The symmetry-diagram as a tool of the pattern recognition

Z. Szakal, I. Zsoldos

Abstract
The base of this paper was a new method for the detection of exact and approximate reflective symmetries. The algorithm was based on the definition of the so called symmetry-parameter which is a rate of the symmetry, a number between 0 and 1 without a dimension and its value does not depend on geometrical measures. A so called symmetry-diagram in other name a shape-diagram was determined from the symmetry-parameters computed for various lines crossing the gravity centre and for points surrounding it. Beside the possibility of the symmetry recognition, the shape-diagram shows an individual shape property of the 2D figures, independently from geometrical measures. In this paper we show a process in which similar and approximately similar 2D figures are sorted out from a multitude of different figures independently from geometrical measures using the definition of the symmetry-diagram.

Keywords - symmetry-diagram, symmetry detection, pattern recognition

I. INTRODUCTION

The problem of symmetry detection has been extensively studied in numerous fields including visual perception, computer vision, robotics, computational geometry and reverse engineering. Early methods concentrated on finding perfect symmetries in 2D or 3D point sets [1], [2]. Since the restriction to exact symmetries limits the use of these methods for real-world objects, a method was introduced for computing approximate global symmetries in 3D point sets [3], but the complexity of the algorithm makes it impractical for large data sets. The notion of approximate symmetry was formalized by expressing symmetry as a continuous feature [4]. The examination of the correlation of the Gaussian image was proposed to recover global reflective and rotational symmetries [5]. A shape descriptor was introduced that concisely encodes global reflective and rotational symmetries [6], [7]. Different applications based on generalized complex moments [8], grouping feature points [9], [10], [11], isometric transforms [12], planar reflective symmetry transform [13] and generalized symmetry transform [14], [15] are used in image processing and mesh processing for detecting exact local and global reflection-symmetry and rotation- symmetry of 2D and 3D images.

The symmetries are often not exactly present, but only approximately present, due to measurement errors in the scanning process, and approximation and numerical errors in model reconstruction during reverse engineering [16]. Beside this, different CAD systems often use different tolerances [17], and what is symmetric in one CAD system may be only approximately symmetric in another. To solve these problems new algorithms based on the B-rep model were developed to find approximate rotational and translational symmetries of 3D forms built from simple geometric units [18] and complex 3D forms [19] in reverse engineering.
In the most prevalent methods of symmetry detection a number of pixels are aligned to the contour. The perpendicular bisectors of various pixel pairs are regarded as hypothetical symmetry axes. The exact and approximate symmetry axes are searched from the set of the perpendicular bisectors e.g. using a symmetry map created from the parameters of the perpendicular bisectors [18], [19], or optimizing the gradient orientations of pixel pairs [14], [15]. In this paper a new method is shown for the detection of exact and approximate reflective symmetry. The new method is worked out for 2D case based on the fact that the symmetry axes cross the gravity centre. Accordingly the hypothetical symmetry axes are the lines crossing the gravity centre and the exact and approximate symmetry axes are selected from the set of these lines. The searching algorithm is based on the definition of the so called symmetry-parameter which is a rate of the symmetry, a number between 0 and 1 without a dimension and its value does not depend on geometrical measures. A so called shape-diagram is determined from the symmetry-parameters computed for various lines crossing the gravity centre. The shape-diagram is applicable to find every exact and approximate symmetry axis.
Beside this the shape-diagram shows an individual property of the 2D figures, it is independent from geometrical measures but it is characteristic of the shape of the 2D figures, independently of the fact that the 2D figure is symmetric or not. In this paper we show a process in which similar 2D figures are sorted out from a multitude of different figures, independently from geometrical measures, using the definition of the shape-diagram.

II. THE ALGORITHM OF THE SYMMETRY DETECTION

The algorithm consists of several simple steps.
We have proceeded from the fact that the symmetry axes of a 2D figure cross the gravity centre because a symmetry axis divides the figure to two coincident parts therefore a

This work was supported by OTKA grants K 73776 in Hungary.
Zoltan Szakal, Szent Istvan University, Faculty of Mechanical Engineering, Pater K.u.1., 2103 Godollo, Hungary, szakal.zoltan@gek.szie.hu,
Ibolya Zsoldos, Szent Istvan University, Faculty of Mechanical Engineering, Pater K.u.1., 2103 Godollo, Hungary, zsoldos.ibolya@gek.szie.hu

