Generating view models of monotone polyhedrons for visual identification. Presentation of results of the PhD thesis

Andrzej Salamonczyk[1], Wojciech S. Mokrzycki[2]
[1]*Faculty of Science, University of Natural Sciences and Humanities, Siedlce, Poland*
andrzej.salamonczyk@uph.edu.pl
[2]*Faculty of Mathematics and Natural Sciences, The Cardinal Wyszynski University, Warsaw, Poland*
w.mokrzycki@uksw.edu.pl

Abstract. Main results the PhD dissertation are presented in which the goals were: to develop appropriate methods and algorithms in order to obtain a complete view representation of the object using the idea of the viewing sphere with the perspective projection, to compare and select the best algorithm for this task, and to show how the viewing models can be matched with objects extracted from range images. It was proved that it is possible to obtain the approximate view representation of a monotone polyhedron using the algorithm with the complexity of $O(n^2)$ and that the minimum number of matching view representations of a monotone polyhedron given with the data from range images is $O(n^2)$. The methods of generating view representations of monotone polyhedrons are investigated for the purpose of visual recognition systems and some of them are presented. View representations are used in many areas including identification tasks, e.g. as an element of visual channel in automation and robotics. The 3D view representation of objects can be constructed offline and saved in a database for further use in the recognition task.

Key words: view representation, generation of view representation, view models.

1. Introduction

The subject of the dissertation *Generating view models of monotone polyhedrons for visual identification* (in Polish: *Generowanie modeli widokowych wielościanów monotonnych do identyfikacji wizualnej*) [12], are the methods of generating the view representation of monotone polyhedrons with the destination for visual recognition systems. Representations of this type are used, among others, in identification tasks, e.g. as an element of a visual channel in automation or robotics in tasks of the type: at the beginning there is a given set of objects that should be recognized in the robot's surroundings.

The goal of the dissertation is to develop methods and algorithms suitable for obtaining a complete visual representation of the object using the concept of the view sphere with perspective projection, comparison and selection of the best algorithm for this task and presenting the method of fitting view models of objects with the appearance of objects acquired from the environment using depth maps.

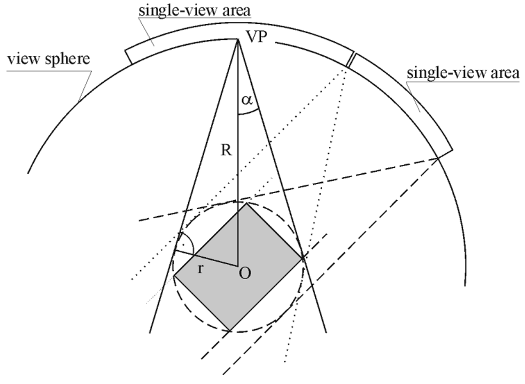

Fig. 1. Idea of viewing sphere with perspective.

The idea of generating views using the view sphere with perspective projection was presented in [3, 4], Fig. 1.

The generation space model is constructed in a following way. First, first circumscribe a sphere on a polyhedron. The small sphere (radius r) and its center is the same as the polyhedron center. On this sphere place a space view cone with angle 2α. This is the *viewing cone*. The vertex of this cone is a model *viewing point VP*. Unconstrained movement of the cone vertex, where the cone is tangent to the small sphere, creates a large sphere with radius R. This sphere is called *view sphere*. Each object has its own viewing sphere, the same for all views of this particular object. Views are obtained taking into account only object features selected for identification. i.e. faces. Set of faces visible from a view point is called *view*. The area on the view sphere, for which the view does not change, is *single-view area*. Database of view models is a set of all views together with information, to which polyhedron (or polyhedrons) the given view belongs. Completeness of the representation means that there are all possible views of the tested polyhedron or that a view obtained from each viewpoint is in the created representation.

This model was initially planned to obtain views for convex polyhedrons. However, it should be used to generate views for a wider class of solids. With this projection method, if it is desired for each wall to have a view in which it is completely visible, a condition is that the segment $\overline{OVP}$ intersects the shell of the polyhedron W at exactly one point.

$$\forall \overrightarrow{VP} \ \exists! p \in R^3 : \ \ W \cap \overline{OVP} = p \ , \tag{1}$$

and the intersections of polyhedron were star-shaped. Such class of polyhedrons is called **monotone polyhedrons**.

