

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Using Discrete Geometric Models in an Automated Layout

Leonid V. Markin

Abstract

The application of discrete (voxel) geometric models in computer-aided design problems is shown. In this case, the most difficult formalized task of computer-aided design is considered—computer-aided layout. The solution to this problem is most relevant when designing products with a high density of layout (primarily transport equipment). From a mathematical point of view, these are placement problems; therefore, their solution is based on the use of a geometric modeling apparatus. The basic provisions and features of discrete modeling of geometric objects, their place in the system of geometric modeling, the advantages and disadvantages of discrete geometric models, and their primary use are described. Their practical use in solving some of the practical problems of automated layout is shown. This is the definition of the embeddability of the placed objects and the task of tracing and evaluating the shading. Algorithms and features of their practical implementation are described. A numerical assessment of the accuracy and performance of the developed geometric modeling algorithms shows the possibility of their implementation even on modern computers of medium power. This allows us to hope for the integration of the developed layout algorithms into modern systems of solid-state geometric modeling in the form of plug-ins.

Keywords: geometric model, discreteness, layout, computer-aided design, voxel (receptor), embeddability, trace, shadowing, algorithm, heuristics, multivalued logic

1. Introduction

In the process of designing any technical structures according to literature data, 80–90% of the processed information is geometric information about the shape of the designed product. In some cases, the requirements for the accuracy of the description must be very high, for example, when modeling the flow around objects or automating technological processes (to ensure specified gaps during assembly of products or reproduction-oriented products on equipment with numerical program control). All this causes a wide range of geometric modeling methods used in modern technology.

Theoretical questions of the geometric modeling method from various aspects were investigated in the works of Russian scientists—Valkova K.I., Ivanova G.S., Kotova I.I., Osipova V.A., Yakunina V.I., Rvacheva V.L. etc.—and the works of Robert Fergusson, Stephen Coons, Pierre Bezier, Charles Hermite, Isaac Jacob Schoenberg, Carl de Boor, Ken Versprille, Eugene Lee, Steve Ginsberg, and others. Their works contain both classical and computer-oriented methods of assignment, calculation, and reproduction of geometric forms of the designed objects.

Of course, such an abundance of methods is focused on the description of the geometric shape of heterogeneous technical objects. To classify the geometric models used in the description of design objects, it is advisable to use the approach, proposed by Semenov O.I.-Osipov V.A. in [1], which is based on the structure of the synthesis of geometric objects from their constituent elements. This classifier divides all geometric objects (GO) into two large groups—geometric objects of complex technical form and geometric objects of complex technical structure. Objects of the first group are limited to compartments of surfaces, each of which is described by sufficiently complex analytical equations or systems of equations. These include aircraft fuselages, car bodies, ship hulls, turbine blades, etc. The objects of the second group are combined on the basis of set-theoretic operations (union, intersection, negation) are described, as a rule, relatively simple geometric shapes. In Markin [2], we proved that based on the specifics of the solved project tasks, the number of such groups should be increased to four (**Figure 1**).

The abundance of geometric forms of objects in engineering, construction, design, etc. requires a library of geometric modeling methods adapted to describe the specific features of the shape of geometric objects. Therefore, in addition to the classification of geometric objects, there are classification systems of geometric modeling methods themselves, which can be divided into three classes (**Figure 2**).

Sculptural methods are used to create geometric models of such objects, and the exact analytical description of which is unknown and can hardly be obtained. In addition to design, sculptural methods are widely used in engineering (aviation, shipbuilding, and automotive), when the shape of the surface is corrected not only for esthetic reasons but also on the basis of aerodynamic or hydrodynamic experiment (**Figure 3a**). However, in the end we get an analytical expression of the geometric shape of these objects with varying degrees of accuracy.

The implementation of this method is based on a fairly large library of surface approximation methods using splines, B-splines, NURBS, Koon's surfaces, Hermite, Lagrange, Bezier, etc.

Analytical approximation methods are used to describe the shape of objects consisting of complex surfaces of the second and higher orders. Since the direct computational processing of surfaces of such a complex geometric shape is difficult, they are approximated by areas of surfaces of lower order (planes, cylinders, spheres, etc.) (**Figure 3b**).

Accurate methods of modeling three-dimensional objects are a set of the following known methods:

- Kinematic
- Parametric
- Wireframe
- Piecewise analytical
- Algebra-logical (R-function method)
- The method of “decomposition into elements”
- The method of constructive geometry of the elementary volume

An illustration of these methods of geometric modeling is shown in **Figure 3b** and **c**.

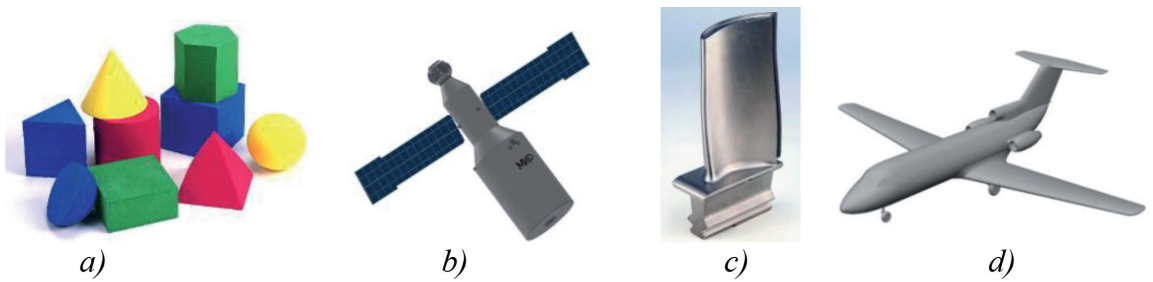


Figure 1.
Classification of geometric objects by level of complexity: (a) primitives, (b) objects of complex technical structures, (c) objects of complex technical forms, and (d) objects of complex technical form and structure.

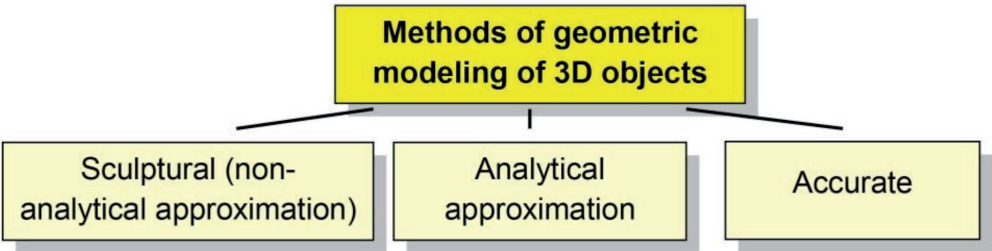


Figure 2.
Classification of methods for modeling geometric objects.

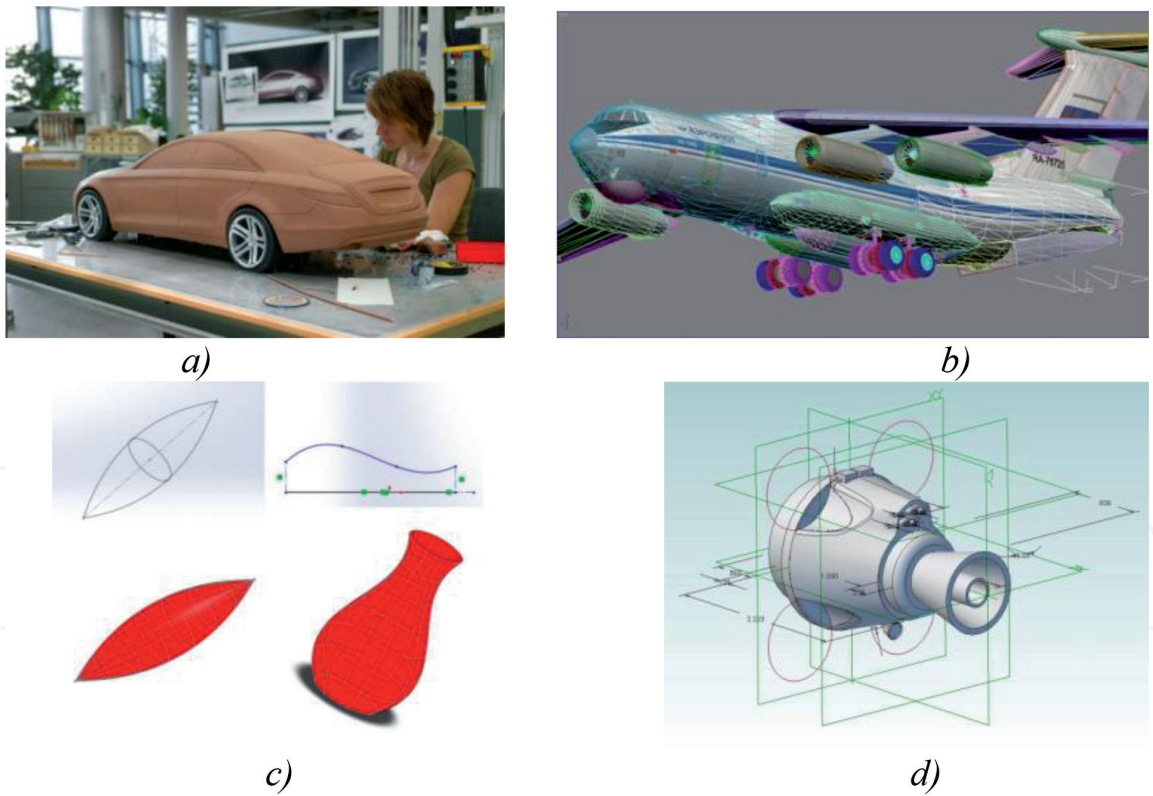


Figure 3.
Examples of technical objects implemented by sculptural methods: (a) analytical approximation methods; (b) exact methods; kinematic (c) and parametric (d).

2. Computer-aided design problems focused on discrete geometric models

Deniskin et al. [3] showed that at present the problem of computer representation of the geometric shape of objects of any complexity can be considered sufficiently solved. However, in some design tasks, the requirement of accuracy of

the description of the form is not the main one. An example of such tasks is the automation of layout calculations. In modern technology, the quality of the layout (i.e., placing the necessary equipment and payload) largely determines the quality of the design.