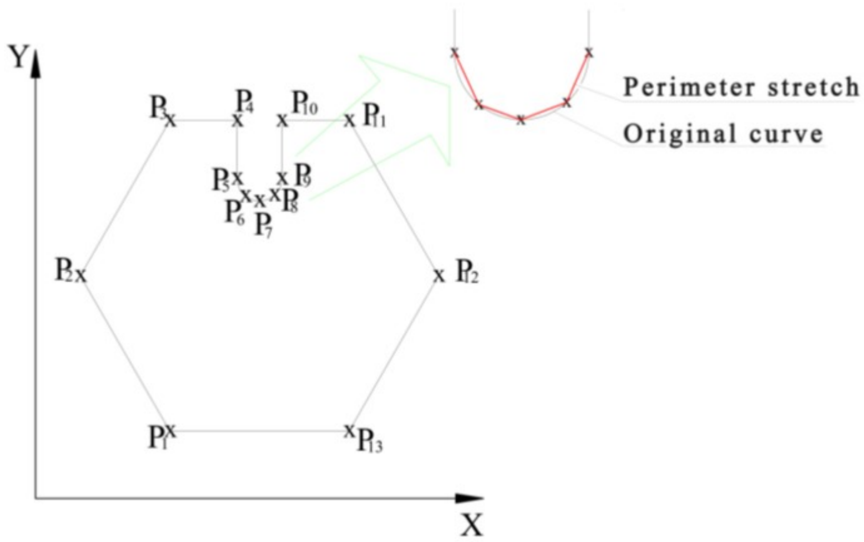

Figure 1 Collecting of points (P_i) on the boundary.

symmetry axis is a median, too. A median has to cross the gravity centre.
The algorithm scans the multitude of the lines crossing the gravity centre and studies how favourable these lines as symmetry axes are. If there is no exact symmetry axis crossing the gravity centre, a best approximate symmetry axis crossing an area surrounding the gravity centre is searched with similar method. Namely, since an exact symmetry axis crosses the gravity centre, an approximate symmetry axis has to cross an area surrounding the gravity centre.
In the followings we detail the steps of the algorithm.

Step 1: Collecting of the input data

The contour of the form is needed for the computation.
The input data set is defined as a set of points aligned to the contour. Such a point set can be created by the use of different tools of electronic image processing e.g. the MatLab. In order to decrease the measure of the point set file and the run time of the algorithm, the point set can be optimized: the points can be fixed rare or thickened depending on the complexity of the geometry. Fewer points are needed at greater curvature and more points are needed at smaller curvature, Figure 1. Such an optimization of the input point set is not required, but if we perform it, it can speed the run of the computer codes. The original contour is approximated with the stretches determined by these points. We name these stretches as perimeter stretches.
The input of the algorithm is the table of the points. The points follow each other in a clockwise direction. Such a table can be created manually and also recorded digitally from a pixel set.

Step 2: Determination of the gravity centre

We determine the gravity centre of the 2D figure with the method used in the geographic information systems. In order to compute the area of the 2D figure trapeziums are defined so that the points aligned to the contour are projected to the horizontal axis of the coordinate system, Figure 2.
The area of the i^{th} trapezium is:

$$A_i = \frac{(x_{i+1} - x_i)\cdot(y_{i+1} + y_i)}{2} \qquad (1)$$

where (x_i , y_i) and (x_{i+1} , y_{i+1}) are the end points of the i^{th} perimeter stretch. The points have to follow each other in a clockwise direction, in contrary case a negative value is derived for the area. The coordinate system is defined so that if the 2D figure is revolved around the gravity centre it always has to remain in the first plane quarter.
The area of the 2D figure is the sum of the areas of the trapeziums:

$$A = \sum_{i=1}^{n} \frac{(x_{i+1} - x_i)\cdot(y_{i+1} + y_i)}{2} \qquad (2)$$

where n is the number of the trapeziums.

The co-ordinates of the gravity centre are determined in general case by

$$x_g \cdot A = \int_A x \cdot dA \qquad (4)$$

$$y_g \cdot A = \int_A y \cdot dA \qquad (5)$$

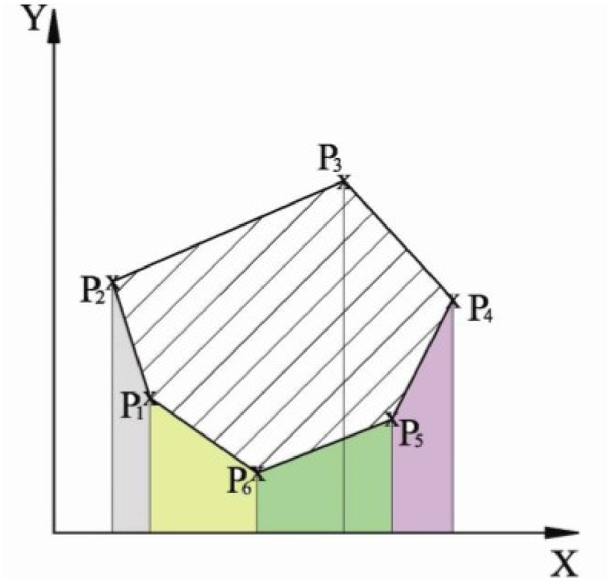

Figure 2 Definition of trapeziums for the computation of the area in the example of a hexagon.

