UDC 004.946
DOI: 10.15587/1729-4061.2020.201694

# CONSTRUCTION OF A MATRIX DISCRETE MODEL OF A THREE-DIMENSIONAL BODY FOR THE RECONSTRUCTION OF ITS SHAPE

**O. Reuta**
PhD, Associate Professor
Department of Software of Computer Systems
Dnipro University of Technology
D. Yavornytskoho ave., 19,
Dnipro, Ukraine, 49005
E-mail: reutaov@gmail.com

**H. Hab Raman**
Postgraduate Student*
E-mail: hadihabrumman@gmail.com

**D. Mozgovoy**
PhD, Associate Professor*
E-mail: m-d-k@i.ua

*Department of Electronic Telecommunications
Oles Honchar Dnipro National University
Gagarina ave., 72, Dnipro, Ukraine, 49010

*Запропонована матрична модель подання просторових об'єктів для задач синтезу, реконструкції і аналізу їх форми. Модель будується на основі дискретних даних про об'єкт, якими, наприклад, являються растрові зображення або показання просторових сканерів. На відміну від подібних воксельных моделей, матричні моделі описують не об'єм, а поверхні об'єктів і, зберігаючи достоїнства воксельных моделей, такі як простота і регулярність структури, усувають властиву їм надмірність. У роботі показано, що зберігаючи інформацію про форму, достатню для візуалізації об'єкту, матрична модель може займати в 1.5–3 рази менший об'єм пам'яті порівняно з воксельною (порівняння проводилося для моделей у форматі VOX пакету MagicaVoxel). Встановлені умови, за яких матрична модель залишається економнішою за воксельну, і показано, що ці умови виконуються для практично значимих випадків.*

*Описаний алгоритм побудови дискретної матричної моделі на основі воксельної.*

*Запропонований загальний підхід до вирішення проблеми редискретизації моделей об'єктів об'ємної графіки, не залежний від розмірності масиву початкових даних. У рамках цього підходу реалізована процедура редискретизации матричної моделі. Описані необхідні перетворення матриць моделі, що включають як редискретизацію, так і реквантовання, що забезпечує їх контрольовану точність подання просторових об'єктів.*

*Для запропонованої матричної моделі також розроблені процедури контролю і відновлення цілісності. Отримані умови контролю цілісності моделі в практично значимих випадках (коли число елементів моделі більше, ніж $15^3$) дозволяють скоротити число елементів, що переглядаються, в порівнянні з воксельною моделлю.*

*Встановлені обмеження матричних моделей, пов'язані з можливою втратою інформації про частину поверхні, приховану від зовнішнього спостерігача*

*Ключові слова: воксельна модель, матрична модель, тривимірні об'єкти, реконструкція форми, редискретизація, усунення надмірності*

*Received date 27.02.2020*
*Accepted date 20.04.2020*
*Published date 30.04.2020*

## 1. Introduction

One of the most common ways of representing three-dimensional bodies in the problems of their synthesis and analysis using computer graphics is discrete models composed of voxels – volume elements in the form of parallelepipeds (most often cubes) or, much less commonly, balls [1]. As a rule, voxel material is homogeneous with fixed transparency. In some cases, the inclusion of voxels of varying degrees of transparency is also possible [2].

Voxel models (VM) are the basis of volumetric graphics and are used, as a rule, for the synthesis of virtual or approximations of real three-dimensional objects in the tasks of their visualization. They are widely used in medical diagnostic systems (computer, magnetic resonance, positron emission tomography), for creating virtual reality of various kinds of simulators, in 3D printers, in systems of modeling of spatial forms. VMs have a number of advantages compared to other models, primarily analytical ones. So, for example, it should be noted that they are independent of the complexity of the modeled objects and their surroundings, the regularity of the structure, the simplicity of processing, which is especially important for solving the problems of shape reconstruction [3, 4].

The relevance of this issue is determined by the fact that VMs contain a significant amount of redundant information that is not directly related to the description of the shape of a three-dimensional object, which is the main obstacle to creating effective procedures for working with them. And although VMs are actively used in systems where information about the volume of an object is significant, however, in applications where the main description of an object is its surface, they are largely redundant. Examples are applications that solve the problem of reconstructing the shape of a spatial object or 3D reconstruction (the «shape from X» task). The initial data in them, as a rule, are discrete data from different types of scanners.

## 2. Literature review and problem statement

A lot of publications are devoted to the reconstruction of forms of three-dimensional objects using voxel models. So, in [5], the results of a study on the development of algorithms and methods for solving the problem of designing complex forms that are generated using voxels are presented. The methods described in the article use the criteria of constructiveness and comfort, such as the proximity of other objects, compactness, size of the building and others. These methods are effective, but the authors do not take into account the processing time and the accuracy of the result.

Automatic refinement of unorganized point clouds obtained by scanning systems is considered in [6]. For a given point cloud, the proposed method first converts the input data to supervoxels using segmentation, and then builds a graph on the nodes of these voxels. Then an iterative resampling method is introduced to project points onto all potential surfaces with given constraints. The authors propose a complicated algorithm of several stages, although this could be done with a lighter algorithm in other ways.

In [7, 8], the issue of the synthesis of three-dimensional images intended for spatial visualization systems is solved. The author offers a simple and effective method that can be implemented in software.

In [9], computer technology for three-dimensional visualization of discrete voxel objects is described, which uses a computationally efficient method for generating lattices consisting of almost any type of cells that can be used for arbitrary external geometry. This article solves the problem of visualization using voxel graphics for modeling systems.

In [10], the authors proposes a semi-automatic procedure for converting a three-dimensional cloud of points of complex objects in three-dimensional models based on finite elements. The procedure considers the point cloud as a summation of the points. The complexity of the clouds is arbitrary, since the procedure is intended for ground-based scanning, which is applied to buildings with irregular geometry.

In [11], the concept of a «voxel group» is proposed: each voxel group consists of several voxels that belong to one real world object. Then, point cloud shapes in each voxel group are recognized, and this shape information is used to combine the voxel group. This article discusses the efficient extraction of LiDAR scan points using a laser scanning system on vehicles (VLS) belonging to different types of buildings in large-scale complex urban environments.

In [12], an algorithm is developed for detecting three-dimensional buildings based on voxel segmentation to separate the voxels of a building. The proposed algorithm first voxelizes a point cloud into a voxel structure in gray scale. In this case, the gray scale corresponds to the quantized average intensity of points in the cloud. A voxelized dataset is segmented into several 3D-connected regions depending on the coherence and similarity of the gray scale. The areas corresponding to the roof of the building and the facade are determined sequentially in accordance with such characteristics as their area, density, elevation and location. This article solved the problem of building detection based on voxel segmentation of onboard LiDAR data.

In [13], a new two-stage computational method is presented for finding the optimal types of surface coverage using voxels, medial objects (MO) and the random key genetic algorithm (RKGA). At the first stage, the proposed method increases the surface of buildings by expanding voxels to determine the subvolume around buildings. Then, the MOs of this subvolume are calculated and points are selected using a Gaussian sample around the MO surface. At the second stage, the optimization problem is formulated as a coverage problem and is solved by searching for points using RKGA and searching. This article discusses the problem of finding the optimal set of points that cover the entire surface of the target geometric model of the object.

Automating point cloud processing plays a key role in decision making systems. In [14], a model of three-dimensional objects based on voxels was proposed, which better characterizes point clusters and provides better performance of controlled or uncontrolled classification. This article solves the problem of providing a point cloud parsing module for extracting semantic clusters with voxel division of a data set.