The development of technology, primarily transport and, especially, aerospace, increasing the density of the layout makes designers constantly improve the methods of design automation. For illustration, two aircrafts of different eras are shown with approximately the same takeoff mass. (30 tons)—ANT-20 (30 years of the last century—**Figure 4a**) and a modern Su-24 (**Figure 4b**). At the same time, much more various onboard equipment is installed on a modern aircraft.

Many requirements must be complied when designing the layout (both automated and traditional methods)—ensuring maximum density, but with the exception of mutual crossing of the hosted objects—to provide a predetermined position of the center of mass of the composed object and the minimum bulk of communications between placed objects, with the exception of the proximity of incompatible objects (such as “hot at work” and “cold”). An additional requirement is to ensure the ergonomics of the layout (the possibility of installation and maintenance of equipment). It is also necessary to ensure the reliability of the functioning of the assembled equipment, which depends on the levels of vibration, pressure drop, temperature, etc., which in turn is determined by the location of this in the designed vehicle.

Taking into account so many of these factors that determine the quality of layout solutions requires either a lot of engineering experience of the developer or the use of information technology in solving this problem. For objects with a high-density layout, even the most careful placement on the drawing does not exclude the possibility of cases of their mutual intersection. To avoid this situation allows the creation of the physical layout, which in scale or life-size simulated layout of a solution. However, despite the attractiveness of physical models, their production is long and expensive. Therefore, the development of methodological, algorithmic, and software processes of automated placement is an actual practical task. Since object placement problems are geometric problems (in geometry they are called positional), their solution should be sought in an extensive library of geometric modeling methods.

Even the first experiments of computerization of design process at the decision of separate private problems have shown their high efficiency. Work on the automation of placement was no exception. The first publications in this area belong to the 60 of the last century and are associated with the names of Russian scientists, L.V. Kantorovich and V.A. Zalgaller, on the cutting of materials by linear programming methods. However, the transition from 2D objects to 3D objects



Figure 4.
Aircraft about the same takeoff weight of different eras: (a) 30 years of the last century and (b) current.

and the complexity of the shape of the placed objects from linear strips to real objects of modern technology caused an avalanche of complexity of the mathematical description of the placement process. In our opinion, the main efforts of scientists-geometers were directed to the study of various aspects of the computer description of the form of technical objects, which are alternative to many classical methods of descriptive geometry. Therefore, now there are no problems to describe the geometric shape of objects of almost any complexity with the necessary accuracy.

The development of methods for automated placement of objects according to specified criteria was much less fortunate. In placement problems, it is important not to accurately describe the geometric shape of the objects being placed, as to solve two critical issues:

- Detection of cases of mutual intersection of the placed objects.
- Generation of options for placing objects in a given space, providing an effective layout. The implementation of the known algorithms of automated placement is based on polygonal models and computational methods of “brute force,” which does not allow them to be implemented in practice for objects of complex structures, even with the modern power of computer equipment.

It can be objected that the use of modern CAD systems allows to carry out modeling of quite complex objects and at the same time to track possible cases of their crossing by means of the CAD system (Figure 5). But in this case, we are not talking about computer-aided design, but only about checking the layout variant already generated taking into account the experience and intuition of the designer.

The problem of computer representation of geometric objects of any complexity can now be considered sufficiently solved. However, other requirements are imposed on automated layout models, of which the accuracy of the form description is not the main thing. We have to choose which is better—an accurate geometric model, automatic layout of which is impossible, or a coarser geometric model, but allows the possibility of automated layout.

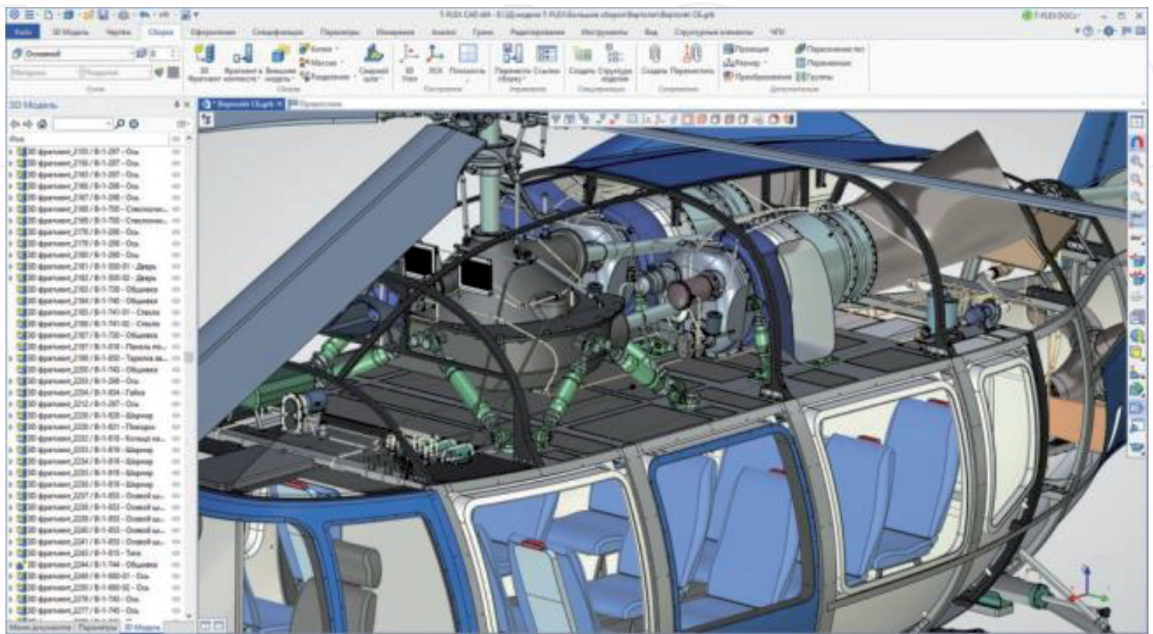


Figure 5.
Modeling layout in CAD systems.

3. Discrete (voxel) geometric models

It is known that the most accurate formal description of a three-dimensional object as a geometric body is its identification with the area of space occupied by it (a point set). However, in this formulation, the problem of the formation of a geometric object (GO) can be considered only theoretically. This concept can be used in practice if as the initial element of the set (E^3) take not an infinitesimal point but, for example, a cube with dimensions $(l \times l \times l)$. In this case, the condition of congruence of all cubes filling the space must be satisfied, and any two cubes must not have common internal points.

The space in this case is called discrete, and the geometric model formed in such a space, respectively, is discrete or voxel model. The term voxel formed from the words volumetric and pixel—elements of the three-dimensional image. Voxels are analogous to pixels in a two-dimensional to three-dimensional space. Voxel models are often used for visualization and analysis of medical and scientific information, as well as in computer graphics (most often in games (**Figure 6**)). Polygonal models inside are empty (and often this is enough—why do we, for example, know what is inside a computer character?). Voxel models are completely filling their insides—volumetric voxel cubes, which can contain additional information about the object.

The discrete method of geometric modeling (in relation to technical applications) was described in the early 70 years of the last century by the Belarusian scientist, Zozulevich in [4], but in those years it did not spread due to the limited capabilities of computers in memory and performance. Although he and the team of his employees solved certain applied problems using this method, it was impossible to count on the effective use of voxel models with computers of those years with 16-bit architecture and 32–128 kb of RAM.

The basis of such models is an approximate representation of a geometric object in a field or voxel space. For the flat case, the voxel field is a uniform rectangular network $m \times n$, each cell of which is considered as a separate voxel, which can have two states—“0” or “1.” Mathematically, the voxel geometric model is described by

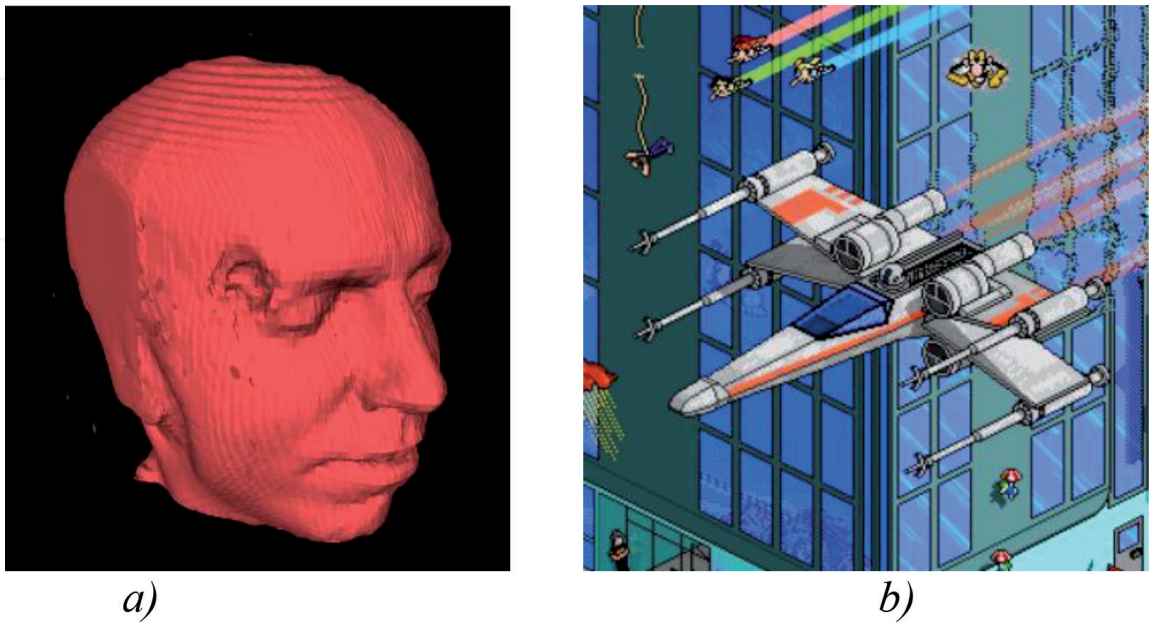


Figure 6. Voxel models in medicine ((a) the result of computed tomography) and in computer graphics ((b) in computer games).

the set $A = \{a_{ij}\}$, where the voxel is considered unexcited if the object boundary does not pass through it and it does not belong to the inner region (**Figure 7**).

3D objects are described by a three-dimensional matrix $A = \{a_{i,j,k}\}$ with dimension a of $m \times n \times p$ (**Figure 8**). Obviously, the accuracy of the description of the geometric shape of the object depends on the chosen discreteness of the voxel matrix.