In our case (because of the trapeziums) the integration reduces itself to a simple summing up:

$$x_g = \sum_{i=1}^{n} \left[(y_i - y_{i+1}) \cdot \frac{(x_i^2 + x_i \cdot x_{i+1} - x_{x+1}^2)}{6 \cdot A} \right] \qquad (5)$$

$$y_g = \sum_{i=1}^{n} \left[(x_{i+1} - x_i) \cdot \frac{(y_i^2 + y_i \cdot y_{i+1} - y_{x+1}^2)}{6 \cdot A} \right] \qquad (6)$$

Step 3: Scanning

The vertical line crossing the gravity centre is regarded as symmetry axis in the case of an optional starting orientation of the 2D figure, Figure 2. The algorithm analyses whether the 2D figure is symmetric for this hypothetical symmetry axis. For this purpose the algorithm scans the figure with horizontal lines (so called ‘measuring lines’) following each other with equal distances, Figure 3. The hypothetical symmetry axis is vertical and the measuring lines corresponding to the scanning levels are perpendicular to the axis namely they are horizontal.

In the course of the scanning the so called measuring stretches are determined. A measuring stretch is the distance between the vertical hypothetical symmetry axis and the intersection of a measuring line and a perimeter stretch. In Figure 4 ‘l’ and ‘r’ are the measuring stretches. In a horizontal scanning level there could be more measuring stretches, as well.

The complete area of the 2D figure is scanned with measuring lines proceeding from up to down, from the maximum (Y_{max}) to the minimum (Y_{min}) vertical coordinate of the points aligned to the contour. The accuracy of the method depends on the distance between the measuring lines: the shorter the distance is the more accurate the result is, however it affects the running time of the computer code as well.

Step 4: Computing the symmetry parameter

The symmetry parameter Z shows the deviation from the perfect symmetry numerically. This value between 0 and 1 can be a rate of the symmetry. In order to define it we introduce first a Z_k parameter for the k^{th} scanning level. In the simplest case when there is one measuring stretch on the left and on the right side of the hypothetical symmetry axis, Figure 4, Z_k can be computed simple:

$$Z_k = 1 - \frac{abs(l-r)}{L_{max}} \qquad (7)$$

where:
’l’ is the measuring stretch on the link side of the symmetry axis,
’r’ is the measuring stretch on the right side of the symmetry axis,
L_{max} is the maximum of the measuring stretches in the course of the scanning.

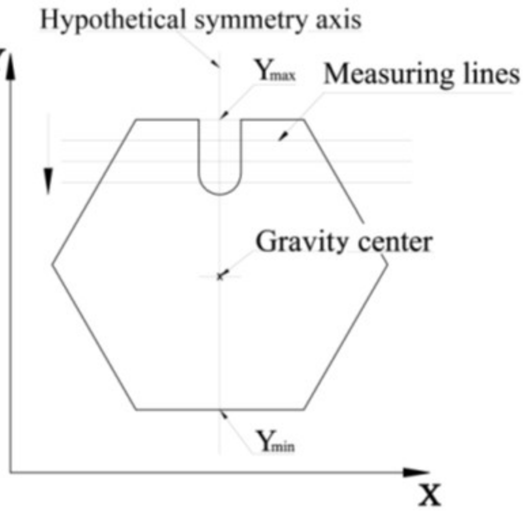

Figure 3 The horizontal measuring lines

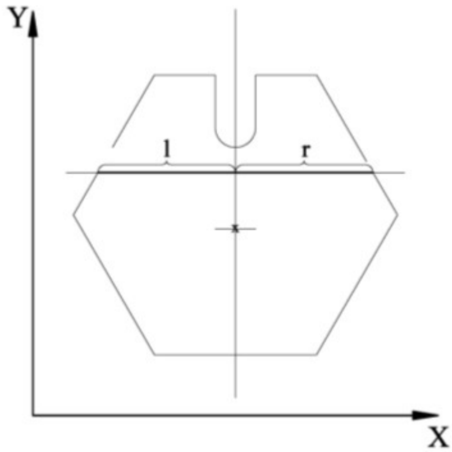

Figure 4 Definition of the measuring stretches in a simple case

Z_k is defined so that:

- $Z_k=1$ if $l=r$ when the k^{th} scanning level is symmetrical to the hypothetical symmetry axis,
- $Z_k=0$ if $l=0$ and $r>0$, or $r=0$ and $l>0$ when a measuring stretch does not have a pair on the other side of the hypothetical symmetry axis which is the worst case regarding the symmetry,
- $0<Z_k<1$ when $l>0$ and $r>0$ and the figure is not symmetrical to the hypothetical symmetry axis, and the closer this value is to 1 the better the k^{th} scanning level approximates the exact symmetry.

Since a ratio of lengths is used in (7) Z_k is a number without a dimension and its value does not depend on the measures.