In [15], a method based on voxel and probabilistic models for segmenting a point cloud is proposed, which is designed for automatic and adaptive separation of a three-dimensional scene. It represents the entire cloud of points first with the help of three-dimensional cubes, using a voxel structure based on octree trees. Then, the normal vector and the center of gravity of the points in each voxel are calculated as an attribute of the voxel. Geometric parameters (proximity, smoothness, closure, and continuity) are calculated based on the attributes of voxels. The authors solve the problem of segmentation of point clouds, which are designed for automatic and adaptive separation of a three-dimensional scene.

The methods and algorithms presented in the cited publications are quite effective and original, however, they do not address the problem of reducing the volume of voxel models in the computer's memory, which is important both for storage and for reducing processing time.

## 3. The aim and objectives of research

The aim of this research is building a model of a three-dimensional object based on matrices to describe its shape, which would eliminate the redundancy of the VM and save in the model only the information that directly describes the shape of the object.

To achieve the aim, the following objectives are set:

– give a description of the discrete matrix model of the spatial object and show its relationship with its voxel model;

– develop an algorithm for constructing a matrix model based on the transition from a set of voxels to a set of faces, or from a voxel model of a spatial body to its matrix model;

– develop procedures for the resampling of a discrete model of a three-dimensional body, taking into account the existing limitations;

– determine the effectiveness of the proposed matrix model in terms of the amount of required RAM.

## 4. The method of constructing a discrete matrix model of a spatial object

### 4. 1. Accepted simplifications and limitations

A three-dimensional body (spatial object) can be represented by its VM, the main disadvantage of which is the need to save and process a set of voxels of significant volume. The problem is compounded by the fact that of this set for forming tasks, only those voxels whose faces form the surface of the body are informative. Therefore, it will be logical to include only them in the body model. On the other hand, although the amount of information used to represent the shape of the body is reduced, due to the irregularity of the connections between the voxels that form the surface of the body, the structure of the data representing this information and, accordingly, the procedures for working with it are greatly complicated.

In the work, to reduce the redundancy of the body model, it includes only information about the external faces of the voxels that form its surface. In the future, the study will concern only such faces. To simplify the data structure intended for their storage, the matrix form of the description of relations will be used.

### 4. 2. The transition from a set of voxels to a set of faces in a discrete model

Let's start with an example (Fig. 1). Let's consider a three-dimensional object (body) in the form of a torus (Fig. 1, *a*) and its voxel model (Fig. 1, *b*).

The matrix model (MM) is formed on the basis of voxels, for which the notation shown in Fig. 1, *c*. Since all voxels are oriented in the same way, the axes of the local coordinate system of each individual voxel *oxyz* (Fig. 1, *c*) are parallel to the corresponding axes of the general coordinate system of the voxel model and the body *OXYZ* (Fig. 1, *d*).

To construct the MM from the BM of the body, let's define a dimensional parallelepiped covering the voxel model, whose faces *OXYZ* where $i=\overline{0,5}$, are oriented in the same way as the corresponding faces of the voxels Fi where $i=\overline{0,5}$, of the components of the voxel model are (Fig. 1, *d*). Moreover, $G_i \parallel F_i$, $i=\overline{0,5}$. The external faces of the voxels that make up the surface of the VM of the same type $F_i$ are separated from the faces of the same type of the parallelepiped $G_i$ by a distance equal to the whole number of linear dimensions of the voxel in the direction of the axis normal to $G_i$. In the future, this quantity will be considered the distance from the voxel face to the face of the overall parallelepiped.

For each of the outer faces of the voxels of the surface of a VM of type $F_i$, it is possible to construct an orthogonal projection onto the face $G_i$ of a dimensional parallelepiped along the normal to $G_i$. If the VM is convex, then the projections of such faces $F_i$ cover all or part of the face $G_i$ without intersection. If the VM is concave, then some projections may coincide, and in this case the face $F_i$ closest to $G_i$ remains in consideration. In any case, part of the $G_i$ face may remain uncovered by $F_i$ projections.

If a rectangular grid is applied to the $G_i$ face so that its cells coincide with the projections $F_i$ and a number is assigned to each cell, then it is possible to obtain a matrix that represents the relief of the VM in the direction of the normal of the $G_i$ face, the relief matrix.

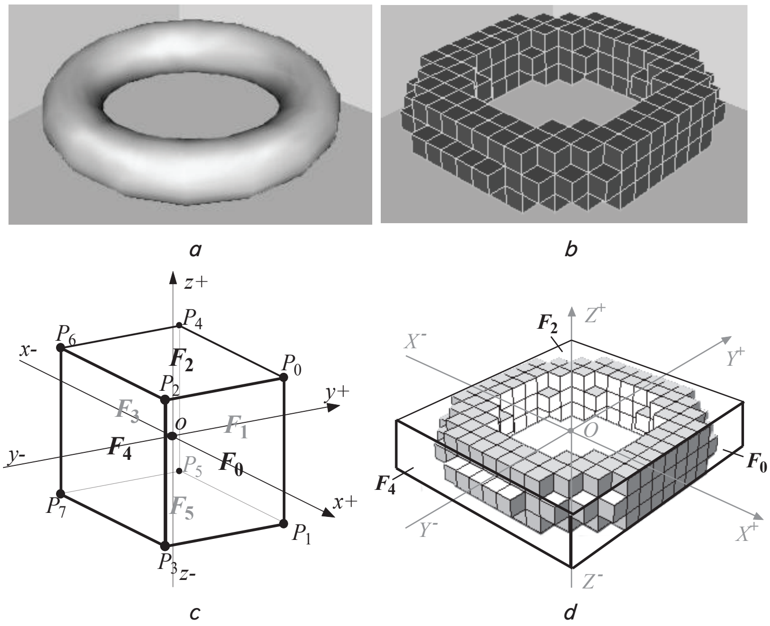

Fig. 1. Voxel model of a three-dimensional object: *a* — three-dimensional object in the form of a torus; *b* — voxel model of the torus; *c* — axis of the local coordinate system; *d* — coordinate system of the voxel model of the torus

### 4. 3. Definition of the matrix model as a set of relief matrices of the voxel model

To build a matrix model on the basis of this VM, let's associate each of the faces of the dimensional parallelepiped $G_i$ with a relief matrix, which elements are numbers equal to the distance from the corresponding $F_i$ to its projection onto $G_i$.

Moreover, if the projections of several $F_i$ coincide, then the value of the corresponding matrix element will be the distance from $G_i$ to the nearest face $F_i$. If no projection corresponds to the matrix element (grid cell on $G_i$), then its values will be considered equal to ∞ (infinity).

As an example, let's consider a relief matrix constructed for the face $G_0$ of the overall BM parallelepiped shown in Fig. 1, *d*:

$$M_0=\begin{pmatrix} \infty & 3 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & \infty \\ 4 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 4 \\ \infty & 3 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & \infty \end{pmatrix}.$$

If in the relief matrix let's replace elements which value is ∞ (infinity) by 0, and all the rest by 1, let's get a stencil matrix, which is often used in identifying the shapes of three-dimensional objects [4]. For example, for the case under consideration, the stencil matrix will look:

$$T_0=\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

A complete discrete matrix model (DMM) of a three-dimensional body is constructed on the basis of its VM and is a set of matrices $\mathbf{M}=\{\mathbf{M}_0, \mathbf{M}_1, \ldots \mathbf{M}_5\}$. Each of them is a relief matrix of the corresponding face of the dimensional parallelepiped $G_i$, where $i=\overline{0,5}$. Matrix determines the location of the outer faces of the voxels that make up the surface of the VM of the corresponding type $F_0$, $F_1$, ..., $F_5$, as well as three numbers that specify the linear dimensions of the dimensional parallelepiped (distances between $G_0$ and $G_3$, $G_1$ and $G_4$, $G_2$ and $G_5$).