The author of the method Prof. Zozulevich D.M. called his method “receptor” by analogy with the receptors of the human brain; each of which can be either excited or not. Other names of this method are also known—“matrix,” “binary,” “enumeration of space elements,” etc. In its geometric essence, the receptor (voxel) method is a special case of the analytical approximation method, which is used to describe three-dimensional objects that include complex surfaces of the second and higher orders. Since computational processing of such surfaces is difficult, they are approximated by areas of surfaces of lower order (planes, cylinders, etc.).

Research and development of voxel geometric models for various applications was carried out in the works of Russian scientists Gorelik AG, Gerasimenko YP, Klishin VV, Romance YA, Pashchenko OB, and Toloka AV, as well as a number of foreign authors—Gargantini I, Requcha AAG, Si Thu Lin, Nyi Nyi Htun, Kyi Min Han, Ye Win Tun, and a several others.

Here, it should be noted that very close in research ideology of Nazarova KM, Ratkova SI, etc., in which as basic object shape is not classic receptor in the form of a cube or parallelepiped, and more complex shapes—for example hexahedron.

The voxel method has both advantages and disadvantages. The main advantage is the homogeneity of computational algorithms and a very simple detection of cases of intersection of objects with each other.

The obvious disadvantages include the discreteness of the model and the need for large amounts of computer memory for its implementation. However, now the increase of computer memory to any volume is neither technical nor economic.

Another drawback is that the voxel geometric model is never primordial. Placed and already placed products are described by the designer, usually parametric geometric models (i.e., specifying the type of object and its parameters—a sphere with radius R , a parallelepiped with dimensions $a \times b \times c$, etc.). Examples of parametric models are shown in **Figure 9**. Therefore, there is a need for an additional software module “parametric model” ↔ “voxel model.” However, now there are ways to form a voxel matrix directly on the solid-state model created in any of the CAD systems (**Figure 10**).

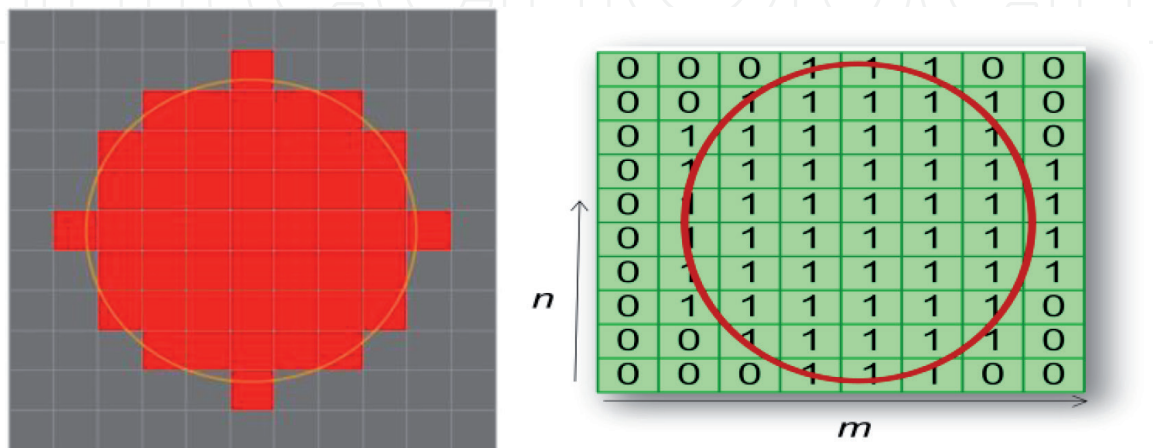


Figure 7.
Voxel model of a 2D object.

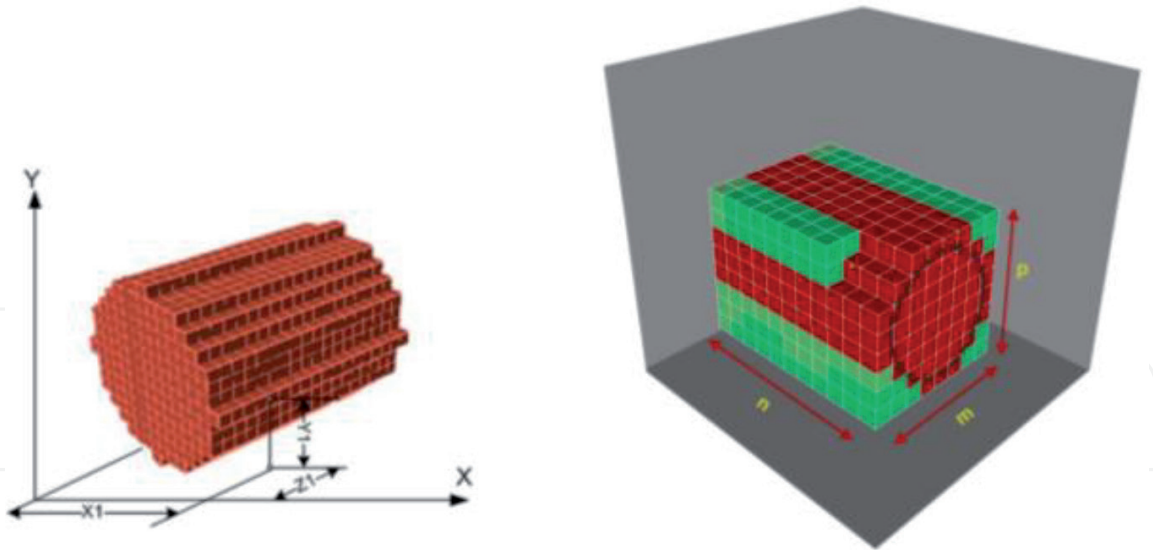


Figure 8.
Voxel model of a 3D object.

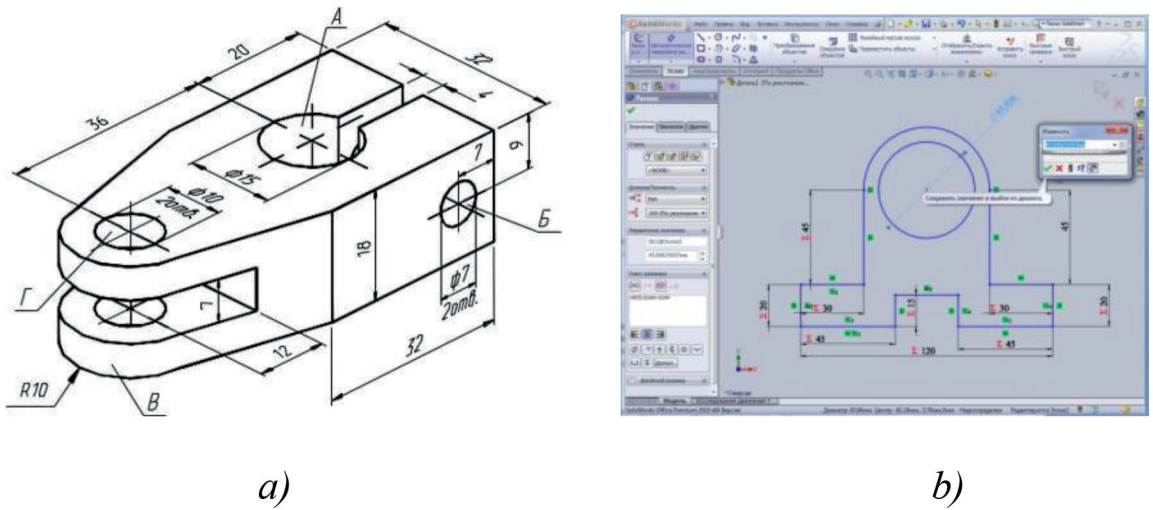


Figure 9.
Setting parameters of geometric objects in traditional drawings (a) and computer modeling (b).

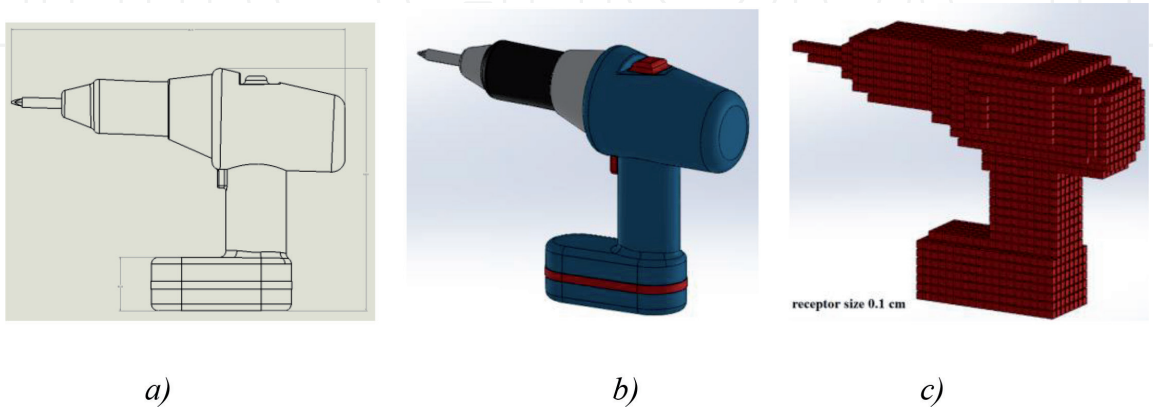


Figure 10.
Stages of construction of voxel model of an electric drill: (a) initial drawing, (b) solid model from CAD system, and (c) voxel model.

4. Examples of computer-aided design problems using voxel geometric models

4.1 Problem “rearrangement”

Consider the problem of placing additional equipment in the technical compartment of the vehicle (the problem of pre-assembly). This problem is often found in the practice of design and from a geometric point of view is reduced to the problem of analyzing the geometric shape of empty spaces among previously placed objects.

We will solve the problem of additional placement in the following statement—there is a closed area (e.g., the technical compartment of the aircraft) with the equipment already partially placed in it (**Figure 11a**). There is a set of equipment that needs to be “repositioned” (**Figure 11b**). The possibilities of “additional placement” are determined by the shape and size of still unfilled spaces between already composed objects.