If there are more than 2 measuring stretches in a scanning level (l_1, l_2, …, l_n on the left side and r_1, r_2, … , r_m on the right side of the hypothetical symmetry axis, Figure 5) then l_i and r_i are ordered into pairs and Z_k is defined according to (8). Let us assume that there are more measuring stretches on the left side: $n>m$. In this case the pairs have to be defined so that $r_i=0$ for every l_i $i=m+1, m+2, \dots, n$. In general case Z_k is computed as the average of the expression defined by (7) from the pairs:

$$Z_k = \frac{1}{n}\sum_{i=1}^{n}\left(1-\frac{abs(l_i-r_i)}{L_{\max}}\right) \tag{8}$$

where n is the number of the pairs in a scanning level.

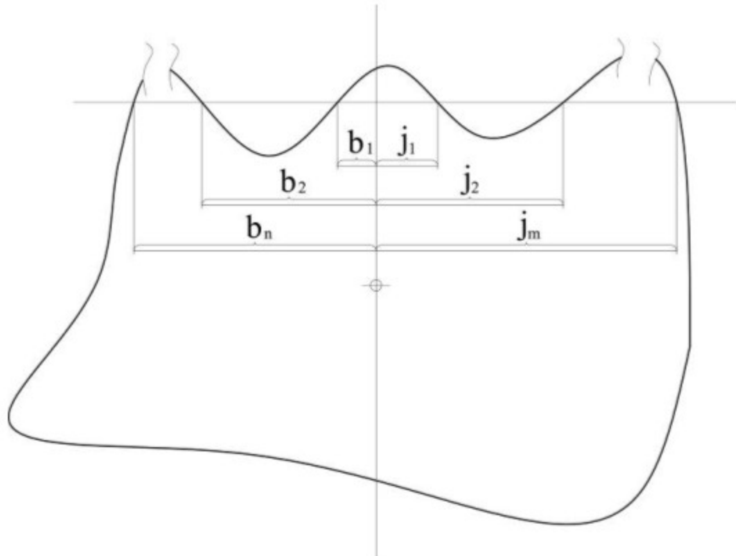

Figure 5 Definition of Z_k in general case

The symmetry-parameter Z is defined as the average of Z_k:

$$Z = \frac{\sum_{k=1}^{N} Z_k}{N} \quad (9)$$

where N is the number of the scanning levels (measuring lines) from Y_{max} to Y_{min}.

Since Z is defined with averages according to (7), (8) and (9), Z keeps the original properties of Z_k:

- the value of Z can change between 0 and 1,
- Z=1 is the case of an exact symmetry,
- Z=0 is the worst case regarding the symmetry (e.g. the full figure is on the left side of the axis),
- in the case of 0<Z<1 the closer the value is to 1 the better the figure approximates the exact symmetry,
- Z is a number without a dimension, it does not depend on the measures, it depends only on the shape.

Step 5: Rotation of the geometry

After the determination of the symmetry-parameter for the starting orientation the 2D figure is scanned by rotating the hypothetical symmetry axis, as well. The computation is simpler if the hypothetical symmetry axis is fixed vertically and the figure is rotated with small angles, step by step, Figure 6. The accuracy of the computation depends on the steps (angles) of the rotation. After every step of rotation the symmetry-parameter is computed.

The method scans the entire 2D figure in such a way that it rotates the examined figure by 180 degrees. Since the algorithm examines the figure both above and below the gravity center, the rotation by 180 degrees means a complete coverage.

Step 6: The symmetry-diagram in other word: the shape-diagram

The symmetry-parameter Z is computed for every rotational step. The values of Z are saved together with the angles of rotation. After the full (180^0) rotation a diagram is drawn where the independent variable is the angle of rotation and the dependent variable is Z. From this diagram the results can be evaluated. Let us name it shape-diagram. Exact symmetry exists where Z=1 and the closer Z is to 1 the better the figure approximates the exact symmetry at a given angle.

The shape-diagram of a square can be seen in Figure 7. The lines signed with numbers (1, 2, 3, 4) are the exact symmetry axes of the square and accordingly these correspond to the angles of rotation where Z=1 ($\alpha=0^0$ or 180^0, $\alpha=45^0$, $\alpha=90^0$ and $\alpha=135^0$).

The shape-diagram shows an individual property of the 2D figures. Since Z is independent of geometrical measures the shape-diagram is a shape parameter of the 2D figures (therefore the name is ‘shape-diagram’). The shape-diagram is the same for similar 2D figures having different measures. But if there is a smallest change on the shape of the 2D figure the shape-diagram changes, as well.