These numbers are integers, because, as noted above, distances and sizes are determined in quantities of the corresponding linear dimensions of the voxel.

It should be noted that for the model presented in Fig. 1, *d*, $\mathbf{M}_0=\mathbf{M}_3$, $\mathbf{M}_1=\mathbf{M}_4$ and $\mathbf{M}_2=\mathbf{M}_5$, which is associated with the symmetry of the initial body and its BM (Fig. 1, *a*, *b*).

### 4. 4. Algorithm for constructing a discrete matrix model of a three-dimensional object

Using the above provisions, the algorithm for constructing a DMM can be formulated as a sequence of the following steps. The initial data for the algorithm is a voxel model of a spatial object, obtained on the basis of discretely presented data about it or directly synthesized.

*Step 1.* The overall container (box) of the model is determined by the linear dimensions of the original VM. It should be noted that the VM can also be limited to a bulk container (Bounding Box). However, if it exceeds the actual dimensions of the model, then its use for constructing a DMM can lead to a decrease in efficiency.

*Step 2.* The discreteness of the DMM corresponding to the discreteness of the VM is set. If in the future, the discreteness of the DMM will need to be changed, then for this it is possible to use the procedures of resampling described below.

*Step 3.* As described in paragraph 4. 3, each face of the overall container is associated with a relief matrix. Elements of this matrix are defined as distances from the corresponding face of the overall container to the nearest face of the nearest VM voxel at the location of the relief matrix element. As noted above, the value of the element can also be «infinity» if the VM does not contain the corresponding voxel.

*Step 4.* The resulting relief matrices together comprise the DMM of the original spatial body.

Let's note that in this paper let's consider a dimensional container in the form of a parallelepiped. However, this is not necessary. For objects with radial symmetry, it may be convenient to use a dimensional cylinder or sphere.

### 4. 5. Change in the dimension of a discrete matrix model

Further manipulations with the shape of the spatial object in the problems of its analysis and synthesis require a change in the dimension of the DMM, for example, using the wavelet transform [16]. Therefore, let's further consider the procedure of resampling the proposed matrix model.

Let's start with the one-dimensional case. Let the model space be the segment OA of length L, divided into $N$ equal parts (discrete). The sample length is $\Delta=L/N$.

Each discrete is assigned a number $g_t$, $t=\overline{1,N}$, which is an element of the model. The set of elements makes up a model that represents a certain one-dimensional image of $\Omega$ (in this case, a histogram). Model resampling consists of dividing the OA into another predetermined number of samples $\bar{N}$ so that the corresponding new image $\bar{\Omega}$ defined by the new set of digits $\bar{g}_i$, $i=\overline{1,\bar{N}}$, has a minimum MSE error, the expression for which has the form:

$$MSE=\frac{1}{L}\int\limits_{OA}\left[\bar{\Omega}(x)-\Omega(x)\right]^2\mathrm{d}x. \tag{1}$$

Taking into account expression (1), a discrete version of which is given in [2, 5], the following procedure for calculating the elements, $\bar{g}_i$, $i=\overline{1,\bar{N}}$ is proposed:

1. The index $i$ is determined by the «range of influence» $[p,\,p']$:

$$p=\left](i-1)/H_N\right[_{>} \text{ and } p'=\left]i/H_N\right[_{\geq}, \tag{2}$$

where $H_N=\bar{N}/N$; the function $\left]x\right[_{>}$ returns the smallest integer greater than $x$, and $\left]x\right[_{\geq}$ – the smallest integer greater than or equal to it.

2. Then the value of the corresponding element is calculated:

$$\overline{g_i}=p_1g_p+p_2\sum_{u=p+1}^{p'-1}g_u+p_3g_{p'} \text{ at } p\neq p'$$

and

$$\overline{g_i}=g_p, \text{ if } p=p', \tag{3}$$

where $p_1=pH_N-i+1$, $p_2=H_N$, $p_3=i-H_N(p'-1)$.

To extend the procedure to models that are represented by two- and three-dimensional arrays (two- and three-dimensional, respectively), it is proposed to use the Cartesian product of sets, the application of which in this case is illustrated by the following relations:

$$g_s\times g_t=g_{st},\quad \sum_s g_s\times g_t=\sum_s g_{st},$$

$$g_s\times\sum_t g_t=\sum_t g_{st},$$

$$\sum_s g_s\times\sum_t g_t=\sum_s\sum_t g_{st},$$

$$\sum_s g_s\times\sum_t g_t\times\sum_z g_z=\sum_s\sum_t\sum_z g_{stz}. \tag{4}$$

In this case, two- and three-dimensional models can be constructed according to one-dimensional models corresponding to perpendicular directions, such as:

$$\bar{g}_{ij}=\bar{g}_i\times\bar{g}_j \text{ (two-dimensional model)}$$

and

$$\bar{g}_{ijk}=\bar{g}_i\times\bar{g}_j\times\bar{g}_k \text{ (three-dimensional model)}. \tag{5}$$

This approach is not limited to the indicated dimensions and can naturally be applied to models of arbitrary dimension.

For example, for the relief matrix [17], taking into account (3)–(5):

$$\begin{aligned}\bar{g}_{ij}={}&P_1Q_1g_{pq}+P_1Q_2\sum_{v=q+1}^{q'-1}g_{pv}+\\&+P_1Q_3g_{pq'}+P_2Q_1\sum_{u=p+1}^{p'-1}g_{uq}+P_2Q_2\sum_{u=p+1}^{p'-1}\sum_{v=q+1}^{q'-1}g_{uv}+\\&+P_2Q_3\sum_{u=p+1}^{p'-1}g_{uq'}+P_3Q_1g_{p'q}+P_3Q_2\sum_{v=q+1}^{q-1}g_{p'v}+P_3Q_2g_{p'q'},\end{aligned} \tag{6}$$

where $q=\left](j-1)/H_M\right[_{>}$, $q'=\left]j/H_M\right[_{\geq}$; $H_M=\bar{M}/M$; $M$ and $\bar{M}$ are, respectively, the old and new number of samples in the direction of change of indices $q$, $q'$ $(q,q'=\overline{1,M})$, and $j$ $(j=\overline{1,\bar{M}})$, $Q_1=qH_M-j+1$; $Q_2=H_M$; $Q_3=j-H_M(q'-1)$.

For the three-dimensional case of a VM, an expression similar to (6) is not given because of its considerable volume (the expression has $3^3=27$ terms).

### 4. 6. Procedures for resampling the voxel model and the discrete matrix model

Next, let's consider the practical aspect of the application of the proposed approach, and, in particular, the procedures for the resampling of VMs and DMMs, as examples of three- and two-dimensional models of volumetric graphics, respectively.

Let's believe that the VM is represented by a three-dimensional array, the elements of which can be logical values that determine the presence or absence of the corresponding voxel, or real values from the range, [0, 1], which indicate the degree of transparency of the voxel. At the same time, 0 defines an absolutely transparent voxel (space does not contain matter), 1 – absolutely opaque or dense voxel (space is filled with substance, does not transmit rays). Intermediate values determine the fraction of radiation transmitted by matter in a given region of space [17, 18].

It should be noted that the first case can be reduced to the second by replacing the logical values «true» and «false» with integers 0 and 1, respectively.

To redistribute the VM, after determining the model elements $g_{ijk}$, $\bar{g}_{ijk}$ are calculated. Their values are real numbers from the range [0, 1]. The boundaries of the range indicate, respectively, the absence and presence of a voxel in a given position of the sampled model. An intermediate value defines a translucent voxel, which can be:

a) left in the model translucent, if allowed by the conditions of the problem;

b) removes from the model if the value is less than a predetermined threshold;

c) remains in the model as opaque if its density value is greater than the threshold.