Many algorithms can be found in the literature describing the generation of view representations, examples are discussed in the next section.

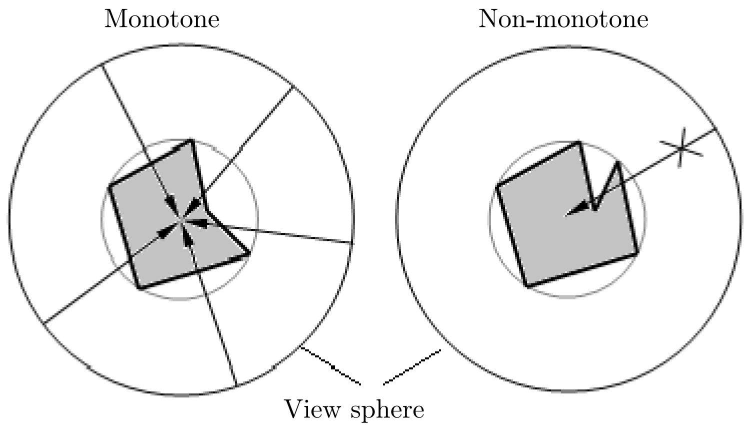

Fig. 2. Monotone and non-monotone polyhedrons.

2. Review of previous works

The way objects are represented is an important step that affects the recognition process. Jain et al. in [2] define the key elements of the design of the object recognition system:

1. Choosing the right representation of objects and creating a database containing these objects.
2. Acquisition of data and isolation of distinguished features.
3. Choosing the right way to fit data with the model.
4. Determining how to select an object (objects) matching the data with the highest probability.
5. Determination of the verification method set in the previous step of the hypothesis.

Choosing the right representation is one of the most important issues. There are many ways to represent geometric objects in the literature. Some of them are used for recognition in visual systems, while others (although commonly used, for example, for rendering) have not been used in recognition systems. It is natural to look for a way to recognize using views. The idea of aspect graphs (conceptually very close to the view representation) was proposed by Koenderink and van Doorn, [1].

An important criterion for the evaluation of view generation algorithms is the completeness of the representation generated by them. One of the ways to generate complete view representations is to iterate generating a view up to covering the entire sphere with single-view areas (hence the term *iterative algorithm*).

This group of algorithms begins with generating a view. The next step is to determine the viewing areas corresponding to the obtained views. Next, it is checked whether the entire viewing sphere is already covered by single-view areas. The exact algorithms for generating these views can be found in [3]. The authors use the observation that each single-view area has as many neighbors as there are edges. One of the most important steps in this algorithm is the accurate determination of the single-view area.

Other ways of generating views do not require checking the completeness of generating the representation; algorithms operate without loops and are classified as non-iterative algorithms. The method of crossing the borders of the seed area, [7], is the following: the first view is calculated at any viewpoint and its single-view area, e.g. by the method using a half plane. This first single view area becomes the starting area on the viewing sphere, to which successive single areas adjacent along all edges of the seed area are be successively added (in a spiral way).

The method of tracking the border of a single-view area, [5], includes the enumeration of views adjacent to the already determined views, using the boundary planes of previously designated single-view areas, and then finding the corresponding single-view areas. Searching for views and single-view areas takes place spirally around the first seeding single-view area and leads to the full coverage of the sphere, that is, to generating a full set of views. The method of moving along the border of the single-view is considered as the best for the convex polyhedrons area by Kowalczyk, as the co-author of the method together with Mokrzycki [5]. The computational complexity of the method is $O(n^4)$.

The concept of the algorithm denoted by $FM^{V_{rep}\cap}$, [6], is based on a vector representation of a polyhedron V_{rep} and was rising from the observation that the convex polyhedron with visibility is determined by the angle between the normal vector and the viewing direction. Therefore, it should be a sufficient vector representation to generate a view representation of a polyhedron. This method uses the concept of a complementary cone.