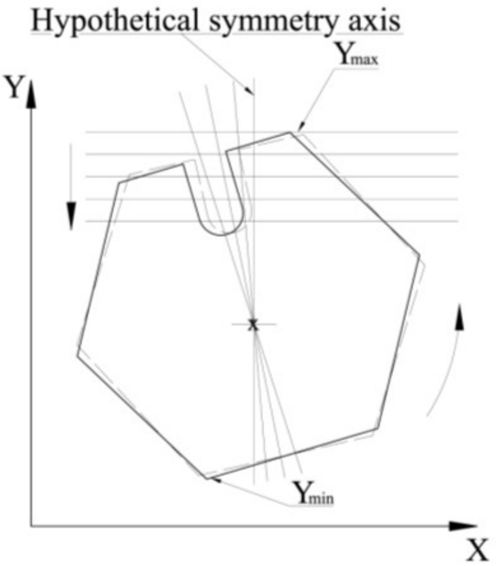

Figure 6 Rotation of the 2D figure with small angles

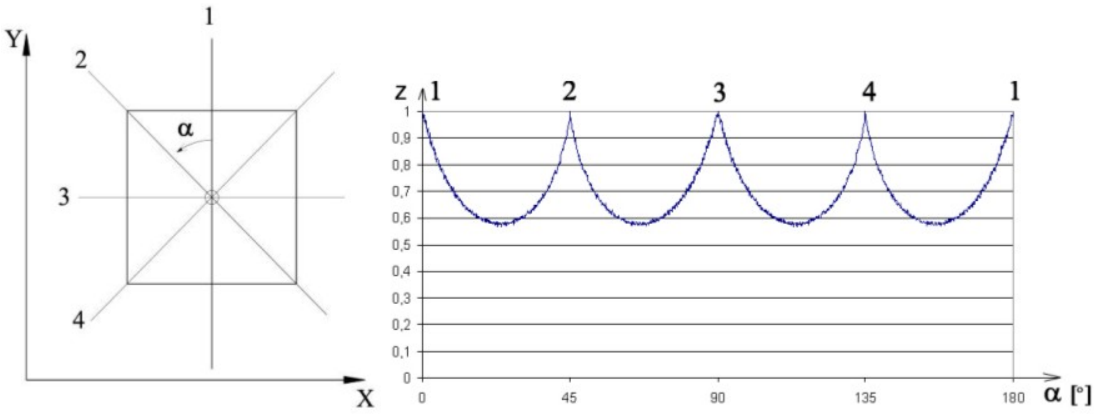

Figure 7 The definition of the shape-diagram

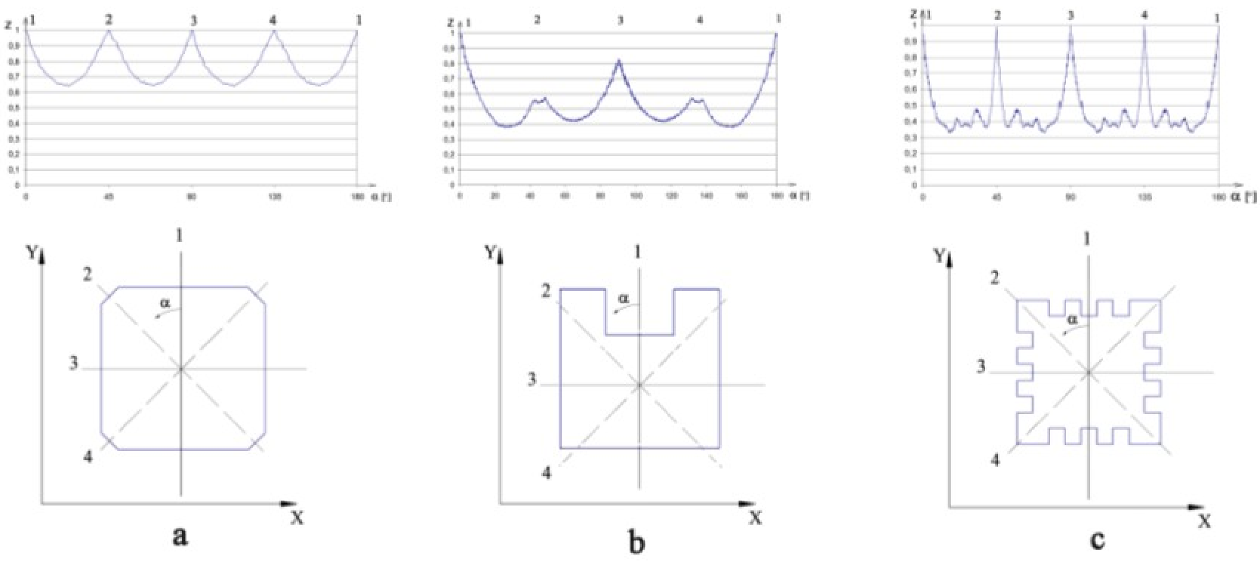

Figure 8 Shape-diagrams for modified squares

We show examples for the individuality of the shape-diagram in Figure 8. Three variously modified squares can be seen and the shape-diagrams are different, as well. In Figure 8a the
corners of the square are cut, the number and the place of the original symmetry axes have not changed but the minimum values have increased from Z=0.58 to Z=0.65. In Figure 8b a part of the square is cut out, only 1 symmetry axis has remained and a weakly approximate symmetry axis signed with number 3 can be found at a local maximum of the diagram. The other two local maximums signed with number 2 and 4 correspond to only very weakly approximate symmetry axes. In Figure 8c several small parts are cut out from the square, the exact symmetry axes have not changed but the curve between the maximums is basically other.