For a person, such a task “re-arrangement” of objects will seem quite easy—he can easily “by eye” determine how several volumes relate to each other and which object fits in another and which does not. To do this, he mentally classifies the object by shape (almost a ball, almost a cylinder) and then mentally correlates their sizes. Unfortunately, this operation of pattern recognition, which is so simple for a person, is of considerable complexity even for modern computers.

The most important issue is the class of geometric objects allowed to be placed. Typically, from a geometric point of view, such equipment is either primitives or a combination of primitives. Assume that the objects to be placed are a composition of primitives that describe the shape of the instrumentation quite well (this can be seen in **Figure 11b**). The technical literature provides data that the composition of primitives effectively describes 95% of the instrument equipment.

The advantage of the voxel geometric model is the ability to identify empty spaces in the layout for subsequent placement of objects not yet placed in them. If there is an empty space among the assembled objects, then on the corresponding section of the voxel matrix it is relatively easy to identify as “clumps” of zero voxels. For this purpose, an algorithm was developed to determine the center of a certain free area (flat) and its dimensions.

To do this, scan the rows and columns to identify the largest cluster of inextricably located receptors and identify their centers. To analyze the individual cross

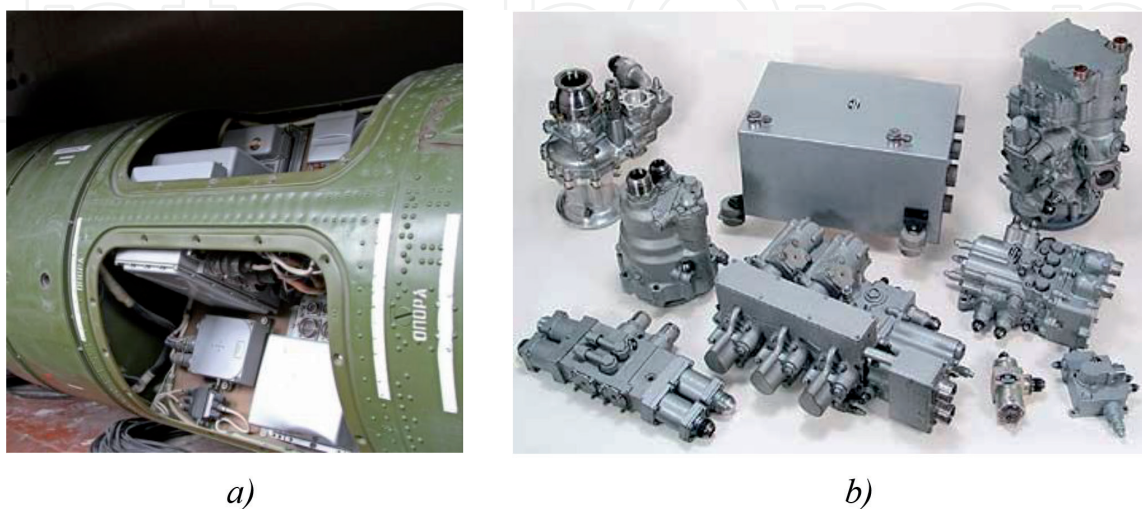


Figure 11.
Technical compartment of the aircraft: (a) “redeployable” and equipment (b).

section of the object, we use the transition from the geometric shape of the object (in the form of a discrete set of receptors) to the “feature space” adopted in the theory of pattern recognition. Such a feature space for us will be the hodograph of the function of the radius of the vector from the center of the section (**Figure 12**):

$$F = R_i(\varphi_i)$$

where R_i is the current radius-vector length for the i receptor and φ_i is the current radius-vector angle for the i th receptor.

After constructing the function $R_i(\varphi_i)$, its analysis begins. If the shape of this region were a circle, the function would be an ideal straight line whose height would show us the radius of this circle (**Figure 12b**), if a regular polygon is a “saw” with the number of vertices equal to the number of sides. The coordinates of the vertices on φ will determine the aspect ratio of the rectangle. To do this, we use the method of testing statistical hypotheses, implemented as a computational software module.

On the real results of the analysis of the hodograph function, naturally superimposed “noise” due to the discreteness of the description of the component objects. An example of such real hodographs of a plane slice of the objects being assembled, statistically identified as slices of “polygon,” “cylinder,” and “sphere,” are shown in **Figure 13a–c**, respectively.

An illustration of the software implementation of this method is presented in **Figure 14**. The obtained results are implemented in the framework of a graphical shell written in C#. If we have a desire to place in this area not a parallelepiped of the maximum size, but, for example, a sphere of a certain radius, then it becomes a full participant of the scene, and after pressing the “analysis” button, a new definition of the configuration of unfilled spaces occurs.

The solution of the task was carried out within the framework of the dissertation research by Situ Lin (Republic of the Union of Myanmar), a postgraduate student of MAI and described in [5].

4.2 Task of tracing channel

The purpose of designing the channel surface is the delivery of a certain trajectory of a material carrier (liquid, gas, electrical energy from one point (entry point) of the technical product to another (exit point)). In this case, one of the tasks of the layout is to solve the problems of tracing, i.e., designing communications between

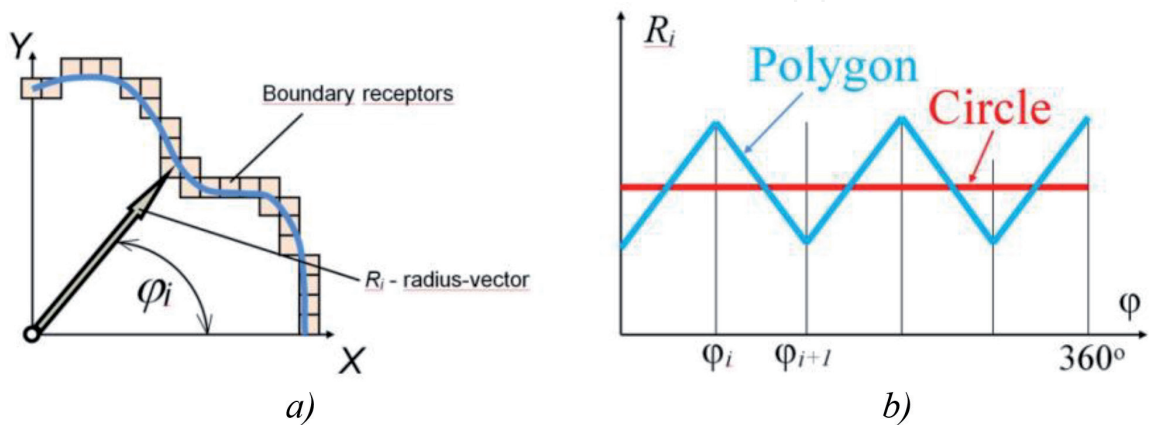


Figure 12. Construction of the hodograph of the radius-vector function (a) and analysis of the unfilled area of space (case-polygon) (b).

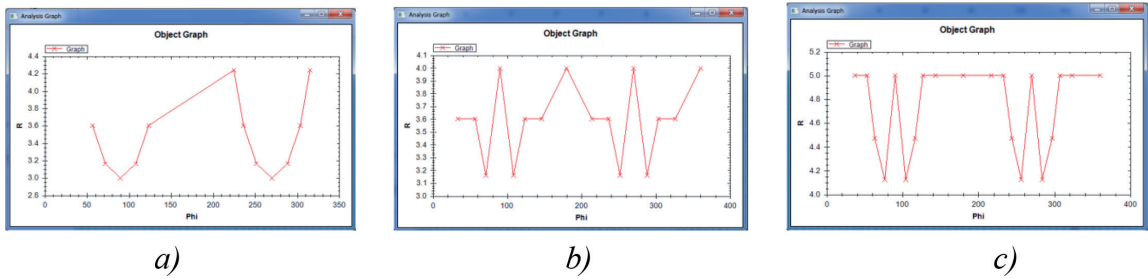


Figure 13.
View of the hodograph of the slice function from for the polygon (a), sphere (b), and cylinder (c).

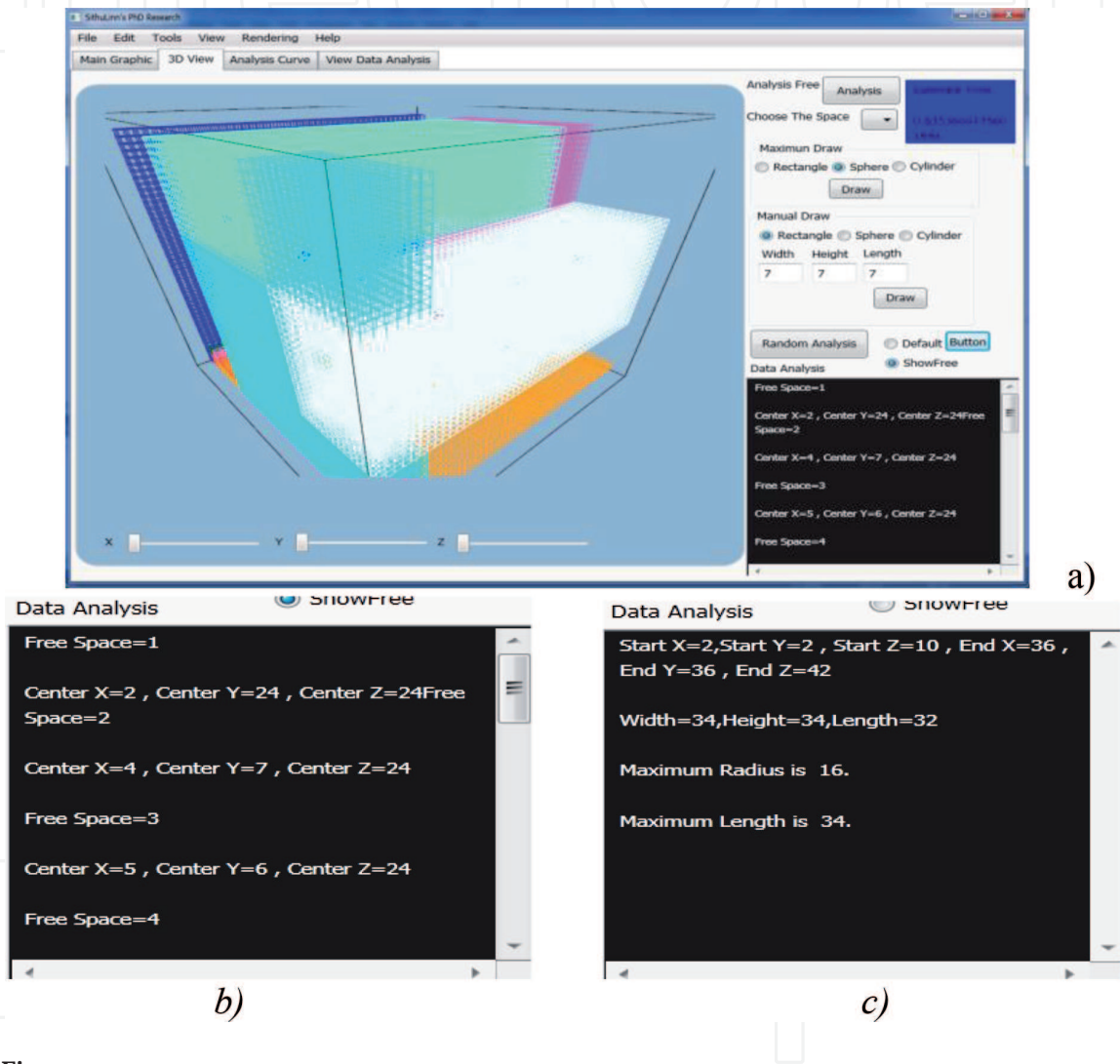


Figure 14.
Determination of the configuration of empty spaces: (a) visualization of space, (b) output of data on the size and position of the free area, and (c) assessment of the possibility of inscribing different shapes.

