**International Journal of Mathematics and Computer Research**
**ISSN: 2320-7167**
**Volume 08 Issue 03 March 2020, Page no. -2028-2034**
**Index Copernicus ICV: 57.55**
**DOI: 10.33826/ijmcr/v8i3.01**

# The Assessment of the Complexity of the Recursive Approach to Voxelization of Functionally Defined Objects in the Euclidean Space $E^n$

**Oleksandr Myltsev[1], Andriy Pozhuyev[2], Viktoriia Leontieva[3], Nataliia Kondratieva[4]**

[1]Department of Software Engineering, Zaporizhzhya National University, Zaporizhzhya, Ukraine
[2]Department of General Education Discipline, Zaporizhzhya National University, Zaporizhzhya, Ukraine
[3]Department of Applied Mathematics and Mechanics, Zaporizhzhya National University, Zaporizhzhya, Ukraine
[4]Department of Applied Mathematics and Mechanics, Zaporizhzhya National University, Zaporizhzhya, Ukraine

**ARTICLE INFO**

Published Online:
31 March 2020

Corresponding Author:
**Oleksandr Myltsev**
Zaporizhzhya National University, Zhukovsky str., 66, Zaporizhzhya, 69600, Ukraine +380666834300

**ABSTRACT**

The aim of this paper is to evaluate the complexity of the recursive approach to the formation of a voxel array of graphical image-models for a function-defined geometric objects. The dependence of the dimension and size of the voxel array on the dimension of the function and the number of recursion steps is determined. The dependence of voxel resolution on the size of the function research area and the number of recursion steps is considered. The dependence of the number of generated graphical image-models (M-images) and the amount of memory for their storage on the dimension of the function and the number of recursion steps are calculated.

**KEYWORDS:** recursive approach, n-variable function, R-function, voxel, voxel array, graphical image-model, graphical M-image.

## I. INTRODUCTION

Analytical spatial modeling requires an auxiliary apparatus of the scanning principle for determining and formatting a body of geometric objects based on a voxel 3D model. For the analytical presentation, the scanning apparatus can be the selected program principle for scanning the research area of function definition (iterative, recursive), and the analytical object itself is described in some compiled problem-oriented language. Thus, the creation of an instrumental system is required, which allows introducing and exploring functional dependence.

One of the most dynamically developing areas in the creation of software for automated systems is to simplify the interface between a functional user and a computer by graphically presenting information, i.e. such a representation that is adequately perceived by the user. One of the relevant issues here is the "accessibility problem" of the graphic image for visual understanding during the analysis by the researcher, which makes it possible for optimal decision making.

The solution to this problem involves the development of new methods to activate the information of interest through the use of various visual approaches. The technical capabilities of modern computing technology allow us to expand the number of approaches to visualization, taking into account the adaptation of an automated system to the features of a functional user. Along with the graphical representation of the simulated object used in the automated control system, it is possible to carry out its graphical analysis in parallel, which can be implemented as a study of the local geometric characteristics of the resulting function surface. The purpose of this analysis, carried out using a mathematical apparatus, is to highlight the basic properties and characteristics of the object under study through the behavior of the surfaces of the level of its geometric representation.

## II. FUNCTIONAL REPRESENTATION OF GEOMETRIC OBJECTS

In general terms, the functional representation of a complex geometric object considers (describes) it as a whole in the form of a closed subset of the Euclidean space $E^n$ defined by one describing function of the following form

$$f(p) \geq 0 \tag{1}$$

where $f$ – a real continuous function defined analytically or piecewise-analytic way, using set-theoretic operations of the theory of R-functions, $p = (x_1, x_2, ..., x_n)$ – point specified by coordinate variables from the function research area $H^n \subset E^n$, $n$ – function dimension [1].

Thus, $f(p) > 0$ defines points inside the geometric object, $f(p) = 0$ defines points on the surface of the object, $f(p) < 0$ defines points outside the object.

An example of such function that describes a chess pawn is presented in the next form:

$$f(x_1, x_2, x_3) = (f_1 \wedge f_2 \wedge f_3) \vee (f_4 \vee f_5) \quad (2)$$

where

$$f_1(x_1, x_2, x_3) = (\sqrt{x_1^2 + x_3^2} - 4)^2 \cdot \frac{7}{16} - x_2,$$

$$f_2(x_1, x_2, x_3) = 9 - x_1^2 - x_3^2,$$

$$f_3(x_1, x_2, x_3) = x_2 \cdot (7 - x_2),$$

$$f_4(x_1, x_2, x_3) = 1 - x_1^2 - x_3^2 - (7 - x_2)^2,$$

$$f_5(x_1, x_2, x_3) = 2 - x_1^2 - x_3^2 - 9 \cdot (6 - x_2)^2$$

The functions of n-variables $f$ on the area of research $H^n$ requires to put in accordance the voxel n-dimensional data array containing the graphical image-models (M-images) of differential geometric characteristics of the investigated function [2].