The algorithm does not have limits in 2D case. It is applicable even if the contour consists of several closed loops. In Figure 9 a circle with two holes can be seen.

It is obvious that the shape-diagram is the constant 1 function in the case of a circle because every line crossing the centre is an exact symmetry axis. Since there are two holes on the circle in Figure 9 accordingly there are only two exact symmetry axes signed with number 1 and 2 where Z=1.

There was an exact symmetry axis in every example shown above, i.e. Z=1 value(s) could be found in every shape-diagram computed for the lines crossing the gravity centre.
In the case when there is no exact symmetry axis, it is not sure, that the best approximate symmetry axis can be found in the lines crossing the gravity centre. However we can assume that the best approximate symmetry axis has to pass near by the gravity centre in the case of approximate symmetrical 2D forms. In this case it is advisable to repeat the Steps 3-5 so that the shape-diagram is created for the points surrounding the gravity centre, as well. For the best approximate symmetry axis we have to search the shape-diagram where the maximum Z value can be found.

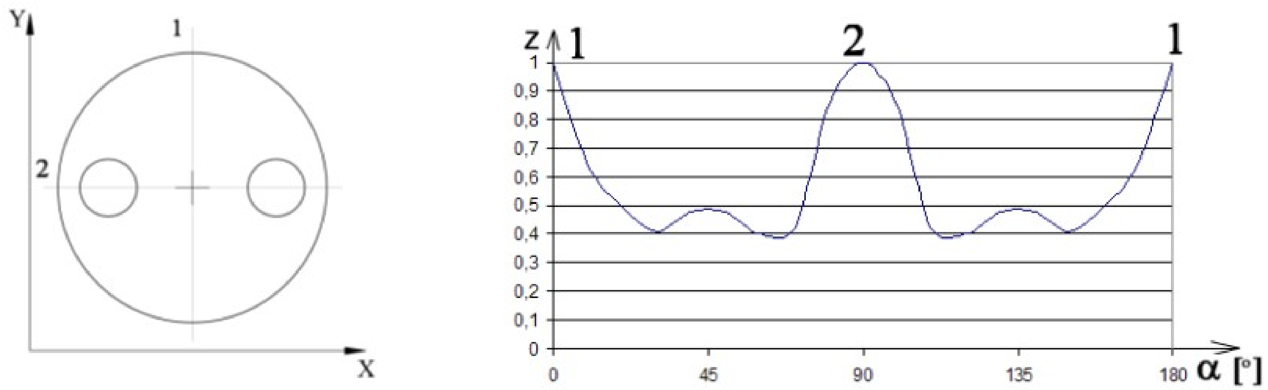

Figure 9 Example of a 2D figure having a contour with several closed loops

III. ALGORITHM FOR SORTING OUT SIMILAR AND APPROXIMATELY SIMILAR FORMS

In numerous studies the tool of the pattern recognition is a neural network algorithm, [20-23]. In this chapter we show that the symmetry-diagram can be used as tool of the pattern recognition independently of geometrical measures.
In this stage of this research work the sorting algorithm is applicable for the case when there are no overlaps between the 2D figures in the multitude.
In the followings we detail the steps of the algorithm parallel with a case study shown in Figure 10 and Figure 11. As an example for the input of the sorting algorithm a multitude of different 2D figures are shown in Figure 10. 12 independent 2D figures can be seen. Similar figures having different size can be seen. Several figures are deformed a bit and in these cases the broken lines sign the figure which would be geometrically similar to other figures. For example the figures signed by numbers 1 and 8 are geometrically similar and the figure signed by number 11 is only approximately similar to them. The geometrical similarity would be exact in the case of the contour signed with broken lines.

Step 1: Collecting of the input data

The input data set is defined as a set of points aligned to the contours of the 2D figures in the multitude. Such a point set can be created by the use of different tools of electronic image processing e.g. the MatLab. The point set consists of the x, y coordinates of the points scanned along the contours.

Step 2: Sorting of the input data

The subsets of the points are sorted out for the independent 2D figures which are not overlapped. In our example 12 subsets of the contour points could be separated, accordingly we signed by number 1-12 the separated figures corresponding to the 12 subset in Figure 10.

Step 3: Computation of the shape-diagrams

In this step the algorithm computes the shape-diagram for every subset. In our example 12 shape-diagram are computed which are shown in Figure 11.

Step 4: Sorting of the shape diagrams

Geometrically similar 2D figures have the same shape-diagram. The identical diagrams can be easily separated.

The only problem in this case is that the shape-diagrams are drawn in different period. In order to avoid this problem by the comparison of two diagrams the orientation of $\alpha=0^0$ is chosen at the place of the maximum in every shape diagram.