Let's note that the visualization of VMs with translucent voxels is possible using the α channel. At the same time, the resulting image is markedly much more similar to a continuous analogue of VM than when removing such voxels from the model or replacing them with opaque ones.

The resampling of the DMM also provides for the necessity of carrying out a requantization procedure – changing those values of the elements of the relief matrices that directly may not have been subjected to resampling, but have been exposed to it [16, 19].

This is due to the fact that if the whole DMM is sampled, rather than a separate matrix, then due to a change in the model discreteness along a certain coordinate axis, the corresponding linear voxel size changes, which, in turn, leads to the need for recalculation of values (requantization) of matrix elements representing the faces of the model to which this coordinate axis is normal.

Thus, each resampling along a separate coordinate axis causes a resampling of the four matrixes of the model parallel to this axis and the requantization of two matrices normal to it.

As an example, Table 1 provides information on which transformations of relief matrices are caused by the resampling along one or another coordinate axis.

Table 1

Matrix transformation caused by model resampling

| Sampling direction | Variable dimension | Sampling of the same size | Sampling of two sizes | Requantization |
|---|---|---|---|---|
| *OX* | *N* | $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}_4$, $\mathbf{M}_5$ | – | $\mathbf{M}_0$, $\mathbf{M}_3$ |
| *OY* | *M* | $\mathbf{M}_0$, $\mathbf{M}_2$, $\mathbf{M}_3$, $\mathbf{M}_5$ | – | $\mathbf{M}_1$, $\mathbf{M}_4$ |
| *OZ* | *K* | $\mathbf{M}_0$, $\mathbf{M}_1$, $\mathbf{M}_3$, $\mathbf{M}_4$ | – | $\mathbf{M}_2$, $\mathbf{M}_5$ |
| *OX* and *OY* | *N*, *M* | $\mathbf{M}_0$, $\mathbf{M}_1$, $\mathbf{M}_3$, $\mathbf{M}_4$ | $\mathbf{M}_2$, $\mathbf{M}_5$ | $\mathbf{M}_0$, $\mathbf{M}_1$, $\mathbf{M}_3$, $\mathbf{M}_4$ |
| *OY* and *OZ* | *M*, *K* | $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}_4$, $\mathbf{M}_5$ | $\mathbf{M}_0$, $\mathbf{M}_3$ | $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}_4$, $\mathbf{M}_5$ |
| *OZ* and *OX* | *K*, *N* | $\mathbf{M}_0$, $\mathbf{M}_2$, $\mathbf{M}_3$, $\mathbf{M}_5$ | $\mathbf{M}_1$, $\mathbf{M}_4$ | $\mathbf{M}_0$, $\mathbf{M}_2$, $\mathbf{M}_3$, $\mathbf{M}_5$ |
| *OX*, *OY* and *OZ* | *N*, *M*, *K* | – | all | all |

In the general case, if the resampling is carried out along the *i* axis (*i* takes values 0, 1 and 2, denoting, respectively, the *OX, OY* and *OZ* axes), then one-size resampling should be performed for all matrices, except for the matrices $\mathbf{M}_i$, $\mathbf{M}_{i+3}$, and requantization of the matrices $\mathbf{M}_i$, $\mathbf{M}_{i+3}$.

If the resampling is carried out along the two axes *i* and *j* themselves, then the sampling of the same size and quantization should be performed for all matrices, except for $\mathbf{M}_{3-i-j}$ and $\mathbf{M}_{6-i-j}$, for which it is necessary to perform the resampling of two sizes. If the resampling is carried out along three axes, then for all six matrices of the discrete matrix model, one should perform the resampling of two sizes and the requantization.

The requantization procedure is as follows. After calculating the new value of the element $\bar{g}_{ij}$ of the relief matrix according to formula (6), it should be adjusted in accordance with the new value of the linear size of the voxels in the direction normal to the face of the dimensional parallelepiped, which is represented by the relief matrix $M_0$.

In this case, the new value of the element is calculated as:

$$\hat{g}_{ij} = \bar{g}_{ij}\,\bar{M}/M = \bar{g}_{ij}H_M, \tag{7}$$

where $M$ and $\bar{M}$ – the old and new number of samples in the direction of the normal to the face of the dimensional parallelepiped, which is the matrix being subjected to requantization.

As a result, the general procedure for the DMM complete resampling, taking into account the necessary requantization, consists of the following steps:

1) according to expression (6), all relevant relief matrices are rediscovered. The order of their resampling is arbitrary;

2) using Table 1 and the relations of the general case considered above, matrices are determined that should be quantized;

3) matrices are recounted in an arbitrary order;

4) the obtained real values of the elements of the matrices approach the integers, taking into account the values representing «infinity».

The proposed general approach to solving the problem of the resampling of models of three-dimensional graphic objects expands the possibilities of using the latter due to their effective adaptation to the conditions of a particular problem, which is confirmed by the considered examples of its application to transformations of a voxel model and a discrete matrix model.

### 4. 7. Restoring the integrity of the discrete matrix model

The next problem that should be solved is the restoration of the integrity of the DMM in the case when it is synthesized artificially, and not according to the existing VM. In this case, a situation may arise when the synthesized DMM can't be associated with one VM.

As a result, it is necessary to determine the conditions that a holistic model of a three-dimensional object satisfies, presented in the form of a VM or a DMM, and does not satisfy a model that has lost its integrity. In addition, based on such conditions, procedures should be developed to restore the lost integrity of the model.

Performing the first part of the task, let's note that the integrity of the model is ensured by the preservation of certain relationships between the elements of the relief matrixes that make up the DMM [17, 20].

Moreover, these relations are established between pairs of matrices (more precisely, between their elements) and, since a DMM consists of 6 relief matrices, the number of conditions is equal to the number of combinations of 6 by 2 or a binomial coefficient $\mathbf{C}_6^2=15$.

Below are these conditions for matrix elements whose values are finite. Fig. 2 illustrates the relationships between the elements of the relief matrices that make up the DMM.

$$m_1\left[i, m_0[i,j]+1\right]\le M-j,\quad i=\overline{1,K},\quad j=\overline{1,M};$$

$$m_0\left[m_2[i,j]+1, M-i+1\right]\le N-j,\quad i=\overline{1,M},\quad j=\overline{1,N};$$

$$m_0[i,j]\le N-m_3[i,M-j+1]-1,\quad i=\overline{1,K},\quad j=\overline{1,M};$$

$$m_4\left[i, N-m_0[i,j]\right]\le j,\quad i=\overline{1,K},\quad j=\overline{1,M};$$

$$m_0\left[K-m_5[i,j], M-i+1\right]\le j,\quad i=\overline{1,M},\quad j=\overline{1,N};$$

$$m_2\left[m_1[i,j]+1, N-j+1\right]\le j,\quad i=\overline{1,N},\quad j=\overline{1,K};$$

$$m_3\left[i, m_1[i,j]+1\right]\le N-j,\quad i=\overline{1,K},\quad j=\overline{1,N};$$

$$m_1[i,j]\le M-m_4[i,N-j+1]-1,\quad i=\overline{1,K},\quad j=\overline{1,N};$$

$$m_5\left[i, m_1[i,j]+1\right]\le K-j,\quad i=\overline{1,N},\quad j=\overline{1,K};$$

$$m_3\left[m_2[i,j]+1, i\right]\le j,\quad i=\overline{1,M},\quad j=\overline{1,N};$$

$$m_4\left[m_2[i,j]+1, j\right]\le M-i,\quad i=\overline{1,M},\quad j=\overline{1,N};$$

$$m_2[i,j]\le K-m_5[i,N-j+1]-1,\quad i=\overline{1,M},\quad j=\overline{1,N};$$

$$m_4\left[i, m_3[i,j]+1\right]\le M-j,\quad i=\overline{1,K},\quad j=\overline{1,M};$$

$$m_5\left[j, N-m_3[i,j]\right]\le K-i,\quad i=\overline{1,K},\quad j=\overline{1,M};$$

$$m_5\left[M-m_4[i,j], N-j+1\right]\le K-i,\quad i=\overline{1,K},\quad j=\overline{1,N},\qquad (8)$$

where $m_q[i,j]$ – the matrix element $\mathbf{M}_q$, $q=\overline{1,6}$; $N$, $M$ and $K$ – the dimensions of the model along the coordinate axes $OX$, $OY$, and $OZ$, respectively.