The advantages of voxel representation are as follows:

– voxel array is a volume representation of three-dimensional objects;

– it allows you to store the internal structure of an object, not just its surface;

– it is a regular data structure, which is essentially used in the methods of processing, analysis and visualization.

Research of initial function $f$ is based on the recursive algorithm of elaboration of rectangular region $H^n$ by the method of half division by mutually perpendicular planes parallel to the coordinate planes.

At each step of the recursion, we obtain $2^n$ new similar subregions to which the same procedure will be applied until the specified recursion accuracy is achieved. As a result, we obtain a voxel n-dimensional data array, which is a discrete structure for the further formation and storage of graphic image-models of the function under study.

### III. EVALUATION OF THE VOXEL DATA ARRAY

The dimension of the voxel data array corresponds to the dimension of the investigated function, i.e. is equal to $n$.

The size of the voxel massif along each $i_1 i_2 ... i_n$ index is defined as

$$I = I(r) = 2^r \quad (3)$$

where $r$ is the number of recursion steps.

Then the total number of elements of the voxel array is defined as

$$L = L(n, r) = 2^{n \cdot r} \quad (4)$$

Table 1 represents the dependence of the size of the voxel array on the dimension $n$ and the number of recursion steps $r$.

Let the size of the function research area $H^n$ along each of the coordinate axes $X_j$ be equal to $\Delta X_j$, where $j \in [1, n]$.

Then the size of the voxels along each of the coordinate axes is defined as:

$$\Delta x_j = \Delta x_j (r, \Delta X_j) = \frac{\Delta X_j}{I(r)} \quad (5)$$

Table 2 presents the dependence of the size of the voxels $\Delta x_j$ on the size of the function research area $\Delta X_j$ and the number of recursion steps $r$.

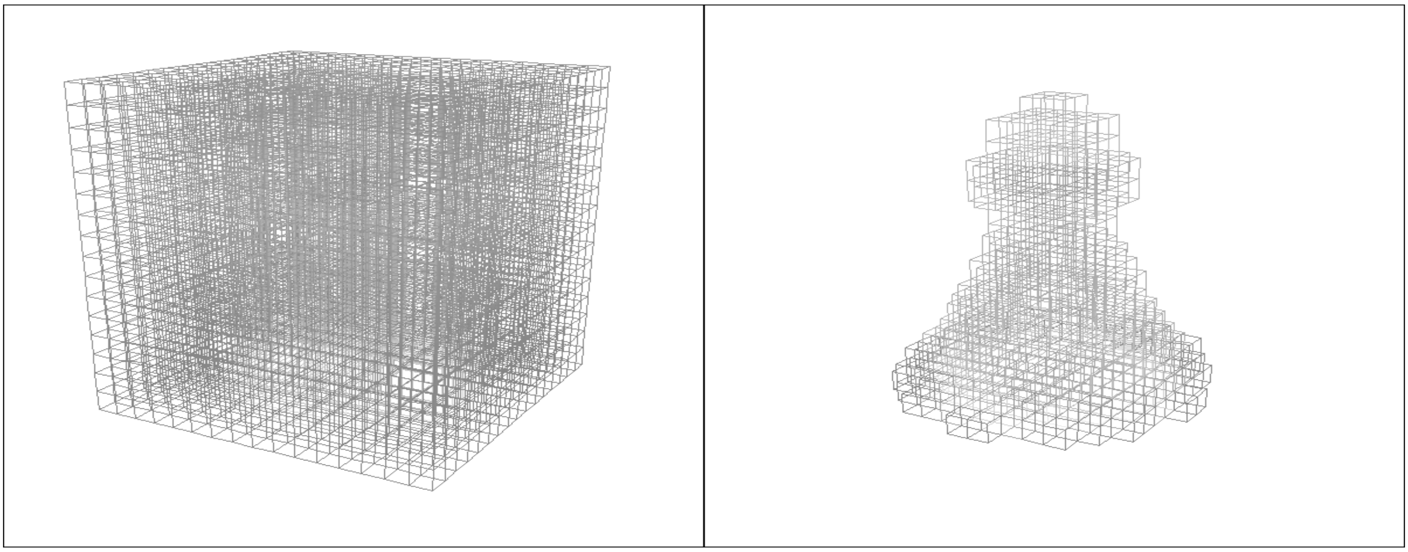
Figures 1-3 are examples of the voxel data array constructed for a function of the form (2) with different recursion steps.