Allowing a small threshold value for the differences between the diagrams, approximately similar figures can be sorted out, as well.

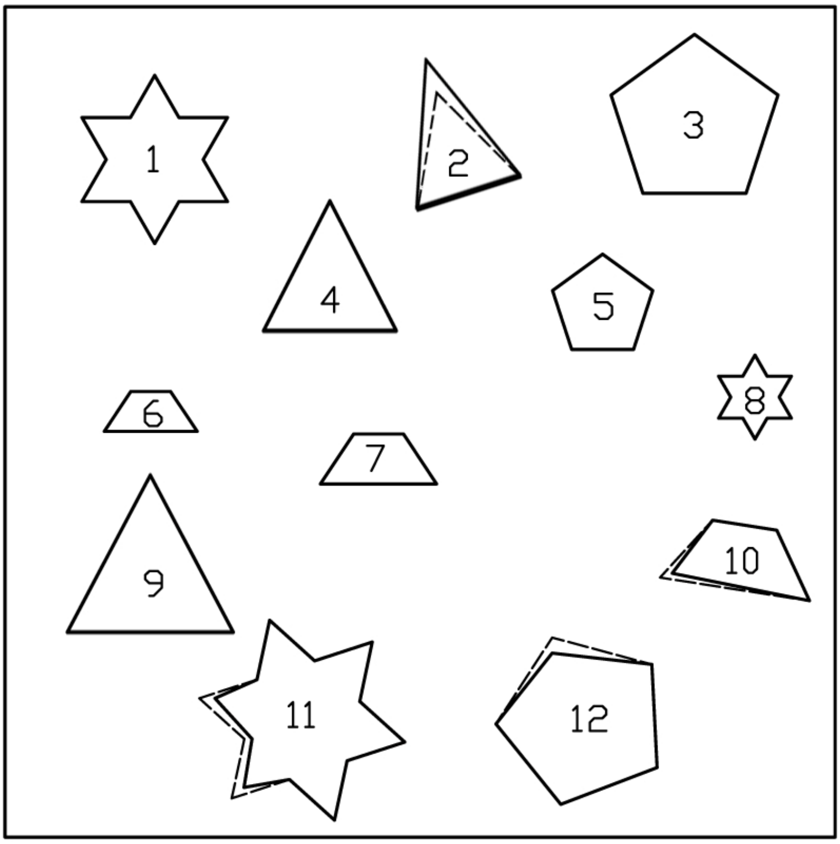

Figure 10 An example for the multitude of 2D figures

In our example four groups of the shape-diagrams could be separated, Figure 11. Three-three equal or almost equal curves could be sorted out in the four groups. In Figure 11 the equal (drawn with green and black colours) or almost equal curves (drawn with red colour) are drawn in the same co-ordinate system. Near the diagrams, on the right side the numbers of the 2D figures having equal or almost equal shape-diagram are represented. The shape diagrams of the geometrical similar stars, pentagons, isosceles triangles and isosceles trapezium can be distinguished unambiguously. There is only small differences between the shape diagrams of the approximately similar figures.
Using the shape diagrams the similar or almost similar 2D figures could be separated.

IV. SUMMARY

The new algorithm of the symmetry detection gives a new possibility to detect exact and approximate symmetry axes of 2D figures. Beside the known methods of symmetry detection the new algorithm defines the so called symmetry-parameter which can be a rate for the symmetry using a number between 0 and 1.
The symmetry diagram created from the values of the symmetry-parameter computed for various lines crossing the gravity centre is independent from geometrical measures and it shows an individual, characteristic shape property of the 2D figures.

The process of sorting of geometrically similar or almost similar 2D figures was shown on an example. The sorting algorithm is based on the use of the shape-diagram of the 2D figures. The algorithm works in the case of the multitudes of 2D figures where there is no overlapping between the contours of the figures, in this stage of this research work.