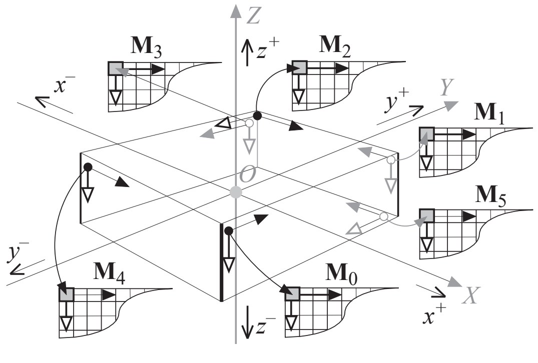

Fig. 2. The relationship between the elements of the relief matrix of the discrete matrix model

If the element of the relief matrix takes on an infinite value, then conditions (8) change and take the following form:

– if $m_0[i,j]=\infty$, then

$$m_1[i,k]\ne M-j,\quad m_3[i,M-j+1]=\infty,$$

$$m_4[i,k]\ne j-1,\quad m_5[M-j+1,k]\ne K-i,\qquad (9)$$

where $i=\overline{1,K}$, $j=\overline{1,M}$, $k=\overline{1,N}$;

– if $m_1[i,j]=\infty$, then

$$m_0[i,k]\ne j-1,\quad m_2[k,N-j+1]\ne i-1,$$

$$m_3[i,k]\ne N-j,\quad m_4[i,N-j+1]=\infty,$$

$$m_5[k,j]\ne K-i,\qquad (10)$$

where $i=\overline{1,K}$, $j=\overline{1,N}$, $k=\overline{1,M}$;

– if $m_2[i,j]=\infty$, then

$$m_0[k,M-i+1]\ne N-j,\quad m_1[k,N-j]\ne i-1,$$

$$m_3[k,i]=j-1,\quad m_4[k,j]\ne M-i,\quad m_5[i,N-j+1]=\infty,\qquad (11)$$

where $i=\overline{1,M}$, $j=\overline{1,N}$, $k=\overline{1,K}$.

Let's note that, due to the symmetry of the model (Fig. 2), condition (9) by the replacement $j'=M-j+1$ extends to the case $m_3[i,j']=\infty$, and conditions (10) and (11) by the replacement $j'=N-j+1$ by the cases $m_4[i,j']=\infty$ and $m_5[i,j']=\infty$, respectively.

To check the conditions (8), one should examine the following number of elements of the relief matrices:

$$E_{NMK}=5(NM+MK+KN)$$

or

$$E_R=15R^2,\ \text{if}\ N=M=K=R,\qquad (12)$$

and to check the conditions (9)–(11):

$$Q_{NMK}=NM+MK+KN+Kn_K+Nn_N+Mn_M.$$

or

$$Q_R=R(R+3n),\ \text{if}\ N=M=K=R$$

and

$$n_N=n_M=n_K=n.\qquad (13)$$

where $n_N$, $n_M$, $n_K$ – the number of elements having the value «infinity» in the matrices $\mathbf{M}_0$ ($\mathbf{M}_3$), $\mathbf{M}_1$ ($\mathbf{M}_4$) and $\mathbf{M}_2$ ($\mathbf{M}_5$), respectively (Fig. 2).

Let's note that in order to verify the integrity of the DMM by comparing it with the existing VM, it is necessary to examine the $N\times M\times K$ elements of the model, at $N=M=K=R$ is $R^3$, therefore, with $R>15$ or with the total number of model elements greater than $R>15$ (which, as a rule, takes place in practice [17, 21]) exceeds the value of (12).

This once again proves the effectiveness of using precisely DMM for analyzing the shape of three-dimensional objects in volumetric graphics tasks.

Turning to the second part of the problem, let's note that conditions (8) also provide the necessary basis for determining the procedure for restoring the integrity of the model.

Since the integrity is restored when inequalities (8) are fulfilled, then for each of them to be fulfilled if it is violated and the original model (Fig. 3, *a*) has lost its integrity (Fig. 3, *b*), it is necessary to change the model volume.

There are two ways:

– increase the corresponding values of the matrix elements until the violated integrity condition is satisfied and, thereby, reduce the volume of the model;

– reduce them, increasing the volume of the model.

As a result, let's have two approaches, one of which ensures restoration of the integrity of the model by increasing its total volume (Fig. 3, *c*), and the second – by reducing (Fig. 3, *d*). The decision on the appropriateness of each of them is taken in each case separately, depending on the conditions of a particular task.

To formalize the procedure for restoring the integrity of the model, let's consider as an example the first inequality of conditions (8), namely:

$$m_1\left[i, m_0\left[i, j\right]+1\right] \le M - j, \quad i=\overline{1,K}, \quad j=\overline{1,M} \tag{14}$$

and let's assume that it is its violation that caused the loss of model integrity.

If the approach to restoring integrity by adopting the model volume is adopted when voxels are attached to it in a doubtful case (Fig. 3, *c*), then the relief matrix $\mathbf{M}_1$ of the element $m_1\left[i, m_0\left[i, j\right]+1\right]$, in which integrity condition (14) is violated, should be assigned the value **M**–*j*. In fact, this means adding voxels to the model $\bar{m}_1 - M + j$, where $\bar{m}_1$ – the old value of the element.

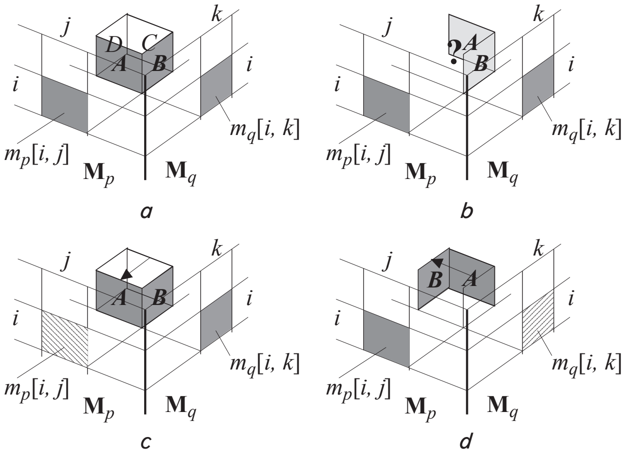

Fig. 3. Options for restoring the integrity of the model: *a* – original model, *b* – example of loss of integrity; *c* – restoration of the integrity of the model by increasing its volume; *d* – restoration of the integrity of the model by reducing its volume

If the approach is taken to restore integrity by reducing the model volume when possible voxels are removed from it (Fig. 3, *d*), then in condition (14) the element $m_0\left[i, j\right]$ of the relief matrix $\mathbf{M}_0$ should be increased by 1 until the condition is satisfied or until it assumes the value *N*–1 (whichever comes first).