**Table 1:** The size of the voxel array depending on the dimension $n$ and the number of recursion steps $r$

| r | $I(r)$ | L(n,r) n = 2 | L(n,r) n = 3 | L(n,r) n = 4 | L(n,r) n = 5 |
|---|---|---|---|---|---|
| 1 | 2 | 4 | 8 | 16 | 32 |
| 2 | 4 | 16 | 64 | 256 | 1 024 |
| 3 | 8 | 64 | 512 | 4 096 | 32 768 |
| 4 | 16 | 256 | 4 096 | 65 536 | 1 048 576 |
| 5 | 32 | 1 024 | 32 768 | 1 048 576 | 33 554 432 |
| 6 | 64 | 4 096 | 262 144 | 16 777 216 | 1 073 741 824 |
| 7 | 128 | 16 384 | 2 097 152 | 268 435 456 | 34 359 738 368 |
| 8 | 256 | 65 536 | 16 777 216 | 4 294 967 296 | 1 099 511 627 776 |
| 9 | 512 | 262 144 | 134 217 728 | 68 719 476 736 | 35 184 372 088 832 |
| 10 | 1 024 | 1 048 576 | 1 073 741 824 | 1 099 511 627 776 | 1 125 899 906 842 620 |

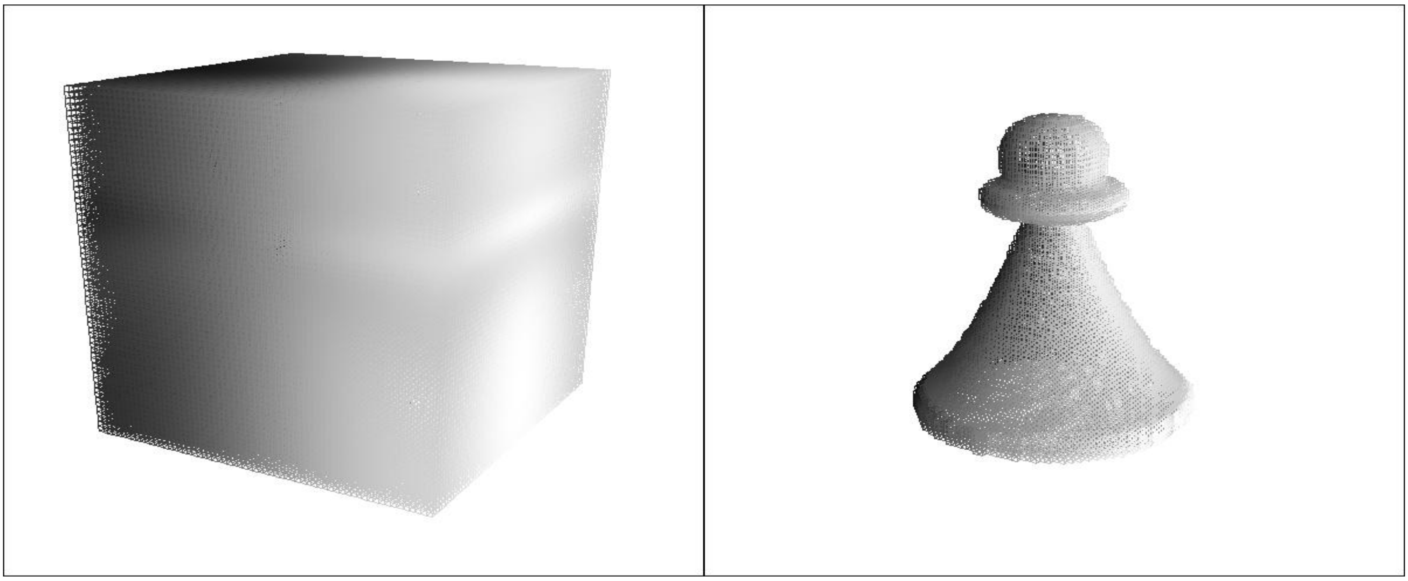
**Table 2:** The size of the voxels depending on the size of the function research area $\Delta X_j$ and the number of recursion steps $r$