One of the first ideas (developed by the author of the present paper) was to move around the face in such a way that it is visible all the time. This gave rise to the concept of limiting the class of solids to monotone polyhedrons, which allowed the entire face to fit into the complementary cone.

The most important difference in relation to the algorithm based on the vector representation is a different than previously, much more complex movement of the scanning cone. In general, the scanning cone carries out scanning movements not around a straight line coinciding with the direction of the normal vector of a face, but around the face itself, being supported on the lateral edges of the lateral pyramid of the viewing face until it touches the initial edge again [8].

Approximate methods form another group for generating a viewing representation. It follows from the observation that in the recognition process a view is needed, rather than an area in which such a view can be obtained, and some single-view areas are small, so there is a little chance to find such a small view. An open problem still remains, however, how to move the view point on the sphere, and how to arrange in the best way

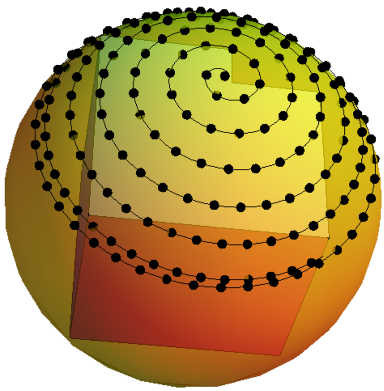

Fig. 3. An example of a trajectory of scanning over the face visibility area.

the viewing points on the track of this movement to generate a complete representation with required accuracy.

One of the methods used, as in the exact generation methods, is to determine all views with the participation of a given face, and to move the view point over the wall in a spiral way, see Fig. 3, [9].

Tests of this algorithm have shown that a complete set of views is generated, with accuracy to the desired resolution. However, what was one of the ideas in creating this method, i.e. generating all views involving a given face, leads to the main drawback of this method, which is multiple scanning of the same area.

The method using a regular icosahedron is free from this disadvantage. In the first step of this method, viewpoints are placed in the vertices of the regular icosahedron inscribed in the viewing sphere. Then, each wall is divided into four smaller triangles, see Fig. 4, by generating new vertices lying on the edge centers. The process is repeated until the number of view points received so far ensures the required minimum distance between points.

The method of generating views using a regular icosahedron involves the following steps.

1. *Creating a set of view points.*
2. *Acquiring a view from each viewpoint* and *adding it to the set of views.*
3. *Removing duplicate views* from the set of views.

The refinement of the method is to take advantage of the fact that the single view area is convex. When the same view from the three corners of the triangle is obtained, there is no need to divide the triangle further, because the same view would be received again, see Fig. 5.

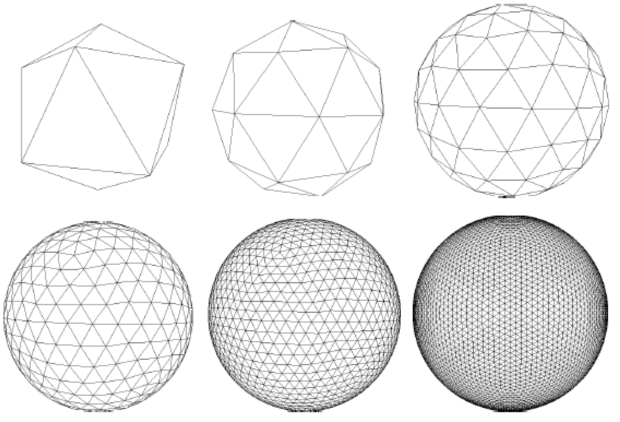

Fig. 4. Creating view points through an iterative division of a regular icosahedron.

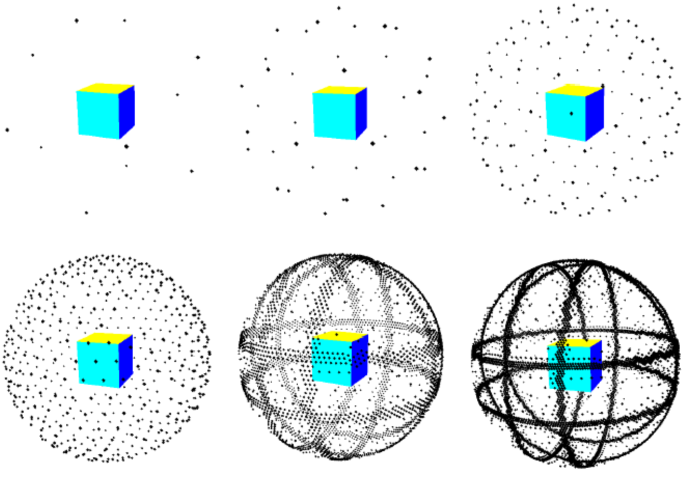

Fig. 5. A set of view points in successive regular icosahedron iterations, taking into account convex one view areas.