In the latter case, if condition (14) is not fulfilled, the element $m_0\left[i, j\right]$ should be assigned the value «infinity». Moreover, $N-1-\bar{m}_0$ doubtful voxels will be removed from the model, where $\bar{m}_0$ – the old value of the element

Similarly, procedures can be defined to restore the integrity of the model in violation of other inequalities (8).

**4. 8. Comparison of volumes of voxel and discrete matrix models of a three-dimensional body**

Let's compare the voxel and discrete matrix models of a three-dimensional body in terms of their volumes and determine the conditions for the effective use of each of them. Given that the models have the same information content for the reconstruction of the shape of an object [22], the criterion for the effectiveness of the model in this study is the saving of computer RAM.

The amount of memory in bits needed to store each of the models is defined as:

$$V = V_{el} \cdot N_{el}, \tag{15}$$

where $V_{el}$ – the volume of one element, and $N_{el}$ – the number of elements in the model.

For VMs, in the simplest case, the volume of one model element is equal to one bit ($V_{el}$=1 bit), the value of which is «1» if the model element defines the volume element that belongs to the body, and «0» otherwise.

The number of VM elements is determined by its resolution or dimension *N* and is $N_{el} = N^3$ (for now, let's assume that the VM's overall container is a cube). Thus, the memory capacity (15) for the VM will be:

$$V_{VM}=1\cdot N^3, \text{ bit.} \tag{16}$$

DMM, built on the basis of the above VM, includes 6 matrices in size $N \times N = N^2$ and, therefore, consists of $N_{el} = 6 \cdot N^2$ elements. Each element must supply a value lying in the range 0…*N*, where the value *N* symbolizes «infinity».

Then the amount of memory designed to store such an element will be a $V_{el} = \log_2\left(N+1\right)$ bit, and the total amount (15) of the discrete matrix model is:

$$V_{DMM} = 6 \cdot N^2 \log_2\left(N+1\right), \text{ bit.} \tag{17}$$

To compare relations (16) and (17), let's construct graphs of the functions $V_{VM}\left(N\right)$ and $V_{DMM}\left(N\right)$ – the gray and black curves in Fig. 4, respectively.

At the same time, it becomes obvious that with small dimensions of the models, the VM has advantages from the point of view of saving the RAM of a computer, and with large dimensions it has a DMM. To clarify this statement, let's solve the equation:

$$V_{VM} = V_{DMM} \text{ or } N^3 = 6 \cdot N^2 \log_2\left(N+1\right). \tag{18}$$

The exact solution to this equation is the expression:

$$N = 1 - \frac{6 \cdot W\left(-\frac{1}{12}\ln\left(2\right) \cdot 2^{5/6}\right)}{\ln\left(2\right)}, \tag{19}$$

where $W\left(x\right)$ – the Lambert function [23].

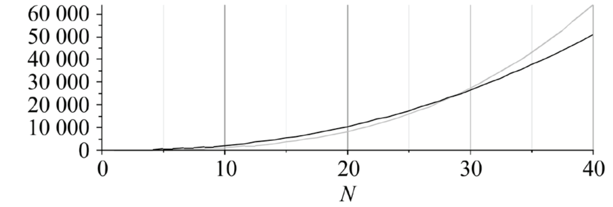

Fig. 4. Graphs of functions $V_{VM}(N)$ (gray curve) and $V_{DMM}(N)$ (black curve) according to (18)

Restricting ourselves to the actual values of the function $W(x)$ in calculating (19), let's obtain that $N \approx 29{,}61778929$. Thus, the condition under which the advantages of the DMM are manifested is $N \geq 30$.

Otherwise, VM is more economical.

It should be noted that the representation of an individual VM element in one bit is possible [23], but in real conditions it is associated with a significant complication of the corresponding procedures for working with the model.

The smallest addressable unit of memory that is efficiently processed in a computer is a byte (8 bits). Therefore, in existing computing systems, VM elements are represented by bytes, the value of which is modeled by the logical concepts of «true» and «false», respectively, the presence and absence of a voxel within the VM geometric representation.

As for the DMM, the byte that represents the matrix element has a value in the range $0 \ldots 255$ and, therefore, is suitable for storing models with $N \leq 255$. Therefore, if the storage unit is a byte, equation (18) takes the form $N^3 = 6 \cdot N^2$ and the efficiency of the DMM begins to manifest itself at $N > 6$. The condition that the DMM is more effective will have the form

In the general case, when an element of a discrete matrix model is represented by $m$ bits (usually $m = 2^k$, where $k = 3, 4, 5, 6 \ldots$), the condition under which the use of a discrete matrix model is more efficient from the point of view of saving computer RAM than a VM takes the form:

$$\frac{3}{4} m < N < 2^m - 1, \tag{20}$$

and the condition under which the use of VM is effective, respectively, is:

$$N < \frac{3}{4} m. \tag{21}$$

It should be noted that with $N = (3/4) \cdot m$ the model they occupy the same areas of RAM in the computer.

To illustrate the relations (20) and (21) in the Table 2 shows the obtained conditions calculated for various (typical) values $m$.

Table 2

Efficiency conditions for using models in terms of saving computer RAM

| Values | VM usage efficiency condition | DMM $m$ usage efficiency condition |
|---|---|---|
| 1 | $N < 30$ | $N \geq 30$ |
| 8 | $N < 6$ | $6 < N \leq 255$ |
| 16 | $N < 12$ | $12 < N \leq 65535$ |
| 32 | $N < 24$ | $24 < N \leq 4294967295$ |
| 64 | $N < 48$ | $48 < N \leq 18446744073709551615$ |

The use of a discrete matrix model under condition (20) leads to a decrease in the required random access memory of the computer in $(4/3) \cdot (N/m)$. And if let's take into account that in some problems of reconstructing the shape of a three-dimensional object, such as, for example, for the purpose of visualizing it or analyzing its shadow formation, it is planned to study no more than three faces [24] and, accordingly, three DMM matrices, while the amount of necessary information decreases three times more and the memory savings will be $(4/9) \cdot (N/m)$ times.

In the case when the geometric representation of the VM has different sizes along different coordinate axes (as, for example, in Fig. 2), equation (18) will have the form:

$$N_x N_y N_z = 2\left(N_x N_y + N_y N_z + N_z N_x\right) \times$$
$$\times \log_2\left(\max\left\{N_x, N_y, N_z\right\} + 1\right), \tag{22}$$

where $N_x$, $N_y$ and $N_z$ – respectively, the resolving power of the VM along the coordinate axes *OX*, *OY*, and *OZ* (Fig. 1), and the condition under which the DMM is more effective from the point of view of saving the computer RAM is:

$$\frac{m}{4} < \frac{N_x N_y N_z}{N_x N_y + N_y N_z + N_z N_x}. \tag{23}$$

Moreover, the discrete matrix model will be more economical than the voxel model in:

$$\frac{4 N_x N_y N_z}{m\left(N_x N_y + N_y N_z + N_z N_x\right)} \text{ times.}$$

For example, for a body which VM is shown in Fig. 2, let's have that $N_x = N_y = 15$, and $N_z = 3$, therefore, a discrete matrix model of such a body will be more economical than its VM if $m < 8{,}571428571$.

This means that when using cells of 1 byte size ($m$=8) to save VM and DMM elements in the computer's RAM, the latter will occupy 1.07 times (7 %) less memory.

Let's note that many existing spatial body representation formats using VMs retain only the minimum number of voxels needed. As a result, the number of actually stored model elements will be less than $N_x \times N_y \times N_z$, as is assumed in (22). As an illustration, consider the VOX format used to represent voxel models in the MagicaVoxel application [25], and compare the VM representation used in it with the corresponding DMM.