already placed objects. Such problems are quite difficult to formalize and difficult to solve because of their inherent multi-extreme nature.

A special and much more complex type of trace is called “solid” trace, that is, such a case when the dimensions (connecting elements) of the trace are comparable to the dimensions of the components. In practice, this is the design of pipelines, air ducts, and other elements of transport systems (**Figure 15**).

To solve this problem, two main approaches to tracing are used, which are determined by the metric used. A metric is a rule that determines the distance between two points in a given space. The first approach is to use the Euclidean metric. In this case, the trace is drawn in the direction of the shortest distance between the

entry and exit points (**Figure 16a**), and the length of the trace is determined by the Pythagorean theorem.

The second approach is to use the Manhattan metric, in which the trace is drawn in the direction of the coordinate axes (**Figure 16b**). In this case the trace is much longer than using the Euclidean metric, but the approach itself provides additional mathematical possibilities for the design of the trace. It is used for tracing large integrated circuits and printed circuit boards; this metric is used by industrial automated tracing systems P-CAD, TopoR, and others. A typical example of the result of wiring in such programs is shown in **Figure 16c**. From this figure it can be seen that the tracing in such systems is carried out either by the Manhattan metric or at 45° angles.

In our opinion, the problems of trace design according to the Manhattan metric are the most developed in the theory of geometric modeling of traces. This is due to the extreme practical importance of solving these problems in the automated wiring of printed circuit boards and large integrated circuits. Fundamental in this field are the theoretical studies of Russian scientists Abraitis LB, Bazilevich RP, Petrenko AP, Tetelbaum AY, Selyutina VA and others, as well as foreign scientists EW Dijkstra, Judea Pearl, Ira Pohl, Daniel Delling, Peter E Hart, etc.

However, the use of the Manhattan metric is unacceptable for channel design, since due to viscous friction, the sharp bends of the channel make it difficult to move liquid or gas through the channel, turning the flow energy into heat. The current line that determines the direction of travel through the channel is called the main guide line (GNL). It is the axis of the channel and is given either by the equation of the spatial curve or by a discrete set of points. However, tasking this parameter alone is not enough—it is necessary to specify the shape and area of individual cross sections of the channel (**Figure 17**). By controlling the position of the GNL



Figure 15.
An example of laying the trace: (a) electric furnace and (b) the air duct channel of the car.

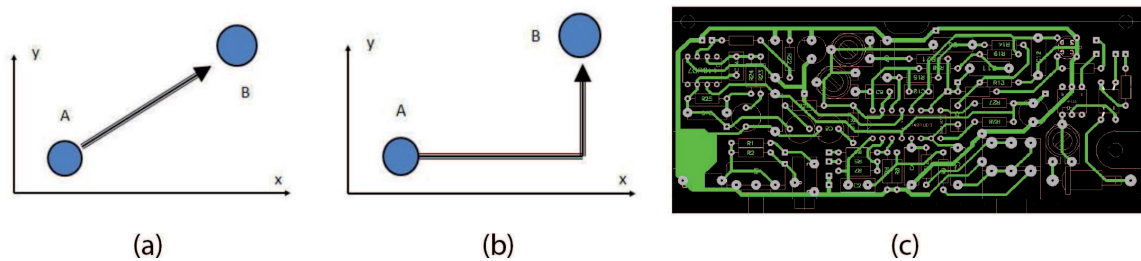


Figure 16.
Designing trace with different metrics.

and the shapes and sizes of the cross sections, we can provide the specified designer characteristics of the flow of liquid or gas through the channel.

Let us complicate the task of designing. If earlier the channel was first designed according to the specified characteristics and then it was already placed, then we are trying to design a channel with the specified characteristics, “inscribed” in an already existing layout. Let there be a rectangular area with dimensions $X \times Y$, in which the areas of prohibition are located. The entry point A and exit point B of the channel are specified (**Figure 18**). It is necessary to draw a trace between the given start and end points A and B. Obviously it is that in such a statement the problem does not always have valid solutions.

From **Figure 18** it can be seen that there are quite a lot of options for its passage. At first glance, of all the traces in **Figure 18**, the one that is shorter will be better. However, it is known from design practice that hydraulic losses are minimal not in the short but in the main channel. Additional requirements are possible, for example, to provide a specified gap between the channel and other elements of the layout.

A significant problem in solving the tracing problem is to avoid obstacles, which are already composed objects or communications between them. The big advantage of the voxel approach is the ease of detecting an obstacle by voxel code (0 or 1). The simplest approach is to ignore obstacles before encountering them. Such an algorithm would look something like this: choose the direction to move toward the target, and move until the target is reached and the direction is free to move (**Figure 19**). Given that the transverse area of the channel can repeatedly exceed the size of one voxel, it is possible that a valid trace option in this layout situation is not at all.

The known trace algorithms closest to our approach are described in the following [6, 7]:

- Dijkstra’s algorithm
- Algorithm A* “A the asterisk”

The simplest approach implemented in these algorithms is to ignore obstacles before encountering them. The structure of the choice of direction of movement is determined by the rule to move back to choose a different direction in accordance with the strategy of the traversal.

This allows us to argue that these algorithms contain elements of artificial intelligence (AI), since the solution is chosen according to the predicative principle of “If”-“Then.”

In the works of Dijkstra EW, Donald Ervin Knuth, Thomas H Cormen, etc., various obstacle avoidance strategies (heuristics) based on both random search and artificial intelligence algorithms are analyzed. Each of them has both its limitations and areas of preferred application. Examples of different trace algorithms are shown in **Figure 20a**. **Figure 20a** shows that although the well-known Dijkstra algorithm was able to pave the way from the start to the end points, it did not do it rationally, making an extremely many unnecessary movements and repeatedly unnecessarily changing the direction of travel.

Voxel algorithm A* operates in a more reasonable way (**Figure 20b**) and is commonly used to find the optimal shortest path. But this algorithm is not able to take into account the given gap (δ -neighborhood)—the route may pass too close to the areas of prohibition. Thus, the search for the path of the trace by the algorithm A* gives the best results but does not always provide a solution to the problem. In

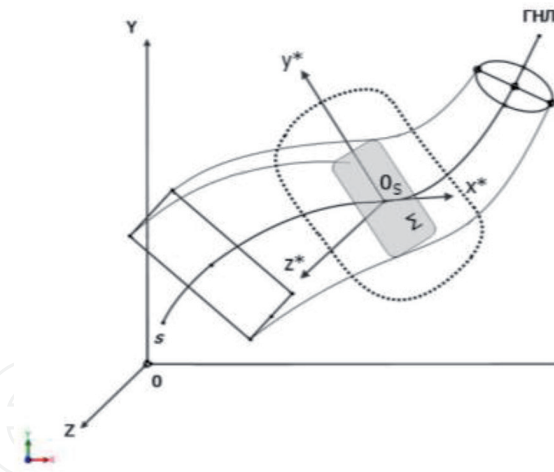


Figure 17.
Geometric parameters that determine the shape of the channel.

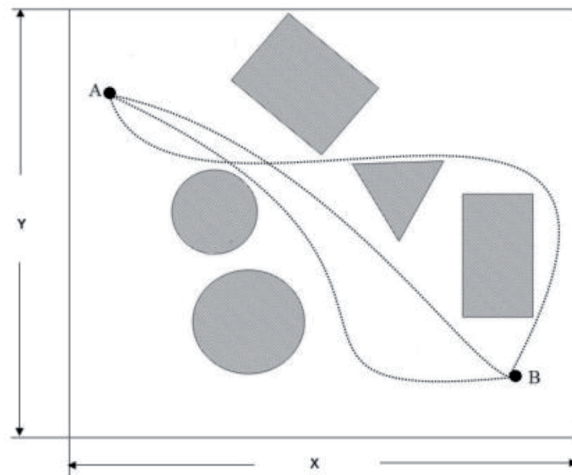


Figure 18.
Finding a rational path between two endpoints A and B in a 2D formulation without taking into account the size of the trace and the restrictions on smoothness.

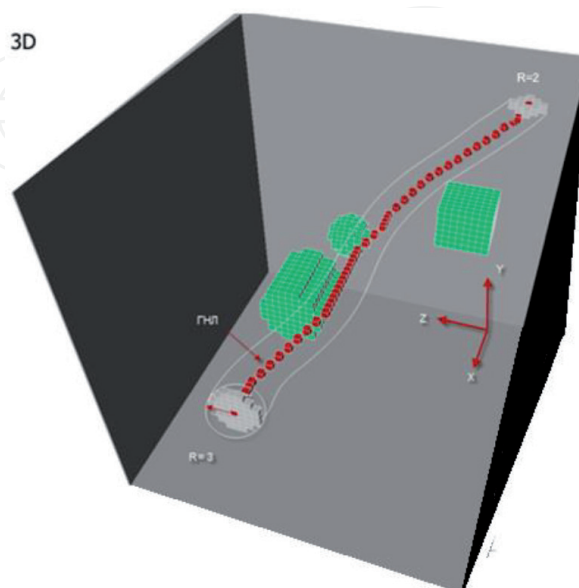


Figure 19.
The principle of obstacle avoidance in the construction of voxel method.