| $r$ | $I(r)$ | $\Delta x_j(r, \Delta X_j)$ | | | | | |
|---|---|---|---|---|---|---|---|
| | | $\Delta X_j = 1$ | $\Delta X_j = 3$ | $\Delta X_j = 5$ | $\Delta X_j = 10$ | $\Delta X_j = 50$ | $\Delta X_j = 100$ |
| 1 | 2 | 0,5 | 1,5 | 2,5 | 5 | 25 | 50 |
| 2 | 4 | 0,25 | 0,75 | 1,25 | 2,5 | 12,5 | 25 |
| 3 | 8 | 0,125 | 0,375 | 0,625 | 1,25 | 6,25 | 12,5 |
| 4 | 16 | 0,0625 | 0,1875 | 0,3125 | 0,625 | 3,125 | 6,25 |
| 5 | 32 | 0,03125 | 0,09375 | 0,15625 | 0,3125 | 1,5625 | 3,125 |
| 6 | 64 | 0,015625 | 0,046875 | 0,078125 | 0,15625 | 0,78125 | 1,5625 |
| 7 | 128 | 0,007813 | 0,023438 | 0,039063 | 0,078125 | 0,390625 | 0,78125 |
| 8 | 256 | 0,003906 | 0,011719 | 0,019531 | 0,039063 | 0,195313 | 0,390625 |
| 9 | 512 | 0,001953 | 0,005859 | 0,009766 | 0,019531 | 0,097656 | 0,195313 |
| 10 | 1 024 | 0,000977 | 0,00293 | 0,004883 | 0,009766 | 0,048828 | 0,097656 |



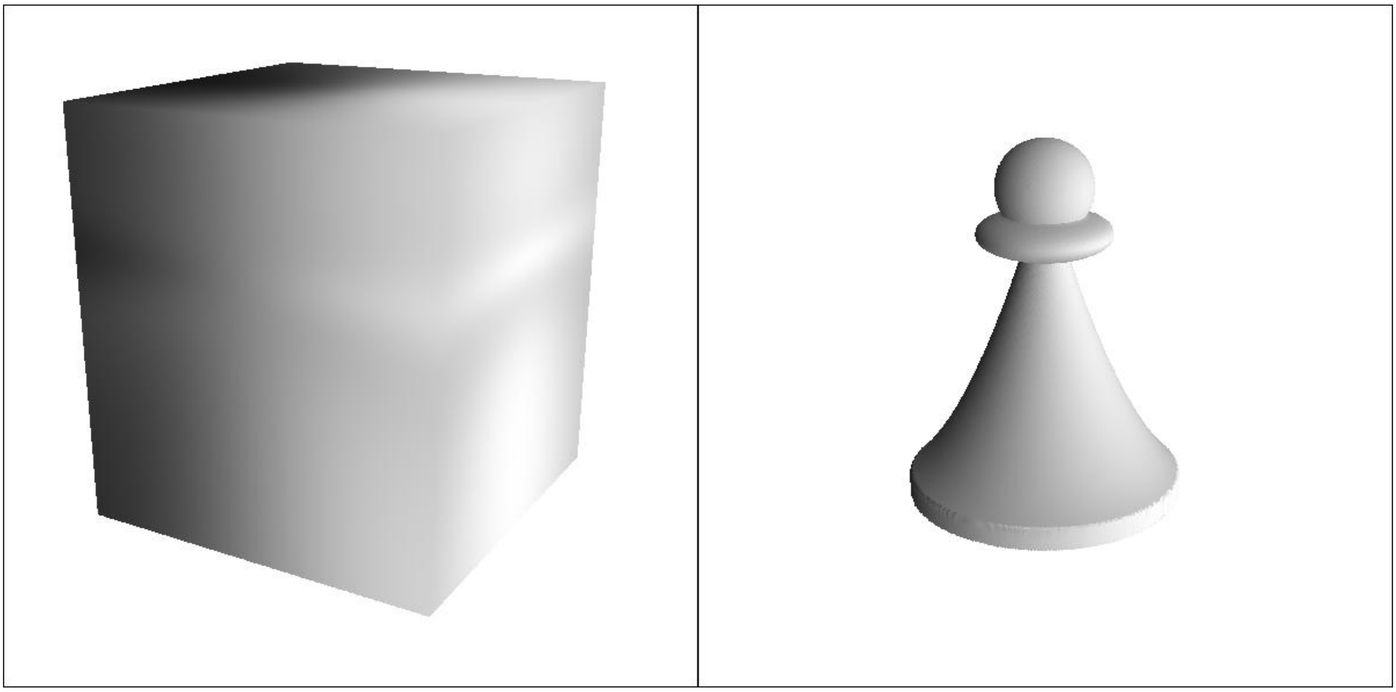
**(a)** **(b)**

**Figure 1:** Voxel data array constructed for a function of the form (2) with recursion steps $r = 4$
(**a**) entire voxel data array and **(b)** voxels inside and on the surface of an object



**(a)** **(b)**

**Figure 2:** Voxel data array constructed for a function of the form (2) with recursion steps $r = 6$
(**a**) entire voxel data array and **(b)** voxels inside and on the surface of an object



