ij}}$ represents the restriction of the Euclidean metrics of the space $\mathbb{R}^3$ on the surface $z = f(d(x,\ y) + \varepsilon)$, $(x,\ y) \in D$.

Earlier we have shown that the numerical approximation of the geodesics of the models without singularities is possible and we have presented an actual way of carrying it out. So, let γ_{ε} be the geodesic (or at least a very precise approximation of it) of the model $(D,\ g_{\varepsilon_{ij}})$ that unites the points A and B from M between which we want to establish a path of access. If the image of γ_{ε} does not intersect the set $\mathfrak{F}$ then the curve γ_{ε} itself can constitute a variant of the path from A to B, or it can be considered as the first approximation of the geodesic of the space $(M,\ g_{ij})$ that unifies the points A and B.

It is important to notice that whenever there are paths between the points A and B that avoid the obstacles of the image $\mathfrak{I}$, then, generally, after a finite number of repetitions of the algorithm presented, one of these paths will be actually obtained. More precisely a curve of the form $\gamma_{\frac{\varepsilon}{2^n}} : [a,\ b] \to M$ with the property $\gamma_{\frac{\varepsilon}{2^n}}(a) = A$, $\gamma_{\frac{\varepsilon}{2^n}}(b) = B$ will be obtained, where n represents the number of repetitions of the algorithm. In the end this curve can be considered a solution of the problem of searching paths

[1] As mentioned earlier, this drawback can be improved by comparing the lenght of the path indicated by the method as possible cross path, with a certain tolerance limit previously established.

between obstacles or it can be optimized with the help of the model $(M,\, g_{ij})$, in which case the curve $\gamma_{\frac{\varepsilon}{2^n}}$ will play the role of initial approximation.

To sum up this paragraph , we remind that the problem of finding an approximation of the solution of the system (11) is equivalent to the problem of determining the path between two given points of set M, that would avoid the possible obstacles interposed between the two points. Indeed, whenever there is $\gamma : [a,\, b] \to M$, $\gamma(t) = (x(t)\,,\, y(t))$, with $\gamma(a) = A(x_a,\, y_a)$, $\gamma(b) = B(x_b,\, y_b)$, the scalars $x_i = \widetilde{x}(t_i)$, $y_i = \widetilde{y}(t_i)$, $i = 1, .., n$, can be considered forming an approximation of the solution of the system (11), where for each i from 1 to n, $t_i = a + i\delta$, and the values $\widetilde{x}(t_i)$, $\widetilde{y}(t_i)$ respectively, are solutions of the equations $x(t_i) = \widetilde{x}(t_i) + \frac{t_i - a}{b-a} x_b + \frac{b-t_i}{b-a} x_a$ and $y(t_i) = \widetilde{y}(t_i) + \frac{t_i-a}{b-a} y_b + \frac{b-t_i}{b-a} y_a$, respectively. Reverse, if $(x_1, .., x_n,\, y_1, .., y_n)$ is an approximation accurate enough of the solution of the system (11), then the curve $\gamma : [a,\, b] \to M$, $\gamma(t) = (x(t)\,,\, y(t))$, defined by $x(t_i) = \widetilde{x}_\delta(t_i) + \frac{t_i-a}{b-a} x_b + \frac{b-t_i}{b-a} x_a$, $y(t_i) = \widetilde{y}_\delta(t_i) + \frac{t_i-a}{b-a} y_b + \frac{b-t_i}{b-a} y_a$, where $\widetilde{x}_\delta(t) = \sum_{i=1}^{n} x_i \varphi_i(t)$, $\widetilde{y}_\delta(t) = \sum_{i=1}^{n} y_i \varphi_i(t)$, represents one of the paths of access between the points $A(x_a,\, y_a)$ and $B(x_b,\, y_b)$.

6 An empirical method to search the paths between obstacles

In very many concrete situations the problem of establishing the path between obstacles can be solved with the help of some programs that analyse the video images without using any kind of mathematical model. Due to the advantages that result from this, we considered necessary to include in this paragraph the presentation of the functioning algorithm of such a program.