3. Results

Main theses of dissertation are:

1. It is possible to obtain the approximate view representation of a monotone polyhedron using the algorithm with the complexity of $O(n^2)$.
2. Minimum number of matching view representation of monotone polyhedron and data from range images is $O(n^2)$.

To prove the first statement we show an example of a method of view point generation.

3.1. Generating view representation in a spiral way

In this method, [10], the spiral trajectory is placed on the entire viewing sphere. We also demand that the view points should be arranged as evenly as possible on the spiral trajectory.

We start with the formulas (2) on the spiral trajectory on the sphere:

$$\begin{cases} x = R\cos(kt)\sin(t) \ , \\ y = R\sin(kt)\sin(t) \ , \\ z = R\cos(t) \ . \end{cases} \tag{2}$$

We use (3) for the correction of the unevenness of the distribution of view points on the trajectory, $f(s) : [0, 2] \to [0, 2\pi]$ (Fig. 6):

$$\begin{cases} t = \arccos(1 - s); \text{ for } s \in [0, 1] \\ t = \pi - \arccos(s - 1); \text{ for } s \in (1, 2]. \end{cases} \tag{3}$$

Substituting for t in (2) the parameter defined by (3) we get a much more even distribution of points.

By requesting that the angle between consecutive turns of the spiral be equal to the desired scan resolution (denoted by σ), we obtain the formula for k:

$$k = \frac{2\pi}{\sigma} \ .$$

Taking into account this equality and the fact that the spiral length defined by (2) is less than $2k + 1$ (this estimation is true for $k > 3$, in the literature (e.g. [11]) more exact estimates can be found, but they contain non-linear functions, increasing the time of generating points and do not significantly reduce the number of viewpoints), formula for the number of points n needed on a spiral trajectory, to obtain the desired resolution σ, can be written down as $n = \frac{2\pi+\sigma}{\sigma^2}$. Thus, the largest distance between consecutive points equals σ and the angle between consecutive turns of spiral is also equal to σ. The most unfavorable location of the view point is in the middle of the square with the side σ, so to ensure the minimum resolution σ we can multiply it by $\sqrt{2}$.

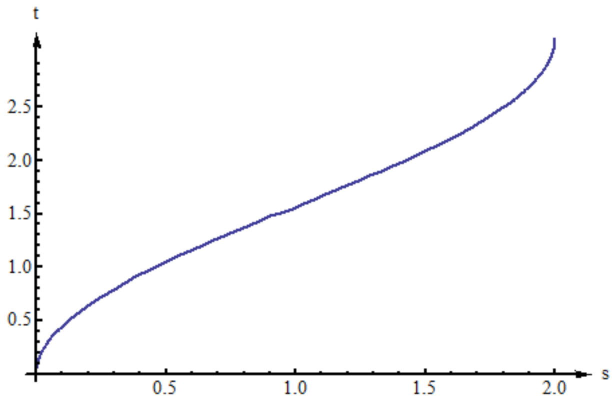

Fig. 6. The function of correction of the distribution of points on the trajectory.