**(a)** **(b)**

**Figure 3:** Voxel data array constructed for a function of the form (2) with recursion steps $r = 8$ **(a**) entire voxel data array and **(b)** voxels inside and on the surface of an object

## IV. EVALUATION OF THE GRAPHICAL M-IMAGES

When forming a voxel data array, the body of the investigated function $x_{n+1} = f(x_1, x_2, ..., x_n)$ is represented as $(n+1)$ scalar fields of the form

$$N_f = N_{x_1}(x_1, x_2, \ldots, x_n) i_1 + N_{x_2}(x_1, x_2, \ldots, x_n) i_2 + \\ + \ldots + N_{x_{n+1}}(x_1, x_2, \ldots, x_n) i_{n+1} \quad (6)$$

where $N_{x_1}$, $N_{x_2}$,..., $N_{x_{n+1}}$ – components of the normal vector $\overline{N}$, which is calculated for each voxel.

Let us establish the correspondence of spatial scalar fields $N_{x_1}$, $N_{x_2}$,..., $N_{x_{n+1}}$ with their graphic voxel representation in the form of basic M-images $C_{x_1}$, $C_{x_2}$,..., $C_{x_{n+1}}$ through the tone intensity of the monochrome palette $P \in [0,255]$ as

$$C_{x_i} = \frac{P(1 + N_{x_i})}{2} \quad (7)$$

$i \in [1, n+1]$

The number of basic M-images is defined as

$$K_1 = K_1(n) = n + 1 \quad (8)$$

As a result, we have $(n+1)$ basic integer graphic images for each element of the n-dimensional voxel data array .

The resulting voxel array of basic graphical M-images allows us to abandon the further use of the analytical form of the function in the following graphic transformations to obtain the required number of the following image-models [3-5].

The following set of M-images characterizes the spatial position of the observer's horizon to the object and is determined through (7) as

$$C_{x_{n+1}x_i} = 2\left| C_{x_i} - P\frac{(1 + \cos\alpha_{x_i})}{2} \right| \quad (9)$$

$i \in [1, n+1]$

where angle $\alpha_{x_i} = \frac{\pi}{2}$ defines the horizon of the observer.

The number of such M-images is defined as

$$K_2 = K_2(n) = n + 1 \quad (10)$$

The following set of M-images of partial derivatives of a function is defined through (9) as

$$C_{dx_i} = \partial f / \partial x_i = \left\| \frac{C_{x_{n+1}x_i}}{C_{x_{n+1}x_{n+1}}} \right\| =$$

$$= \begin{cases} \frac{C_{x_{n+1}x_i}}{C_{x_{n+1}x_{n+1}}} \le 1 \to C_{dx_i} = P - \frac{PC_{x_{n+1}x_i}}{2C_{x_{n+1}x_{n+1}}} \\ \frac{C_{x_{n+1}x_i}}{C_{x_{n+1}x_{n+1}}} > 1 \to C_{dx_i} = \frac{PC_{x_{n+1}x_{n+1}}}{2C_{x_{n+1}x_i}} \end{cases} \quad (11)$$

$i \in [1, n]$

The number of such M-images is defined as

$$K_3 = K_3(n) = n \quad (12)$$

The following set of M-images obtained by differentiation is defined through (7) and (9) as

$$C_{x_i x_j} = \partial x_i / \partial x_j = \left\| \frac{C_{x_{n+1}x_i}}{C_{x_{n+1}x_j}} \right\| =$$

$$= \begin{cases} C_{x_i} \ge \frac{P}{2} \to \begin{cases} \frac{C_{x_{n+1}x_j}}{C_{x_{n+1}x_i}} \le 1 \to C_{x_i x_j} = \frac{PC_{x_{n+1}x_j}}{4C_{x_{n+1}x_i}} \\ \frac{C_{x_{n+1}x_j}}{C_{x_{n+1}x_i}} > 1 \to C_{x_i x_j} = \frac{P}{2} - \frac{PC_{x_{n+1}x_i}}{4C_{x_{n+1}x_j}} \end{cases} \\ C_{x_i} < \frac{P}{2} \to \begin{cases} \frac{C_{x_{n+1}x_j}}{C_{x_{n+1}x_i}} \le 1 \to C_{x_i x_j} = P - \frac{PC_{x_{n+1}x_j}}{4C_{x_{n+1}x_i}} \\ \frac{C_{x_{n+1}x_j}}{C_{x_{n+1}x_i}} > 1 \to C_{x_i x_j} = P - \left( \frac{P}{2} - \frac{PC_{x_{n+1}x_i}}{4C_{x_{n+1}x_j}} \right) \end{cases} \end{cases} \quad (13)$$

$i \in [1, n]$

$j \in [1, n]$

$i \ne j$

The number of such M-images is defined as

$$K_4 = K_4(n) = n \cdot (n-1) \quad (14)$$

The total number of M-images for the function of n-variables is defined as

$$K = K(n) = K_1(n) + K_2(n) + K_3(n) + K_4(n) = \\ = n^2 + 2 \cdot (n+1) \quad (15)$$

Table 3 presents the dependence of the number of M-images on the dimension of the function.