The VOX format involves the description of one or more models and additional information about the structure of the file itself and the number of models stored in it, the color palette and the parameters of the materials used [26]. A separate voxel model is characterized by the number of voxels *N* and a list of *N* elements, each of which is specified by four 16-bit numbers corresponding to the coordinates of the voxel ($x$, $y$, $z$) and its color index in the color palette. Fig. 5 shows three models included in the standard package of MagicaVoxel, and in table. Table 3 presents the results of comparing the number of elements for each model in its standard voxel representation in the VOX file and in the DMM representation. In this case, when calculating the amount of memory for each element, it was taken into account that the VM uses 3 bytes per element (excluding the color index), and the DMM uses 1 byte. Also, for the model shown in Fig. 5, *a* (chr_knight), the overall container used was shown in the figure with a white outline. For the remaining models, the VM and DMM containers coincide, because in these cases the container selected for the VM corresponds to its definition for DMM.

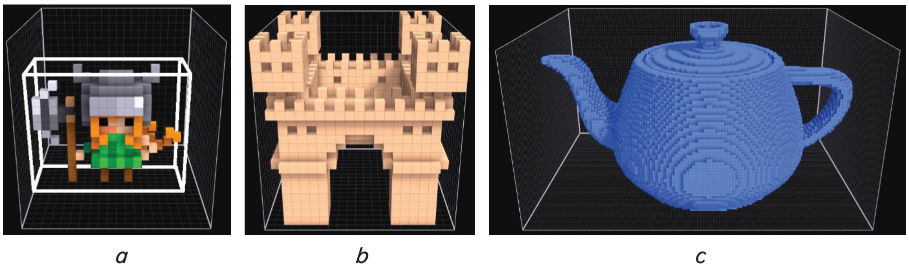

Fig. 5. Standard models of the MagicaVoxel package: *a* – chr_knight (the overall container for DMM is highlighted in white); *b* – castle; *c* – teapot

Table 3

Comparison of the representation of three models (Fig. 5) in the VM and DMM formats

| Model | Overall container | Amount of elements | | The total amount of memory occupied by the elements, bytes | | The ratio of VM volume to the DMM volume |
|---|---|---|---|---|---|---|
| | | DMM | VM | DMM | VM | |
| chr_knight | 17×8×14 | 972 | 516 | 972 | 1548 | 1.59 |
| castle | 21×21×21 | 2646 | 2745 | 2646 | 8235 | 3.11 |
| teapot | 126×80×61 | 45292 | 28528 | 45292 | 85584 | 1.89 |

Table 3 analysis shows that the smaller the number of VM voxels in the used overall container, the lower the DMM efficiency. As a rule, the number of VM elements is less than that of the DMM (in case the VM includes only the minimum necessary number of voxels, as in the VOX format). However, due to the fact that each element of the VM is represented by three coordinates, the volume of the VM is larger. In addition, it is possible to further reduce the DMM volume (with the complication of the processing of a file storing the model) by using the minimum required number of bits to represent a single element, according to Table 2. So, for example, for the chr_knight and castle models (Fig. 5, *a*, *b*), it is possible to use 5 bits instead of 8, while for the teapot model (Fig. 5, *c*) – 7.

## 5. Discussion of the proposed methods and models

The result of the work is the constructed matrix model, which, like the existing voxel models, based on the discreteness of the initial data on the object and having their simplicity, unlike the latter, takes into account only the surface shape of the spatial body, and thereby eliminates the redundancy associated with the need to store information about the volume of the object.

This is due to the fact that the DMM determines the shape of the object, describing its surface by the set of distances to the overall container (Fig. 1, *d*) and replaces the three-dimensional array of voxels with six relief matrices. Due to this, a discrete representation of the surface of the object is preserved, but the total volume of model elements is reduced. The conditions for such a reduction (20) and (23) are given for specific values of the bitness of model elements in Table 2 and demonstrated on specific models (Table 3).

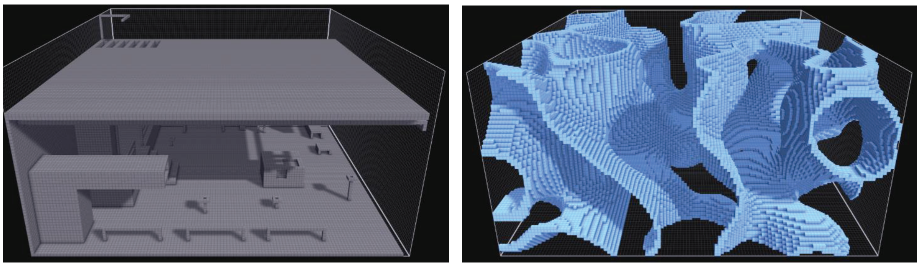

Fig. 6. Forms inaccessible to description with DMM

In order to adapt the model to specific sets of input data, a generalized procedure for resampling (3), (4) of the initial array has been developed, which does not depend on its dimension. It is shown how it is used for the two-dimensional case of DMM relief matrices (6). The fact of the effect of resampling of relief matrices on their values (Table 1) is established, which requires their corresponding requantization.

Manipulations with models can lead to violation of their integrity. To control the integrity of the DMM, in violation of which the restoration of the shape of the object becomes impossible, the integrity conditions (8) and (11) are obtained. It is shown that the use of these conditions in practically significant cases (when the number of model elements is more than 15) requires viewing fewer data elements than a direct comparison of DMM with VM.

Comparison of the proposed matrix model for representing the same spatial features (Fig. 5) as voxel models in the VOX format used in the MagicaVoxel package confirms their efficiency (Table 3). Depending on the specific form, a DMM requires a memory capacity that is 1.5–3 times smaller than for a VM.

The limitations of the proposed model include the lack of transparency, which is overcome in voxel models by using voxels of different optical densities. This issue requires further study.

Also a natural limitation of the proposed model is the possible loss of information about part of the surface of a spatial object, if it has more than one projection on the verge of a dimensional container (Fig. 6). However, this restriction is not critical in the applications that the DMM is oriented to, namely, where the part of the object's surface visible to the external observer is used.

In addition, at the moment, there is no procedure for visualizing spatial forms directly on the basis of DMM. For this, the transition to a VM is used, followed by the application of specific imaging methods (such as the Marching Cubes method and its modifications, which are widely used in computed tomography).

As a result, one of the directions of future research is finding an effective transformation of the matrix model into a description of the form based on the graphic primitives of modern modeling systems (e. g., Blender) or visualization libraries (e. g., OpenGL).

In addition, studies require the prospect of using the proposed DMM in applications using surface analysis of objects for their reconstruction and identification.

## 6. Conclusions

1. A matrix model is proposed that describes the surface of a spatial object in the form of six relief matrices and possesses the advantages of existing voxel models. Moreover, it eliminates the inherent redundancy of VM associated with the need to store information about the volume, which is not used in the problems of analysis and reconstruction of the surface of an object, i. e., its shape.

2. An algorithm for constructing a matrix model based on the existing voxel model is presented. It consists in determining the overall container covering the VM, and the subsequent assignment of six relief matrices.

3. A generalized method is described for determining the procedures of the resampling, which does not depend on the dimension of the source data array, which allows one to determine the procedures of the resampling of arrays of arbitrary dimension. Using it, a resampling procedure was obtained for the relief matrices of the DMM.

A relationship has been established between the processes of resampling the matrix model and its corresponding voxel model. It is shown that in the case of the matrix model, resampling is reduced to transformations of relief matrices, which reduces the total amount of calculations.