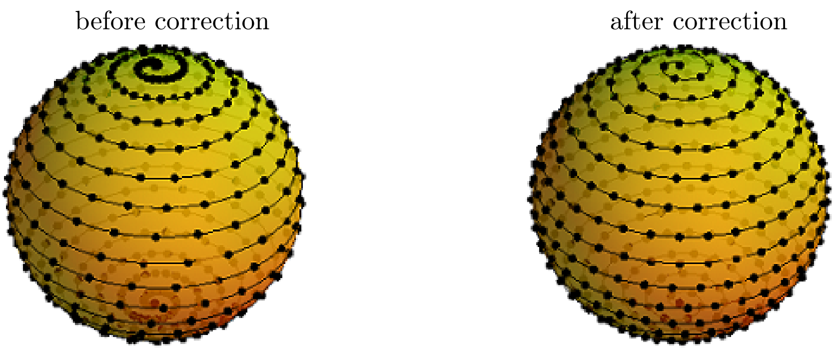

Fig. 7. Spiral distribution of points on the viewing sphere obtained by the equation 2 and after applying the distribution correction function 3.

Taking into account these estimates, we obtain the following formula for the number of view points N in the spiral scanning method of the viewing sphere:

$$N = \frac{4\pi + \sqrt{2}\sigma}{2\left(\sigma\right)^2}\,. \tag{4}$$

The number of viewpoints is inversely proportional to the square of the scanning resolution. An exemplary distribution of points on the trajectory obtained by this method is shown in Fig. 7. The resolution does not depend on the shape of polyhedron, but on the technical parameters of the acquisition device.

The way of generating views in this method can be presented in three steps:

1. *Determining the view points* placed on a spiral trajectory on the viewing sphere for a given scanning resolution.
2. *Generating a view from each view point* and *adding it to the set of views.*
3. *Removing duplicate views* from the set of generated views.

By proceeding in the described way, a set of views is obtained with the accuracy according to the desired resolution.

In the dissertation two lemmas were proved.

Lemma 3.1. The angle between consecutive turns of the spiral is constant and equal to $\frac{2\pi}{k}$.

Lemma 3.2. The angle between consecutive viewpoints arranged on a spiral trajectory does not exceed σ.

With the help of lemma 3.1 and lemma 3.2, the following theorem is proven:

Theorem 3.1. Each view, in the viewing area of which a cone can be inscribed with a vertex lying in the center of the viewing sphere and with a half of cone angle equal to the scan resolution σ, will be generated by the spiral scanning method.

Further, between the angles θ and ϕ the following relationship occurs:

$$\theta = k \cdot \phi \, . \tag{5}$$

The angle $\phi \in [0, \pi]$ and the number of full revolutions around the z axis (this axis was adopted to facilitate the recording) is determined by the value of $2k$. In this equation, the angle between consecutive spiral turns is equal to $\frac{2\pi}{k}$. However, the points obtained on the trajectory are not evenly distributed (much more densely close to the poles, see Fig. 7).

The main advantages of the described way of generating the views are:

1. *no overlapping areas* of scanning, which leads to a significantly smaller number of viewpoints;
2. generally *less viewpoints* are generated (e.g. Fig. 8);
3. the set of viewpoints is *ordered* (it is convenient in constructing other structures, e.g. for generating aspect graphs);
4. the set of view points is distributed more *evenly* (In comparison with the algorithm using division of a regular icosahedron).

3.2. The computational complexity of the spiral method

The computational complexity of the method for determining the polyhedron views depends of course on the number of faces in the polyhedron (let us denote it by n) and the number of viewpoints on the sphere (assume the notation N) and can be described by the following relationship:

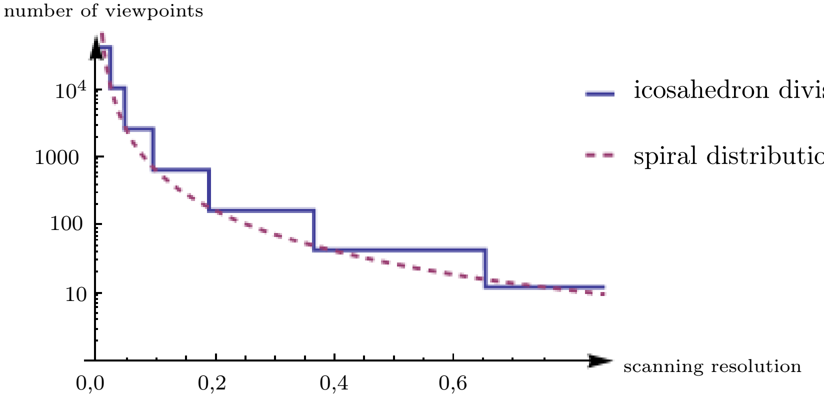

Fig. 8. Comparison of the number of viewpoints depending on the desired resolution for the methods: with the division of the regular icosahedron and the even spiral.