## V. Evaluation of the Information Volume

Let $B_p = 1$ Byte – the amount of memory for the tone intensity of a monochrome palette $P \in [0, 255]$, $B_S = 1$ Byte – the amount of memory needed to store additional features of the function.

Then the amount of information obtained during the formation of the basic graphical M-images of the function of n-variables with the number of recursion steps $r$ is equal to

$$B_1 = B_1(n, r, B_P, B_S) = \\ = L(n, r) \cdot (K_1(n) \cdot B_p + B_S) = \\ = 2^{n \cdot r} \cdot ((n+1) \cdot B_P + B_S) \quad (16)$$

The amount of memory required to store all graphical M-images of the function of n-variables with the number of steps of the recursion $r$ is equal to

$$B = B(n, r, B_P, B_S) = \\ = L(n, r) \cdot (K(n) \cdot B_p + B_S) = \\ = 2^{n \cdot r} \cdot ((n^2 + 2 \cdot (n+1)) \cdot B_P + B_S) \quad (17)$$

Tables 4-5 represent the dependence of amount of memory to store the graphical M-images on the number of steps of the recursion $r$.

**Table 3:** The number of M-images for the function of n-variables

| $n$ | $K_1$ | $K_2$ | $K_3$ | $K_4$ | $K$ |
|---|---|---|---|---|---|
| 1 | 2 | 2 | 1 | 0 | 5 |
| 2 | 3 | 3 | 2 | 2 | 10 |
| 3 | 4 | 4 | 3 | 6 | 17 |
| 4 | 5 | 5 | 4 | 12 | 26 |
| 5 | 6 | 6 | 5 | 20 | 37 |

**Table 4:** The amount of information obtained during the formation of basic graphical M-images

| $r$ | $B_1(n, r, B_P, B_S)$ (Byte) | | | |
|---|---|---|---|---|
| | $n = 2$ | $n = 3$ | $n = 4$ | $n = 5$ |
| 1 | 16 | 40 | 96 | 224 |
| 2 | 64 | 320 | 1 536 | 7 168 |
| 3 | 256 | 2 560 | 24 576 | 229 376 |
| 4 | 1 024 | 20 480 | 393 216 | 7 340 032 |
| 5 | 4 096 | 163 840 | 6 291 456 | 234 881 024 |
| 6 | 16 384 | 1 310 720 | 100 663 296 | 7 516 192 768 |
| 7 | 65 536 | 10 485 760 | 1 610 612 736 | 240 518 168 576 |
| 8 | 262 144 | 83 886 080 | 25 769 803 776 | 7 696 581 394 432 |
| 9 | 1 048 576 | 671 088 640 | 412 316 860 416 | 246 290 604 621 824 |
| 10 | 4 194 304 | 5 368 709 120 | 6 597 069 766 656 | 7 881 299 347 898 370 |

**Table 5:** The amount of memory required to store all graphical M-images

| $r$ | $B(n,r,B_P,B_S)$ (Byte) | | | |
|---|---|---|---|---|
| | $n=2$ | $n=3$ | $n=4$ | $n=5$ |
| 1 | 44 | 144 | 432 | 1 216 |
| 2 | 176 | 1 152 | 6 912 | 38 912 |
| 3 | 704 | 9 216 | 110 592 | 1 245 184 |
| 4 | 2 816 | 73 728 | 1 769 472 | 39 845 888 |
| 5 | 11 264 | 589 824 | 28 311 552 | 1 275 068 416 |
| 6 | 45 056 | 4 718 592 | 452 984 832 | 40 802 189 312 |
| 7 | 180 224 | 37 748 736 | 7 247 757 312 | 1 305 670 057 984 |
| 8 | 720 896 | 301 989 888 | 115 964 116 992 | 41 781 441 855 488 |
| 9 | 2 883 584 | 2 415 919 104 | 1 855 425 871 872 | 1 337 006 139 375 620 |
| 10 | 11 534 336 | 19 327 352 832 | 29 686 813 949 952 | 42 784 196 460 019 700 |