The general case of violation of the integrity of the matrix model is considered, which leads to the ambiguity of the description of the form. Two options for its recovery are proposed. One is associated with an increase in the volume of the represented object, and the second is associated with its decrease. At the same time, the amount of data used by the model itself remains unchanged. Both options equally restore the integrity of the model and the choice of one of them is determined only by preferences regarding the change in the volume of the object – in the direction of its increase or decrease.

4. The effectiveness of the proposed matrix model is compared with the corresponding voxel model in terms of the amount of required RAM. It is shown that the condition for the efficiency of the matrix model is $N \geq 30$, where $N$ – the model dimension. It is noted that this condition is fulfilled for the vast majority of practically significant cases.

## References

1. Li, Dzh., Uer, B. (2002). Trehmernaya grafika i animatsiya. Moscow: «Vil'yams», 640.
2. Reuta, O. V. (2008). Vyznachennia tini vokselia zahalnoho polozhennia u zadachakh modeliuvannia tryvymirnykh obiektiv. Pratsi Tavriiskoho derzhavnoho ahrotekhnichnoho universytetu. Prykladna heometriia ta inzhenerna hrafika, 41 (4), 44–49.
3. Musa, S., Ziatdiniv, R., Griffiths, C. (2013). Introduction to computer animation and its possible educational applications. New Challenges in Education. Retrospection of history of education to the future in the interdisciplinary dialogue among didactics of various school subjects. Ružomberok, Slovakia: VERBUM – vydavateľstvo Katolíckej univerzity v Ružomberku, 177–205.
4. Reuta, O. V. (2009). Matrychna dyskretna model tryvymirnoho tila dlia zadachi analizu yoho tineutvorennia. Heometrychne ta kompiuterne modeliuvannia, 25, 68–72.
5. Darmon, I. Voxel computational morphogenesis in urban context: proposition and analysis of rules-based generative algorithms considering solar access. Available at: https://www.researchgate.net/publication/329738763_Voxel_computational_morphogenesis_in_urban_context_proposition_and_analysis_of_rules-based_generative_algorithms_considering_solar_access
6. Li, M., Sun, C. (2018). Refinement of LiDAR point clouds using a super voxel based approach. ISPRS Journal of Photogrammetry and Remote Sensing, 143, 213–221. doi: https://doi.org/10.1016/j.isprsjprs.2018.03.010
7. Al-Oraiqat, A. M., Bashkov, E. A., Zori, S. A., Amro, A. M. (2017). Generalized 3D Voxel Image Synthesis Architecture for Volumetric Spatial Visualization. IPASJ International Journal of Information Technology (IIJIT), 5 (1). Available at: https://www.researchgate.net/publication/313397537_Generalized_3D_Voxel_Image_Synthesis_Architecture_for_Volumetric_Spatial_Visualization
8. Aremu, A. O., Brennan-Craddock, J. P. J., Panesar, A., Ashcroft, I. A., Hague, R. J. M., Wildman, R. D., Tuck, C. (2017). A voxel-based method of constructing and skinning conformal and functionally graded lattice structures suitable for additive manufacturing. Additive Manufacturing, 13, 1–13. doi: https://doi.org/10.1016/j.addma.2016.10.006
9. Vitiska, N. I., Gulyaev, N. A. (2015). An approach to visualization of three-dimensional scenes and objects via voxel graphics for simulation systems. Izvestiya Yuzhnogo federal'nogo universiteta. Tehnicheskie nauki. Available at: https://cyberleninka.ru/article/n/metod-vizualizatsii-tryohmernyh-stsen-i-obektov-vokselnoy-grafiki-dlya-sistem-imitatsionnogo-modelirovaniya
10. Castellazzi, G., D'Altri, A., Bitelli, G., Selvaggi, I., Lambertini, A. (2015). From Laser Scanning to Finite Element Analysis of Complex Buildings by Using a Semi-Automatic Procedure. Sensors, 15 (8), 18360–18380. doi: https://doi.org/10.3390/s150818360
11. Wang, Y., Cheng, L., Chen, Y., Wu, Y., Li, M. (2016). Building Point Detection from Vehicle-Borne LiDAR Data Based on Voxel Group and Horizontal Hollow Analysis. Remote Sensing, 8 (5), 419. doi: https://doi.org/10.3390/rs8050419
12. Wang, L., Xu, Y., Li, Y., Zhao, Y. (2018). Voxel segmentation-based 3D building detection algorithm for airborne LIDAR data. PLOS ONE, 13 (12), e0208996. doi: https://doi.org/10.1371/journal.pone.0208996
13. Jing, W., Shimada, K. (2017). Model-based view planning for building inspection and surveillance using voxel dilation, Medial Objects, and Random-Key Genetic Algorithm. Journal of Computational Design and Engineering, 5 (3), 337–347. doi: https://doi.org/10.1016/j.jcde.2017.11.013

14. Poux, F., Billen, R. (2019). Voxel-based 3D Point Cloud Semantic Segmentation: Unsupervised Geometric and Relationship Featuring vs Deep Learning Methods. ISPRS International Journal of Geo-Information, 8 (5), 213. doi: https://doi.org/10.3390/ijgi8050213
15. Xu, Y., Tuttas, S., Hoegner, L., Stilla, U. (2018). Voxel-based segmentation of 3D point clouds from construction sites using a probabilistic connectivity model. Pattern Recognition Letters, 102, 67–74. doi: https://doi.org/10.1016/j.patrec.2017.12.016
16. Reuta, O. V. (2010). Redyskretyzatsiya modeli tryvymirnoho tila dlia zadachi analizu yoho tineutvorennia. Pratsi Tavriyskoho derzhavnoho ahrotekhnichnoho universytetu. Prykladna heometriya ta inzhenerna hrafika, 41 (4), 44–49.
17. Reuta, O. V. (2008). Vokselna model tryvymirnoho obiektu v zadachakh rekonstruktsiyi yoho formy. Prykladna heometriia ta inzhenerna hrafika, 80, 500–504.
18. Pyysalo, U., Sarjakoski, T. (2008). Voxel approach to landscape modeling. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXVII, 563–568.
19. Hawick, K. A. (2010). 3D Visualisation of Simulation Model Voxel Hyperbricks and the Cubes Program. Available at: http://cssg.massey.ac.nz/cstn/082/cstn-082.pdf
20. Liu, Z., Tang, H., Lin, Y., Han, S. (2019). Point-Voxel CNN for Efficient 3D Deep Learning. 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Vancouver.
21. Poux, F., Billen, R. (2019). Voxel-based 3D Point Cloud Semantic Segmentation: Unsupervised Geometric and Relationship Featuring vs Deep Learning Methods. ISPRS International Journal of Geo-Information, 8 (5), 213. doi: https://doi.org/10.3390/ijgi8050213
22. Milian, F. M., Garcia, F., Guevara, M. V. M., Da Silva Correia I. K., Meira, N. G. (2009). Development of voxel models from 3D surfaces. Optimized methodology. XIV Congresso Brasileiro de Física Médica.
23. Shin, D., Fowlkes, C. C., Hoiem, D. (2018). Pixels, Voxels, and Views: A Study of Shape Representations for Single View 3D Object Shape Prediction. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. doi: https://doi.org/10.1109/cvpr.2018.00323
24. Knyaz, V. A., Kniaz, V. V., Remondino, F. (2019). Image-to-Voxel Model Translation with Conditional Adversarial Networks. Computer Vision – ECCV 2018 Workshops, 601–618. doi: https://doi.org/10.1007/978-3-030-11009-3_37
25. MagicaVoxel @ ephtracy (Win/Mac). Available at: https://ephtracy.github.io/
26. MagicaVoxel-file-format-vox.txt. Available at: https://github.com/ephtracy/voxel-model/blob/master/MagicaVoxel-file-format-vox.txt