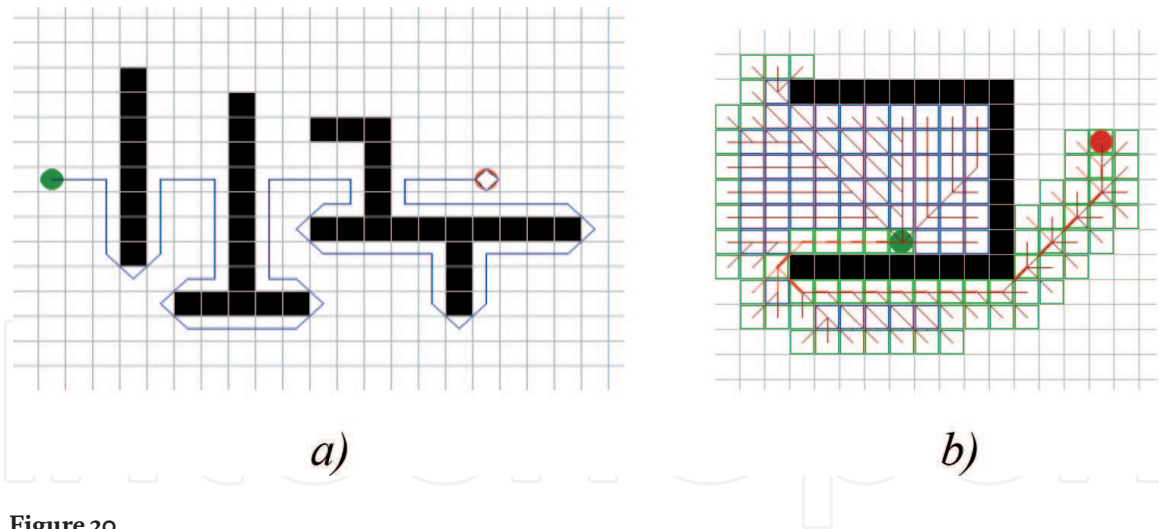


Figure 20.
 Examples of tracing when bypassing various obstacles in 2D staging.

addition, the algorithm A^* is not able to design trajectories with a given degree of smoothness, since such a task has never been set before.

As it was already noted earlier, the strongest side of the algorithm A^* , which led to its choice by us as a prototype, is the possibility of optimal and heuristic trajectory search. The literature shows that in many cases heuristic search works better than other search strategies. The heuristic function of the algorithm determines the choice of the search direction to the target vertex. If the heuristic function is valid (that is, does not exceed the minimum cost of the graph to the target vertex), then the algorithm A^* is guaranteed to find the shortest path. The modified algorithm A^* uses a set of heuristics that provide for multidirectional search. The search for direction is conducted not as usually on 4 and 8 directions (respectively in a plane and space (**Figure 21a** and **b**)) and in 8 directions, if the construction of the channel occurs in the plane and 26 adjacent vertices in the design of the spatial channel (**Figure 21c** and **d**).

To implement the proposed trace algorithm, the Advanced Pathfinder System (APS) program was written in Microsoft Visual Studio 2010, using the C# programming language. The solution of the task was carried out within the framework of the dissertation research by Nyi Nyi Htun (Republic of the Union of Myanmar), a postgraduate student of MAI and described in [8]. With this program, you can:

1. Use the improved algorithm A^* and find a rational trace between two points in 2D and 3D spaces, taking into account the areas of prohibitions.
2. Carry out smoothing of the trace received at the previous stage either on any set radius or on the maximum possible radius with the subsequent check of accomplishment of a condition that the minimum radius is not less than the set R_{min} .
3. Ensure that the trace passes at the specified minimum distance δ from already placed objects and areas of prohibition.

Testing showed significantly higher performance of our modified algorithms than the original one. It should be recognized that the described program, which implements the voxel method of body tracing, is not integrated into any existing CAD system, so the geometric information is entered into the program in parametric form. This reduces the performance of the build process. Integration of this program with any CAD system in the form of a separate built-in calculation module is an actual technical task.

4.3 Task of solar layout

When designing the SC, the question on estimation of the effective area of solar panels arises, taking into account their inevitable shading by each other and other structural elements of the spacecraft SC (**Figure 22**). All this significantly limits the functionality of the SC. When designing SC or ground-mounted solar power plants, we have to decide on the area of solar panels—if there are a small number of panels, then the solar energy absorption will be small, and if there are too many, they will work inefficiently, shading each other (not to mention the additional costs for them and increasing the mass of the entire SC). Therefore, the solution of this question can be considered as an optimization problem of mathematical programming.

Voxel geometric models do not require complex formulas or logical constructions for their implementation. However, their practical implementation has its own specific complexity. In addition to the need to convert the initial parametric model specified by the constructor into a receptor model, the complexity is conditioned by the need to take into account the position and value of each voxel (out of many millions in the voxel matrix), as well as to create a mechanism for visualizing the results. The solution of the task was carried out within the framework of the dissertation research by Kui Min Khan (Republic of the Union of Myanmar), a postgraduate student of MAI and described in [9].

An essential feature of our approach is that we will not use the classical voxel matrix (filled with “0” and “1”) in the calculations, but a *multi-digit* one, in which additional codes will be added. Specifically, it will be three-digit—“0,” free space, “1,” space occupied by the space station residential module, and “2,” space occupied of solar panels (**Figure 23**).

Using a multi-digit voxel geometric model allows you to proceed directly to the calculations of shading. We will move a slice of the voxel matrix with thickness of 1

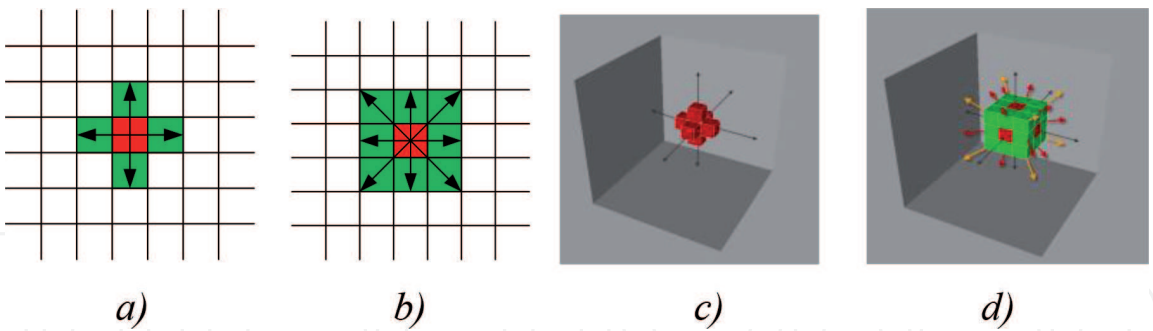


Figure 21. Increasing the directions of the path search in the original and modified algorithms.



Figure 22. Partial shading of solar panels in space on the International Space Station (ISS).

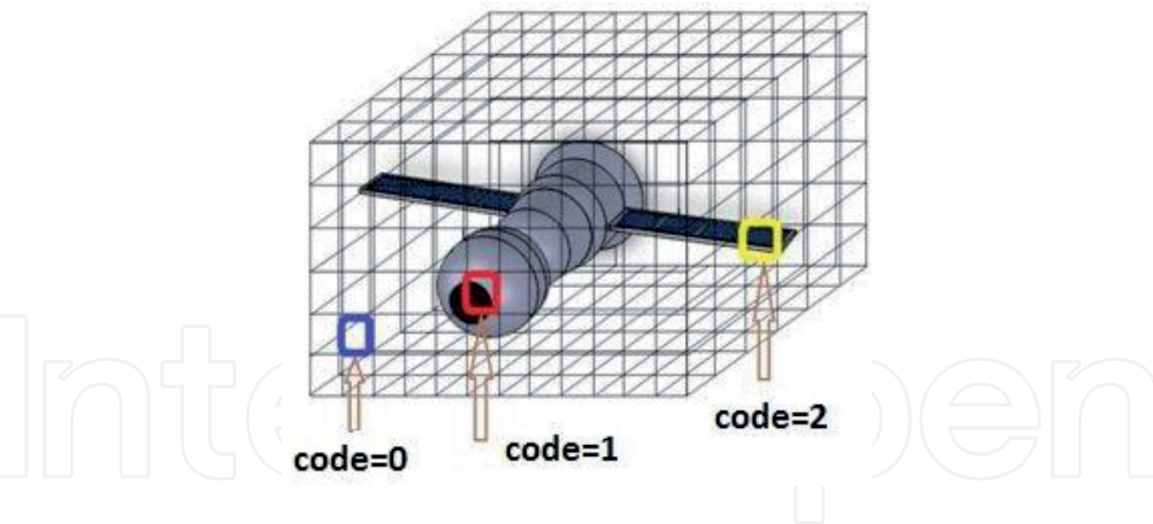


Figure 23.
Representation of SC by multi-digit voxel matrix.

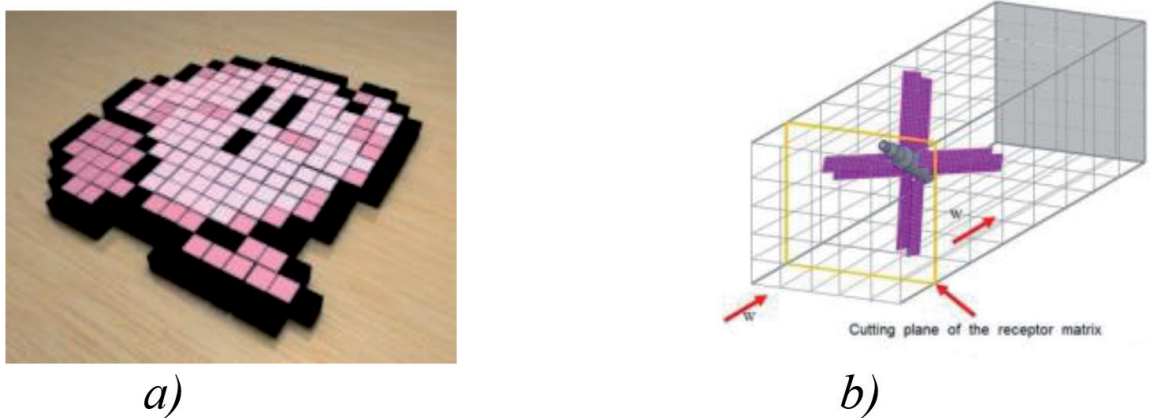


Figure 24.
Single-layer slice of the voxel matrix (a), moving this slice along the voxel matrix (b).

voxel (**Figure 24a**) as a cutting plane along the coordinate plane from the beginning to the end of the voxel matrix (**Figure 24b**).