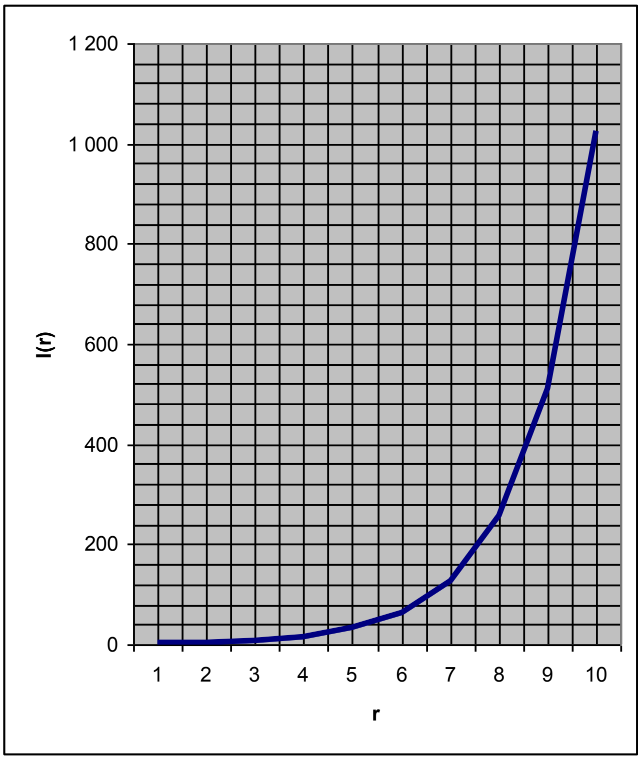
## VI. CONLCUSION

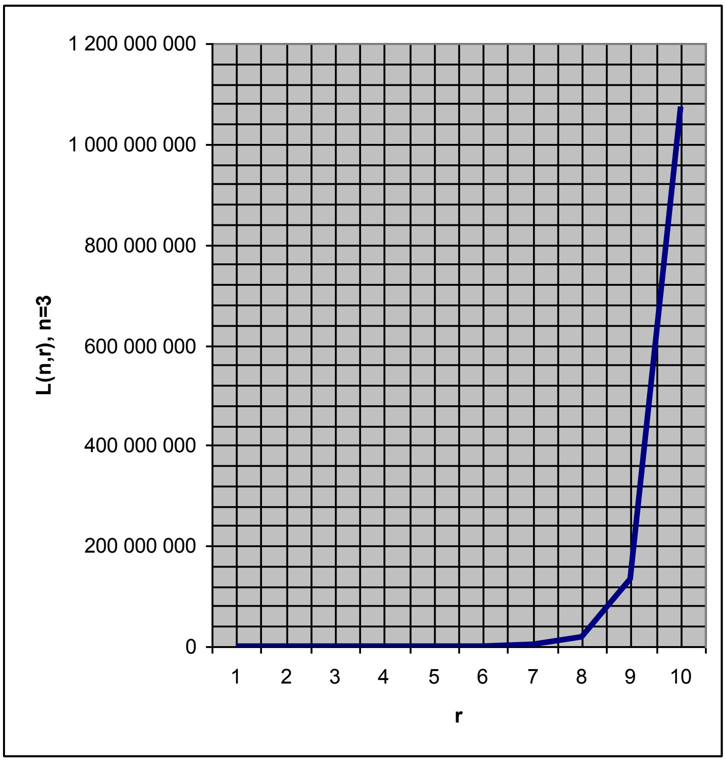
In Figure 4, we can evaluate the dependence of the size of the voxel data array and the amount of information obtained during the formation of the basic graphic M-images of the function of three variables on the number of recursion steps.

Based on these findings, it can be concluded that the total number of elements of the voxel array $L$ and the amount of information $B_1$ begin to increase sharply at $r>8$, and the size of the voxel $\Delta x_j$ decreases by not much.

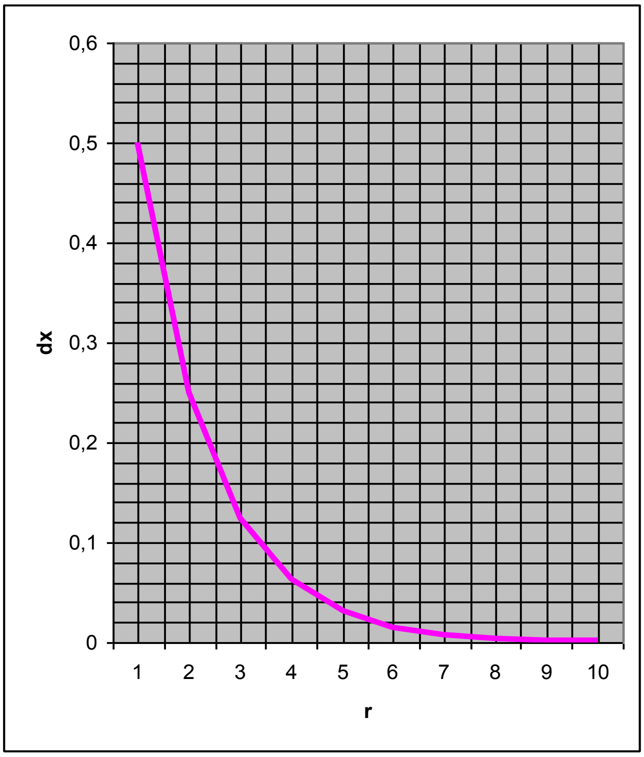
All things considered, we can assume that the optimal combination of the quality of the formed voxel graphic M-images and memory costs is achieved at $r=8$.