We fix a number $\varepsilon > 0$. If $(x_a,\, y_a)$ represent the coordinates of the pixel on the screen of the computer that designates the departure point A of the automaton, and $(x_b,\, y_b)$ represent the coordinates of the pixel on the screen that designates the point B where the automaton is expected to arrive, then at step 1 we surround the pixel $(x_a,\, y_a)$ with the pixels $(x_a + i,\, y_a + j)$, $|i| \le 1$, $|j| \le 1$, $\max\{|i|\,,\, |j|\} = 1$, $i,\, j \in \mathbb{Z}$, and in a data base (array variable with dimension 2) we retain the coordinates of those pixels from M (having the form mentioned earlier) for whom the function distance d, (built in section 3) is bigger than number ε - in order to simplify the explanation we will denote this data base with $\mathfrak{DB}_A$. After that, we surround the pixel $(x_b,\, y_b)$ with the pixels $(x_b + i,\, y_b + j)$, $|i| \le 1$, $|j| \le 1$, $\max\{|i|\,,\, |j|\} = 1$, $i,\, j \in \mathbb{Z}$, and in a data base (that we will denote by DB_B, for the same reasons mentioned above) we retain the coordinates of those pixels from M (of the form mentioned above) for whom the function distance $d = d(x,\, y)$ is bigger than the number ε, too[2]. In the end we verify if there are pixels common to the two data bases. If such pixels exist, we denote one of these by $(x^*,\, y^*)$. In this situation, the points $(x_a,\, y_a)$, $(x^*,\, y^*)$, $(x_b,\, y_b)$ set an access way from the departure location $(x_a,\, y_a)$ to the arrival location $(x_b,\, y_b)$ of the automaton. We denote by $\gamma(t) = (x(t)\,,\, y(t))$, $t \in [a,\, b]$, the piecewise linear curve (the zigzag path) that at the $t = a$ moment goes through the point $(x_a,\, y_a)$, at the $t = \frac{a+b}{2}$ moment goes through the point $(x^*,\, y^*)$ and at the $t = b$ moment goes through the point $(x_b,\, y_b)$.

Observation *In the actual work hypothesis (image I is regarded as a set of pixels and not as a set of points of dimension 0) this curve is completely included in M.*■

In case that $\mathfrak{DB}_A \cap \mathfrak{DB}_B = \emptyset$ (the set of pixels that surround the pixel of coordinates $(x_a,\, y_a)$ does not intersect the set of pixels that surround the pixel of coordinates $(x_b,\, y_b)$) we go to step 2.

At step 2, we surround the surface occupied by pixels $(x_a + i,\, y_a + j)$, $i,\, j \in \{-1,\, 0,\, 1\}$, with the pixels $(x_a + i,\, y_a + j)$, $|i| \le 2$, $|j| \le 2$, $\max\{|i|\,,\, |j|\} = 2$, $i,\, j \in \mathbb{Z}$, and we memorize (in $\mathfrak{DB}_A$) those from M in which the function $d = d(x,\, y)$ takes values higher than ε. Next, we do the same thing for the region covered by the pixels $(x_b + i,\, y_b + j)$, $i,\, j \in \{-1,\, 0,\, 1\}$: we surround this region with the pixels $(x_b + i,\, y_b + j)$, $|i| \le 2$, $|j| \le 2$, $\max\{|i|\,,\, |j|\} = 2$, $i,\, j \in \mathbb{Z}$, and we memorize (in $\mathfrak{DB}_B$) those (from M) in which the function $d = d(x,\, y)$ takes values higher than ε. Finally, we verify if there are pixels common to the two data bases. If there are such pixels, we denote by $(x^*,\, y^*)$ the coordinates of one of these. Then we verify whether between the points $(x_a,\, y_a)$, $(x^*,\, y^*)$ and $(x^*,\, y^*)$, $(x_b,\, y_b)$ respectively, exist any paths of access. This thing is realized by resuming the algorithm proposed for the case when $(x_a,\, y_a)$ is the initial point and $(x^*,\, y^*)$ is the final point, respectively for the case when $(x^*,\, y^*)$ is the initial point and $(x_b,\, y_b)$

[2] While running the program, because of the finite dimensions of the working surface, it is possible that not all the pixels that we want to use in order to surround the pixel having coordinates $(x_a,\, y_a)$, or the pixel having coordinates $(x_b,\, y_b)$, belong to the screen on which the analysed image is projected. In all these cases it must be implied the fact that in the databases DB $_A$, and DB $_B$ respectively, only the coordinates of the pixels belonging to the working surface will be memorized.

is the final point. In the affirmative case, the reunion of the two paths represents a path of access from location $(x_a,\ y_a)$ to location $(x_b,\ y_b)$.