In **Figure 25a**, it can be seen that on each slice we can put each specific voxel in accordance with either “1” (if it coincides with the body of the SC) or “2” (if it coincides with solar panels). If there is no match with any elements of the SC, the value of the voxel on the slice remains the initially set value of “0.” Looking at a single-layer slice of the voxel matrix from the direction of the energy flow W (**Figure 25b**), we see the layer filled with nonzero code that allows to calculate the total area of space (number “0”), the total area of the habitable modules of the SC (number “1”), and, most interestingly for us, the total area of solar panels (number “2”). Next, everything seems to be simple—summing up the area of voxel with codes “2” on all sections, we get the area of unshaded zones of solar panels.

In this calculation model, there are situations when along the thickness of the solar panel may be not one, but several layers of the voxel matrix (for example, 4 layers), resulting in the unreasonable increase of the effective area of solar panels by four times. It is also necessary to exclude unreasonable repeat account of already screened objects. To do this, we introduce an additional code “3” in the voxel matrix, which will exclude the account of the areas of the corresponding voxels. The essence of the model change is that once absorbed part of the energy flow should no longer be taken into account. Therefore, starting with some slice of the voxel matrix, everything that follows this slice, the element with the code “2,” is forcibly

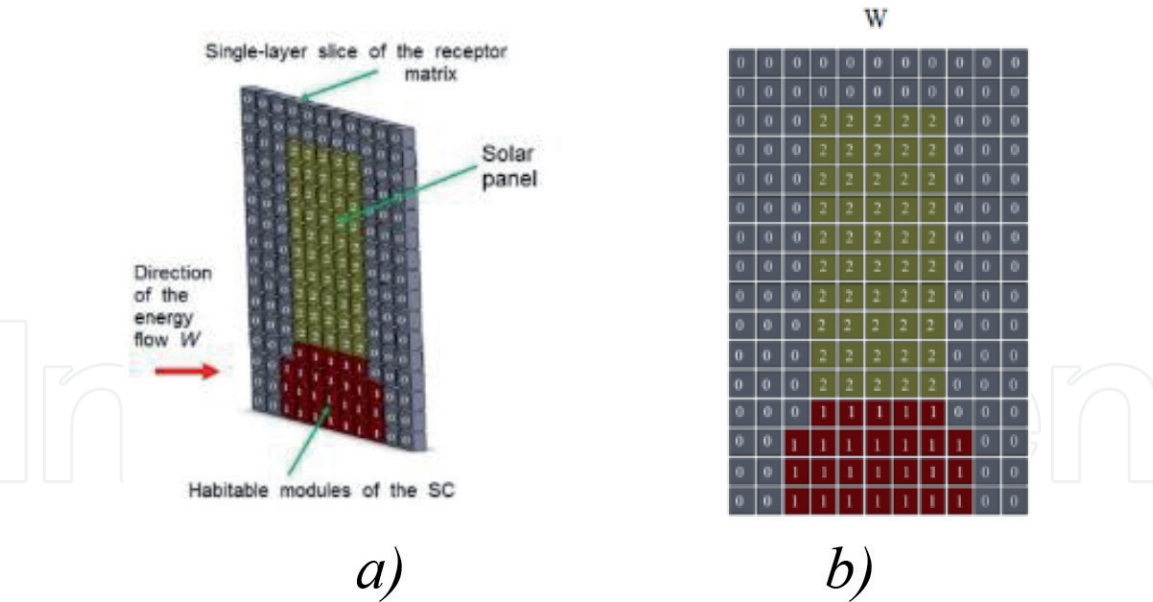


Figure 25. Single-layer slice of the receptor matrix (a), the view of this slice toward the flow (b).

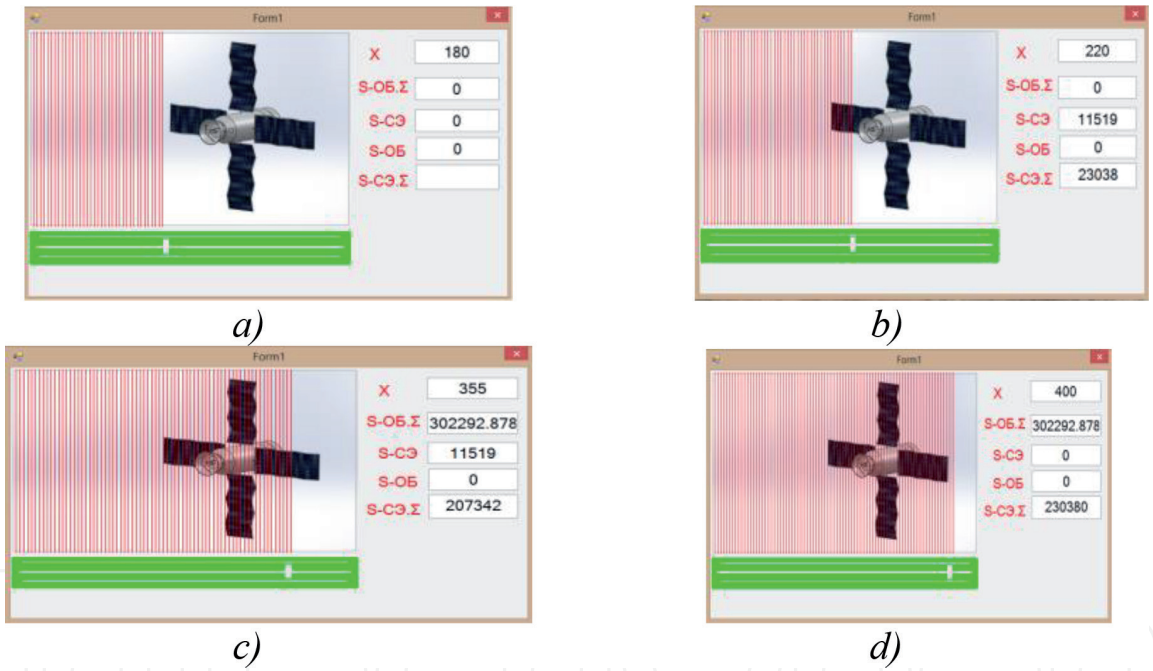


Figure 26. Scanning stages of 3D model SC with inclined position for calculation of sectional areas of solar panels and SC body.

filled with the prohibiting code “3,” which does not allow the use of voxels with this code in any calculations.

However, the shading of solar panels in the W direction, reducing their efficiency, is possible not only by other solar panels but, in some cases, by other elements of the SC (e.g., the body). Therefore, we make another change in our model—filling the entire voxel matrix in the direction of W with codes “3” after the first detection on the slice of any element of the SC. As in the previous case, the entire remaining part of the voxel matrix in the direction of the energy flow W is filled with codes “3,” which excludes the participation of voxels with this code in the calculation of the effective area of solar panels S . Therefore, in the modified (4-digit) voxel model, voxels with the code “3” do not participate in any area calculations.

On the basis of the geometric model described above, a software package was created, implemented in the C# language, allowing to simulate the effective area of solar concentrators. At the same time, a graphical shell is developed that visualizes the calculation process and the calculated parameters of the effective area.

The work of the software package is as follows. After entering the information about the geometric dimensions of the station and solar panels (in parametric form), a layer-by-layer scan of the sections begins. A 2D matrix is formed in each layer, the form of which was previously shown in **Figure 24b**, from the 3D matrix, in which our entire object (SC) is immersed. In each section (slice) of the voxel matrix, the areas of the current section of the solar panels, the effective (cumulative) sectional area of the solar panels, and the cumulative sectional area of the body of the space station are calculated (although this parameter has no practical value for us). **Figure 26a** shows that the cutting plane of voxel matrix has not yet reached the model of SC, so all the sectional areas are zero.

Figure 26b shows that the cutting plane already passes on the SC itself, crossing both the solar panels and the SC body, so specific calculated values are obtained in each section of slice area, which will be visualized in the corresponding program windows. And finally, **Figure c** shows that the cutting plane completely passed through all 3D model of SC, therefore both current and the cumulative sums of the areas will not change any more in the program windows. Thus, we have solved the task—to determine the total visible area (from a certain angle) both of the body of the space station and its solar panels.

5. To evaluate the accuracy and efficiency of voxel model

The advantage of the voxel method is the simplicity of determining many geometric parameters and solving many geometric problems of processing three-dimensional objects. However, when using voxel model, the shape of the object is described approximately. The desire to increase the accuracy of the description in r times leads to an increase in the necessary computer memory for the implementation of the method in times where q is the dimension of the modeling space.

To determine the appropriate accuracy of the representation of 3D objects voxel matrix as a test example, consider the most unfavorable for the accuracy of the description of the case of the ball. Let us assume that a ball with a radius of 500 mm is inscribed in a voxel matrix with a variable number of voxel, which varies in the range from 1000 to 25,000 (**Figure 27a**).

For **Figure 27b**, the results of the calculation of the value of the most probable absolute error depending on the number of voxels for this matrix with dimensions $1\text{ m} \times 1\text{ m} \times 1\text{ m}$ are given. It can be seen from this graph that the error of the voxel model shape description is expected to depend on the voxel size, which, in turn, is determined by the number of voxels in the matrix. There is a threshold (in our case it is about 5000 voxels), with a decrease in which the absolute accuracy of the description of the form is sharply reduced. In [10] it is shown that for the most typical sizes of technical objects, even with a not very high number of voxels (about 5000 on each axis), the relative accuracy of the description remains quite high (below 0.05%), which is quite enough for most technical applications.