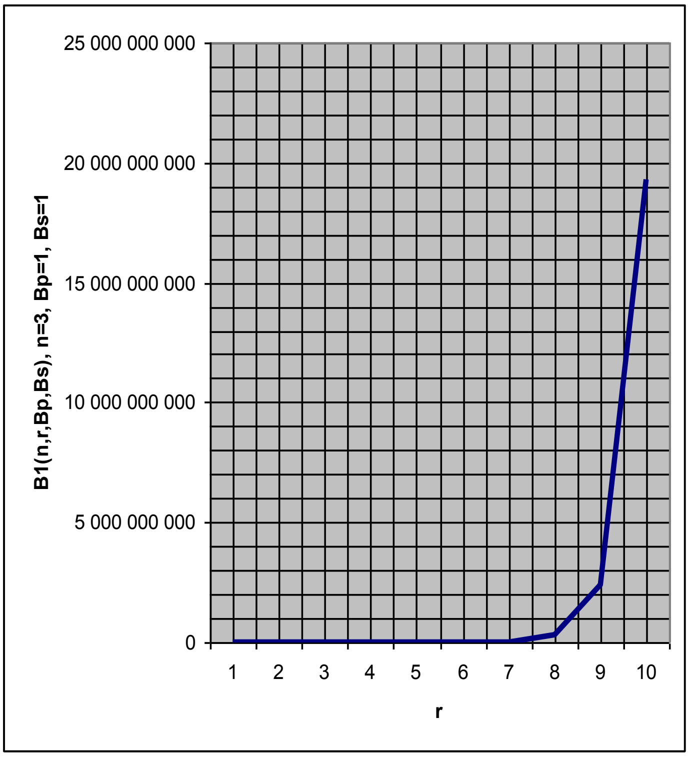
Moreover, we can estimate the computational complexity of the algorithm as $O(2^{n \cdot r})$ or, in general, as $O(2^N)$.




**(a)** **(b)**




**(c)** **(d)**

**Figure 4:** Dependency graphs

**(a)** $I(r)$ **(b)** $L(n,r)$, $n=3$ **(c)** $\Delta x_j(r,\Delta X_j)$, $\Delta X_j=1$ **(d)** $B_1(n,r,B_P,B_S)$, $n=3$, $B_P=1Byte$, $B_S=1Byte$

## REFERENCES


1. Rvachev V.L., Tolok A.V., Uvarov R.A., & Sheyko T.Y. (2000). New approaches to the construction of equations of three-dimensional loci using R-functions. Visnyk Zaporiz′koho Derzhavnoho universytetu: Zbirnyk naukovykh statey. Fizyko-matematychni nauky, No.2, pp.119–131.
2. Tolok A.V., Myltsev A.M. & Korohod V.L. (2006). Analytical modeling based on graphic transformations in the RANOK system. Visnyk Zaporiz′koho natsional′noho universytetu: Zbirnyk naukovykh statey. Fizyko-matematychni nauky, No.1, pp.124–133.
3. Myltsev O.M. (2018). Analysis of the functions of three variables based on voxel structures of images-models in the system "RANOK". Visnyk Zaporiz′koho natsional′noho universytetu: Zbirnyk naukovykh statey. Fizyko-matematychni nauky, No.1, pp.89–97.
4. Myltsev O.M. Kondratieva N.O. & Leontieva V.V. (2018). Functional model of the basic business processes of the "RANOK" system. Visnyk Zaporiz′koho natsional′noho universytetu: Zbirnyk naukovykh statey. Fizyko-matematychni nauky, No.2, pp.88–99.
5. Morozov D.N., Gnezdovskiy A.V., Myltsev A.M., & Tolok A.V. (2010). Cognitive computer graphics in the process of solving optimization problems of mathematical modeling. Prikladna geometríya ta ínzhenerna grafíka, Issue 86, pp.112–117.