$$Z = k_1 + N \cdot k_2 + k_3, \tag{6}$$

where k_1 is the cost of determining N view points (linear dependence) (step 1), k_2 is the cost of obtaining a view from the viewpoint, and k_3 is the cost of removing repeating views in the base. The cost of determining the visibility of the face in the polyhedron is linear, and thus the cost of defining the visible faces of the polyhedron from the viewing point has a complexity of n^2. The next step is to remove duplicate views from the N -element set, which can be done at the cost of $n \cdot N \log N$ operations [1]. Therefore, the computational complexity of the method can be written with the formula:

$$Z = O(N + N \cdot n^2 + n \cdot N \log N)\ . \tag{7}$$

If a resolution of a scanning process is constant, and the cost of the algorithm depends only on the shape of the solid, then $N = \text{const}$ and $N \log N = \mathit{const}$ and we get the estimation:

$$Z = O(n^2)\ , \tag{8}$$

so that the problem has a square complexity. In this way the first thesis of the dissertation was proved.

[1]This is due to the fact that view labels (length n) can be sorted, which can be done in time $O(n \cdot N \log N)$, and then in a linear time the repeating elements are deleted.

3.3. Views used for identification purpose

Identification of a 3D object based on one view of the object obtained from the view sphere is not always possible (but it can be stated that this is not the object we are looking for). A question can be asked: how many views are needed to uniquely identify a monotone polyhedron? This problem can also be formulated as: *what is the minimum number of matches* of the acquired data sufficient to correctly identify the monotone polyhedron.

Note that in every monotone polyhedron view, at least one face is completely visible (in a special case we see only this face). A view containing a single completely visible face is enough to identify a face, but it does not give any information about its position relative to other faces. Therefore, this view should generally be supplemented with views containing other faces.

A set of such views that fully covers the polyhedron surface can be constructed in the following way:

1. For each wall, we select views in which this face is completely visible.
2. In addition, we request that these views include neighboring faces.
3. We match the acquired view with the view from the database that meets the given assumptions and if there is such a match, we repeat the process for other faces.

Thus, the minimum number of matches is certainly less than or equal to the sum of the number of neighbors for each wall. It follows that the minimum number of matches $M_{\min}$ can be estimated in the following way:

$$M_{\min} = O(n^2). \tag{9}$$

In this way, the second thesis of the dissertation was shown. Proceeding in the manner described above, we obtain a full coverage of the tested polyhedron by a set of views sufficient for the identification of the object.

3.4. Examples

Obtained views for two solids are shown here. For a polyhedron shown in Fig. 9 we obtain 46 views. The vertices, faces and views for this polyhedron are collected in Table 1. For a monotone polyhedron shown in Fig. 10 there are 66 views. Its vertices, faces and views for this polyhedron are collected in Table 2.

4. Conclusion

In the presented dissertation, and also in this paper, the methods of generating the viewing representation are shown. Important reasons for abandoning the precise representation of the title class of solids were:

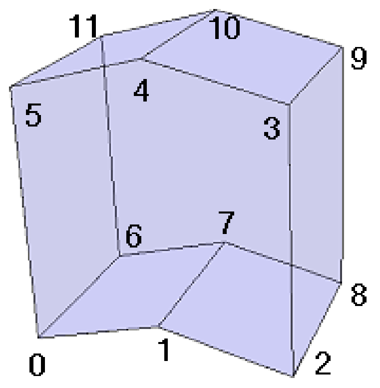

Fig. 9. Polyhedron with 8 faces (46 views).

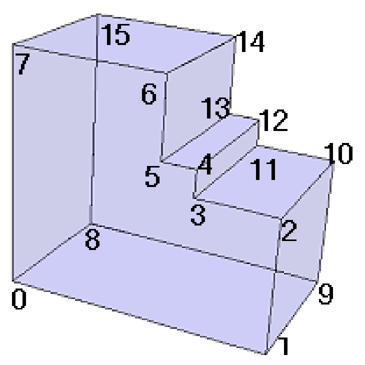

Fig. 10. Monotone polyhedron with 10 faces (66 views).