Since receptor models are discrete by definition, the discussion of the results raises the question not only about the accuracy of their description but also about the computational resources required for their implementation. Studies show that with a seemingly huge number of computational operations, they are performed surprisingly quickly. This is probably due to the homogeneity of the calculations and the use of only the RAM of the computer for their execution. The time of

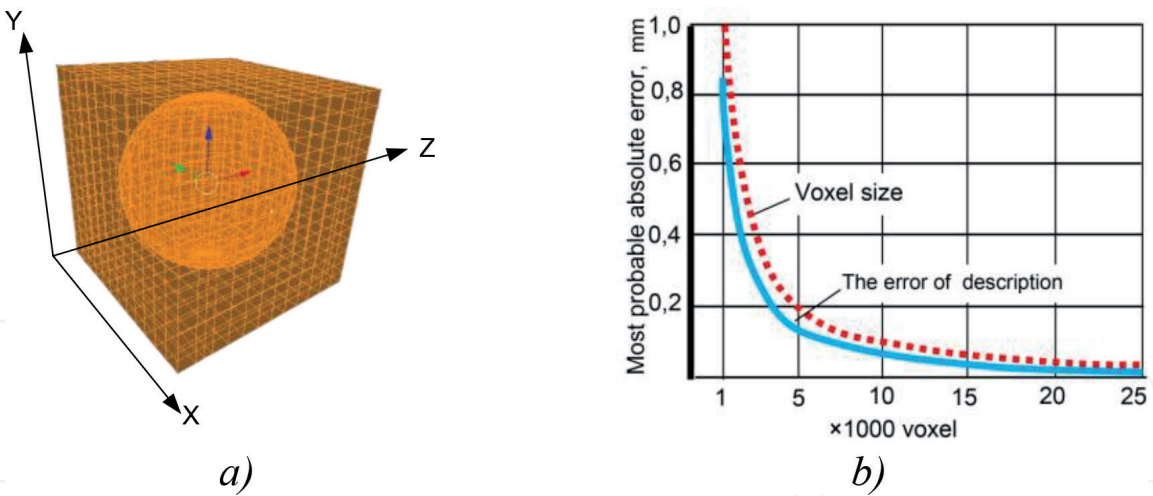


Figure 27.
Results of calculation of the error of the shape description by the voxel matrix.

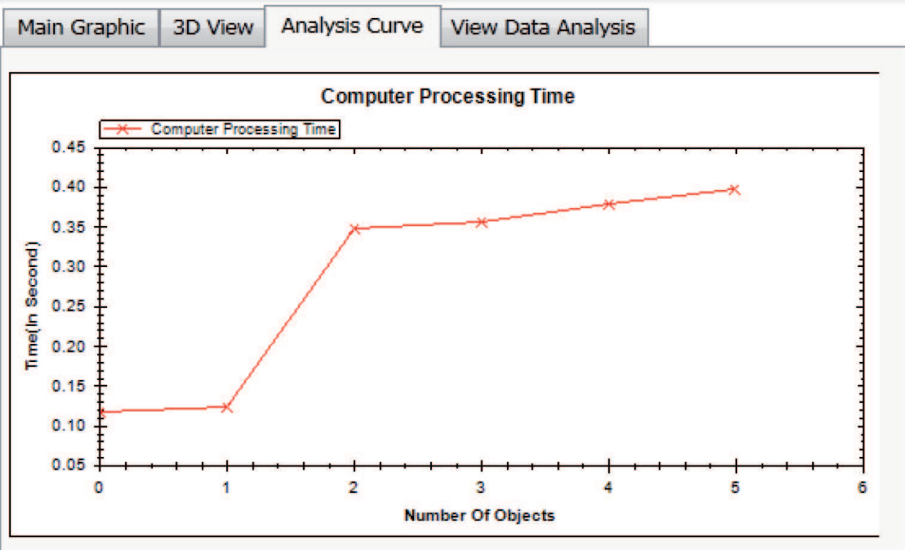


Figure 28.
CPU time required to determine empty spaces depending on the number of objects already placed.

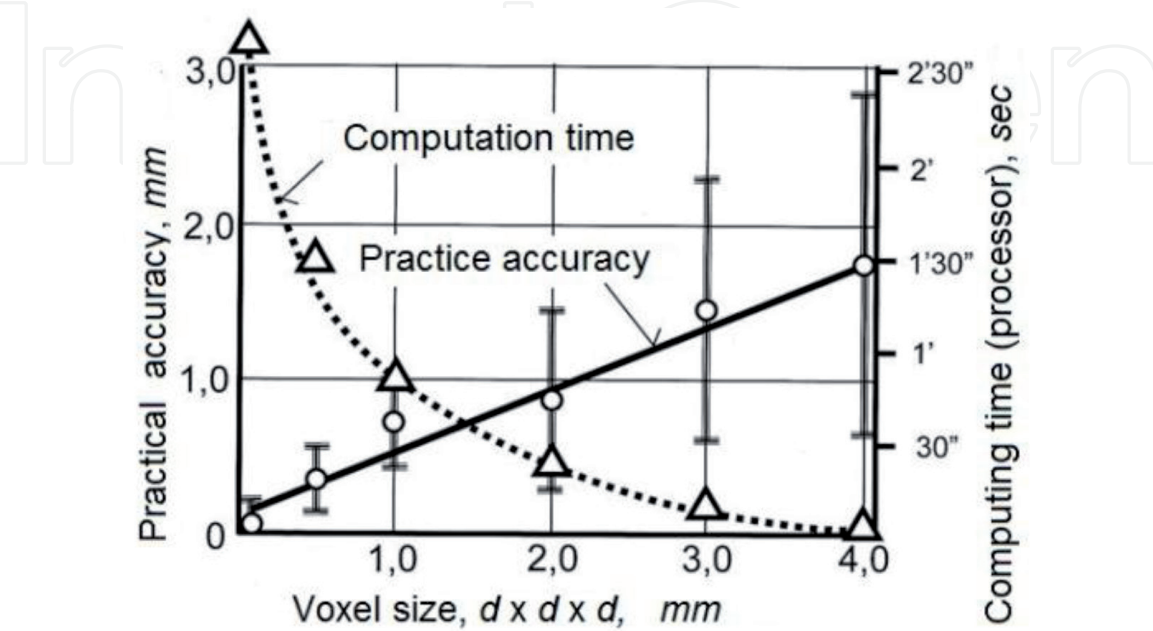


Figure 29.
Calculated characteristics of voxel geometric model accuracy in designing a channel between obstacles.

computer processing of voxel models is determined by both the hardware capabilities of the computer and the parameters of a particular scene—the number of voxels and the number of objects already placed. In **Figure 28** the estimation of processor time in solving the problem of recognition of a flat area with a fixed number of voxels (1000×1000) is given. **Figure 28** shows that the CPU time is still relatively small (fractions of a second), even for a medium-power computer. For a spatial scene, the time to obtain a solution increases by about an order of magnitude, but still remains acceptable for interactive mode. But it should be borne in mind that for voxel geometric models are possible methods of decomposition and parallelization of calculations, which will significantly speed up the calculation time.

Evaluation of the accuracy and performance of the channel design between obstacles in the test example using a voxel matrix size $1 \text{ m} \times 1 \text{ m} \times 1 \text{ m}$ is shown in **Figure 29**. They also show acceptable practice accuracy and simulation time.

6. Conclusions

Discrete methods of geometric modeling, which include voxel, can effectively track the cases of intersection of simulated objects in space, which makes it advisable to use them in automated layout problems. Voxel models allow you to create intelligent algorithms for automated placement. This eliminates the need to use multiple methods of search of accommodation options.


The unsolved problem of using voxel geometric models in the problems of automated layout is their integration into modern CAD systems, which limits their widespread introduction into the practice of computer-aided design.

Author details

Leonid V. Markin
Moscow Aviation Institute (National Research University), Moscow,
Russian Federation

*Address all correspondence to: markinl@list.ru

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Osipov VA. Mashinnye metody proektirovaniya nepreryvno karkasnyh poverhnostej [Machine Methods for Designing Continuous-Frame Surfaces]. Moscow: Mashinostroenie Publishers, 1979. 248 p
- [2] Markin LV. O putyakh sozdaniya geometricheskikh modelej avtomatizirovannoj komponovki [On the Ways of Creating Geometric Models of Automated Layout]. Geometriya i grafika [Geometry and Graphics]. 2015;3(1):64-69
- [3] Deniskin YI, Egorov EV, Nartova LG, Kuprikov MY. Prikladnaya Geometriya. Nauchnye Osnovaniya i Primenenie v Tekhnike [Applied Geometry. Scientific Grounds and Application in Technology]. Moscow: MAI Press Publishers; 2010. 385 p
- [4] Zozulevich DM. Mashinnaya Grafika v Avtomatizirovannom Proektirovanii [Computer Graphics in Computer Aided Design]. Moscow: Mashinostroenie Publishers; 1976. 240 p
- [5] Lin S. Razrabotka Metodov i Geometricheskikh Modelej Analiza Nezapolnennyh Prostranstv v Zadachah Razmeshcheniya [Development of methods and geometric models for the analysis of unfilled spaces in placement problems] [thesis]. Moscow Aviation Institute: Moscow; 2011
- [6] Dijkstra EW. A note on two problems in connexion with graphs. *Numerische Mathematik*. 1959;1:269-271
- [7] Khantanapoka K, Chinnasarn K. Pathfinding of 2D & 3D game real-time strategy with depth direction A* algorithm for multi-layer. In: Eighth International Symposium on Natural Language Processing, Thailand. 2009. pp. 184-188
- [8] Htun NN. Razrabotka i issledovanie receptornyh geometricheskikh modelej telesnoj trassirovki [Development and research of receptor geometric models of body tracing] [thesis]. Moscow: Moscow Aviation Institute; 2014
- [9] Khan KM. Matematicheskoe i programmnoe obespechenie rascheta zatenennosti solnechnyh batarej kosmicheskikh letatel'nyh apparatov [Mathematical and software for calculating the shading of solar panels spacecraft] [thesis]. Moscow: Moscow Aviation Institute; 2018
- [10] Khan KM, Markin LV, Tun EV, Korn GV. Receptornye Modeli v Zadachah Avtomatizirovannoj Komponovki Tekhniki [Receptor Models in the Problems of Automated Layout Technology]. Saarbrücken: Lambert Publishers; 2016. 110 p