In the negative case the steps presented above must be resumed for another point $(x^*,\ y^*)$ belonging to the intersection of the two data bases. If for none of the points of the intersection $\mathfrak{DB}_A \cap \mathfrak{DB}_B$ we cannot build path between A and B, or if $\mathfrak{DB}_A \cap \mathfrak{DB}_B = \emptyset$, then we go to step 3, which consists in repeating the operations described in the previous steps.

Observations 1) *After undertaking a number of steps that will not exceed the number of pixels that form the screen of the computer, due to this algorithm, we can decide whether there are or not ways of access between two points $A\,(x_a,\ y_a)$ and $B\,(x_b,\ y_b)$ situated on the surface of the analyzed image. Moreover, if between the points $A\,(x_a,\ y_a)$ and $B\,(x_b,\ y_b)$ there is a way of access, then for a well chosen $\varepsilon > 0$ the algorithm described earlier will help us to find this way, after a finite number of steps, which demonstrates that the algorithm described is utilizable.*

2) *The main advantage of the algorithm presented consist in the fact that for very many practical applications the simple use of them is sufficient to solve the problem related to finding some access paths through the obstacles of a given video image.*

3) *In the cases when we want to optimize the solutions offered by the present algorithm we can appeal to one of the two geometrical models presented in this paper, in which case, the solutions we want to optimize will help determine the initial approximations of the solutions of the system.(11).*

4) *In certain situations (for example, when we want to increase the response speed of the computer used for the analysis of the image $\mathfrak{I}$ or when during the process of tracking down the access path between two given points we want, as well, to take into account the dimensions of the automaton that is moving) the notion of pixel, used to describe the above mentioned algorithm, can be extrapolated at parts of surfaces of larger dimensions.* ■

7 Final comments

Despite their two-dimensional character the methods presented in this paper can be used to solve certain problems to the real world. Indeed, any of these methods can successfully be used to analyze certain images taken from high altitude, such as those taken from a plane or a satellite. In addition to this fact, the two mathematical models can be generalized to the three dimensional case, providing access paths through possible obstacles in space. It is to be noticed that the three dimensional model related to the movement through obstacles can lead to solutions that are not always terrestrial. Thus, it can be only used for dynamic systems that can fly.

References

[1] M. Demi, New Approach to Automatic Contour Detection from Image Sequences: An Application to Ventriculographic Images, *Computers and Biomedical Research*, 27, 1994, 157-177;

[2] B. P. Demidovich, I. A. Maron, Computational Mathematics, *MIR Publishers*, Moscow, 1973;

[3] Y. Hanafiah, M. Yamano, Y. Nasu, M. Ohka, Performance of a Research Prototype Humanoid Robot Bonten-Maru II to Attain Human-Like Motions, *WSEAS Transactions on Systems and Control*, Issue 9, Vol. 2, 2007, 458-467;

[4] C. R. Lucatero, The problem of Robo random motion racking learning algori hms, *Proceedings of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation*, Corfu Island, Greece, February 16-19, 2007 219-224;

[5] P. A. Ravirat, J.-M. Thomas, Introduction à l'analyse numérique des équations aux dérivées partielles, *MASSON* Paris - Milan - Barcelone - Bon, 1992;

[6] M. Šeda, Roadmap Methods vs. Cell Decomposition in Robot Motion Planning, *Proceedings of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation*, Corfu Island, Greece, February 16-19, 2007, 127-132;

[7] O. I. Şandru, A. Şandru, Mathematical filters of video information processing. Automatic image components detection and marking algorithm, *Tensor*, (to appear);

[8] L. Vacariu, F. Roman, M. Timar, T. Stanciu, R. Banabic, O. Cret, Mobile Robot Path-planning Implementation in Software and Hardware, *Proceedings of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation*, Corfu Island, Greece, February 16-19, 2007, 140-145.