- a large number of potential single-view areas on the viewing sphere $O(n^6)$;
- difficulties with describing the intersection area of the event surface of the type *second degree curve* with the view sphere;
- generating views that are of little importance to recognition systems (low probability of obtaining a view, often smaller than the accuracy of the acquiring device) and increasing the number of views.

The results presented in this work expand the class of solids for which it is able to determine its visual representation (previously it was able to do for convex polyhedrons and only the first attempts were made for a wider class of solids). The approximate methods of generating the viewing representation presented in the work are so universal

Tab. 1. Vertices, faces and views of the polyhedron of Fig. 9.

Vertices:

No.	Coordinates
0	(-4.0, -4.0, 3.0)
1	(0.0, -3.0, 3.0)
2	(4.0, -4.0, 3.0)
3	(4.0, 4.0, 3.0)
4	(0.0, 5.0, 3.0)
5	(-4.0, 4.0, 3.0)
6	(-4.0, -4.0, -3.0)
7	(0.0, -3.0, -3.0)
8	(4.0, -4.0, -3.0)
9	(4.0, 4.0, -3.0)
10	(0.0, 5.0, -3.0)
11	(-4.0, 4.0, -3.0)

Faces:

No.	Vertices
0	1, 0, 6, 7
1	2, 1, 7, 8
2	3, 2, 8, 9
3	4, 3, 9, 10
4	5, 4, 10, 11
5	0, 5, 11, 6
6	1, 2, 3, 4, 5, 0
7	6, 11, 10, 9, 8, 7

Views:

No.	Faces
0	6
1	1 6
2	0 6
3	0 1 6
4	5 6
5	2 6
6	1 5 6
7	0 2 6
8	3 6
9	4 6
10	3 4 6
11	0 1 2 6
12	2 3 6
13	4 5 6
14	0 1 5 6

No.	Faces
15	2 3 4 6
16	3 4 5 6
17	0 1 2
18	0 2
19	2
20	2 3
21	2 3 4
22	3 4
23	3 4 5
24	4 5
25	5
26	1 5
27	0 1 5
28	0 1
29	3 4 7

No.	Faces
30	3 4 5 7
31	4 5 7
32	5 7
33	1 5 7
34	0 1 5 7
35	0 1 7
36	0 1 2 7
37	0 2 7
38	2 7
39	2 3 7
40	2 3 4 7
41	4 7
42	3 7
43	7
44	1 7

No.	Faces
45	0 7

that after a slight modification (this applies mainly to the method of determining visible surfaces), they can be successfully used for solids of other types. This is especially true for the general polyhedron class, but it can also be used for other objects.

The main directions of further research pointed in the dissertation are the development of ways to generate the representation of objects with a curvilinear surface, as well as practical applications of visual representation in identification and identification tasks, especially in robotics.

Tab. 2. Vertices, faces and views of the polyhedron of Fig. 10.

Vertices:

No.	Coordinates
0	(-5.0, -4.0, 3.0)
1	(5.0, -4.0, 3.0)
2	(5.0, 1.0, 3.0)
3	(2.0, 1.0, 3.0)
4	(2.0, 2.0, 3.0)
5	(0.75, 2.0, 3.0)
6	(0.75, 5.0, 3.0)
7	(-5.0, 5.0, 3.0)
8	(-5.0, -4.0, -3.0)
9	(5.0, -4.0, -3.0)
10	(5.0, 1.0, -3.0)
11	(2.0, 1.0, -3.0)
12	(2.0, 2.0, -3.0)
13	(0.75, 2.0, -3.0)
14	(0.75, 5.0, -3.0)
15	(-5.0, 5.0, -3.0)

Faces:

No.	Vertices
0	0,1,2,3,4,5,6,7
1	2,1,9,10
2	3,2,10,11
3	4,3,11,12
4	5,4,12,13
5	6,5,13,14
6	7,6,14,15
7	0,7,15,8
8	1,0,8,9
9	15,14,13,12,11,10,9

Views:

No.	Faces
0	0
1	0 5
2	0 2
3	0 2 5
4	0 2 4
5	0 3 5
6	0 2 4 5
7	0 2 3 5
8	0 2 3 4 5
9	0 8
10	0 5 8
11	0 3 5 8
12	0 1 3 5

No.	Faces
13	0 2 4 6
14	0 7
15	0 2 4 5 6
16	0 2 7
17	0 1 2 3 5
18	0 2 4 7
19	0 1 2 3 4 5
20	0 2 3 4 5 6
21	0 7 8
22	0 1 3 5 8
23	0 2 4 6 7
24	0 1 2 3 4 5 6
25	2 3 4 5 6

No.	Faces
26	2 4 5 6
27	2 4 6
28	2 4 6 7
29	2 4 7
30	2 7
31	7
32	7 8
33	8
34	5 8
35	3 5 8
36	1 3 5 8
37	1 3 5
38	1 2 3 5

No.	Faces
39	1 2 3 4 5
40	1 2 3 4 5 6
41	1 3 5 8 9
42	1 3 5 9
43	1 2 3 5 9
44	1 2 3 4 5 9
45	1 2 3 4 5 6 9
46	2 3 4 5 6 9
47	2 4 5 6 9
48	2 4 6 9
49	2 4 6 7 9
50	2 4 7 9
51	2 7 9

No.	Faces
52	7 9
53	7 8 9
54	8 9
55	5 8 9
56	3 5 8 9
57	2 3 4 5 9
58	2 4 9
59	9
60	3 5 9
61	2 4 5 9
62	2 9
63	2 3 5 9
64	5 9
65	2 5 9

References

1979

[1] J. Koenderink, A. van Doorn. The Internal Representation of Solid Shape with Respect to Vision. Biological Cybernetics, 32(4):211–216.

1995

[2] R. Jain, R. Kasturi, B. G. Schunck. Machine Vision. McGraw-Hill, New York, USA.

1996

[3] M. Dabkowska,W. S. Mokrzycki. Generowanie widoków 3W obiektów do systemu wizualnej identyfikacji. Prace IPI PAN, number 809.

1997

[4] M. Dabkowska, W. S. Mokrzycki: Multi-view models of convex polyhedron. Machine Graphics and Vision, 6(4):419-450.

2003

[5] M. Kowalczyk, W. S. Mokrzycki. Obtaining Complete 2.5D View Representation of Polyhedra Using Concept of Seedling Single-View Area. Comput. Vis. Image Underst., 91(3):280–301.

2004

[6] M. Frydler, W. S. Mokrzycki. New, Fast Algorithm of 3D Multiview Polyhedron Representation Generation on View Sphere with Perspective. Proc. Advanced Computer Systems CCV(ACV)'04, Prague, May 15-16.

2005

[7] M. Kowalczyk, W. Mokrzycki. Concept of 3D View Model of Non-Convex Polyhedra Generation Using View Sphere Partitioning into Single-View Areas. Computer Recognition Systems, Advances in Soft Computing vol. 30, 411–418.

2007

[8] W. S. Mokrzycki, A. Salamończyk. New Results in Generation of Accurate 3D Multiview Representation of Monotonous Polyhedrons. Computer Recognition Systems 2, Advances in Soft Computing vol. 45, 100–107.

[9] W. S. Mokrzycki, A. Salamończyk. Spiral Scanning of Faces to Obtain Complete 3D View Representation of Monotonous Polyhedra. Computer Recognition Systems 2, Advances in Soft Computing vol. 45, 171–178.

2010

[10] A. Salamonczyk, W. Mokrzycki. Generation of View Representation from View Points on Spiral Trajectory. Image Processing and Communications Challenges 2, Advances in Soft Computing vol. 84, 67–74.

2011

[11] E. Neuman. Inequalities and Bounds for Generalized Complete Elliptic Integrals. Journal of Mathematical Analysis and Applications, 373(1):203–213.

2012

[12] A. Salamonczyk. Generowanie modeli widokowych wielościanów monotonnych do identyfikacji wizualnej. Phd thesis. Institute of Fundamental Technological Research, Polish Academy of Sciences.