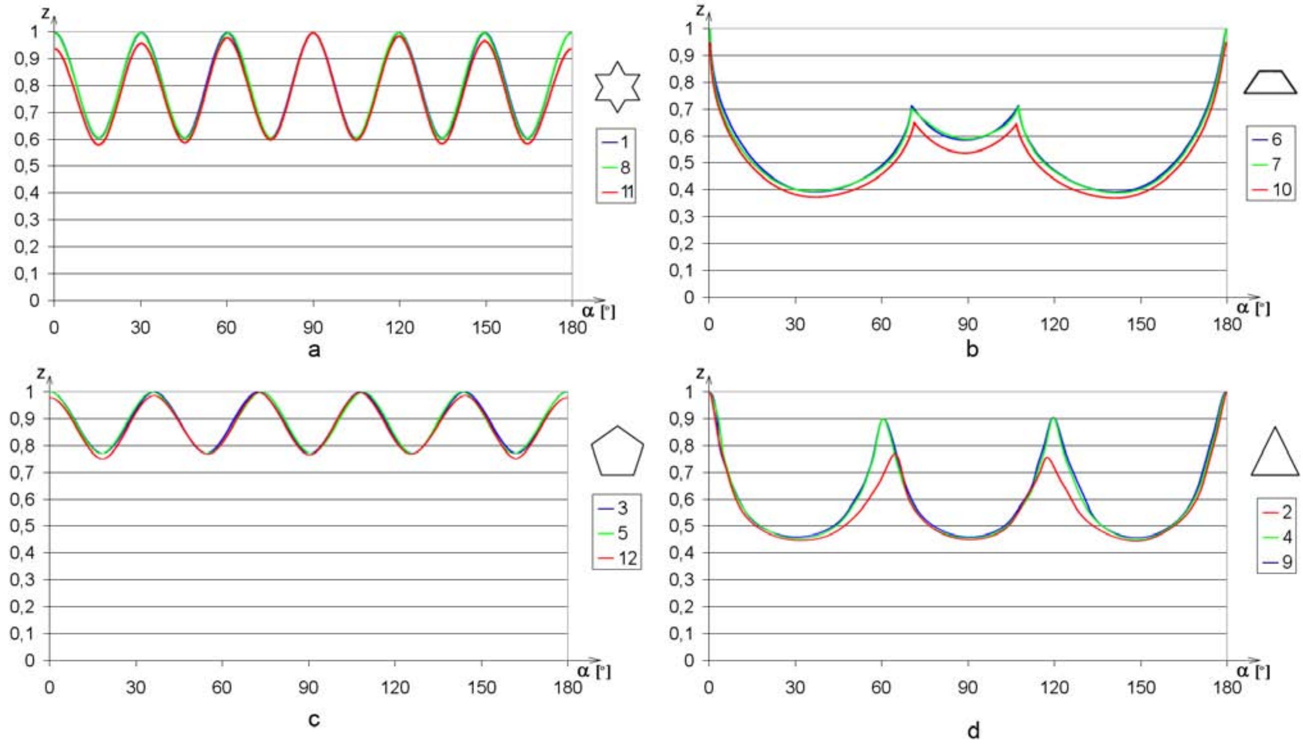

Figure 11 The sorted shape-diagrams of the 2D figures

ACKNOWLEDGMENTS

This work was supported by OTKA grants K 73776 in Hungary.

REFERENCES

[1] J. WOLTER, T. WOO, R. VOLZ, Optimal algorithms for symmetry detection in two and three dimensions. The Visual Computer, 1985.

[2] M. ATALLAH, On symmetry detection. IEEE Trans. On Computers 34, 7, 2004, pp. 663–666.

[3] H. ALT, K. MEHLHORN, H. WAGENER, E. WELZL, Congruence, Similarity and symmetries of geometric objects. Discrete Comput. Geom. 3, 1988, pp. 237–256.

[4] H. ZABRODSKY, S. PELEG, D. AVNIR, Symmetry as a continuous feature. IEEE PAMI, 17, 1995.

[5] C.Sun, J. Sherrah, 3D symmetry detection using the extended Gaussian image. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19, 2, 1997, pp. 164–8.

[6] M.KAZHDAN, B. CHAZELLE, D.P. DOBKIN, A. FINKELSTEIN, T. FUNKHOUSER, A reflective symmetry descriptor. In Proceedings of ECCV, 2002, pp. 642–656.

[7] M.KAZHDAN, T. FUNKHOUSER, S. RUSINKIEWICZ, Symmetry descriptors and 3d shape matching. In Symposium on Geometry Processing, 2004, pp. 116–125.

[8] D. Shen, H. Ip, K. Cheung, E. Teoh, Symmetry detection by generalized complex (GC) moments: A close-form solution. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21, 5, 1999, pp. 466–76.

[9] G. Loy, J. Eklundh, Detecting symmetry and symmetric constellations of features. In: ECCV 2006, Part II. LNCS, Vol. 3952. Springer-Verlag; 2006, pp. 508–21.

[10] N. Mitra, L. Guibas, M. Pauly, Partial and approximate symmetry detection for 3D geometry. ACM Transactions on Graphics, 25, 3, 2006, pp. 560–8.

[11] N. Mitra, L. Guibas L, M. Pauly, Symmetrization. ACM Transactions on Graphics, 26, 3, 2007, Art. No. 63

[12] A. Martinet, C. Soler, N. Holzschuch, FX. Sillion, Accurate detection of symmetries in 3D shapes. ACM Transactions on Graphics, 25, 2, 2006, pp. 439–64.

[13] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, T. Funkhouser, A planar-reflective symmetry transform for 3D shapes. ACM Transactions on Graphics 25, 3, 2006, pp. 549–59.

[14] D. Reisfeld, H. Wolfson, Y. Yeshurun, Context Free Attentional Operators: the Generalized Symmetry Transform, Int. J. Comput. Vision 14, 2, 1995, pp. 119-130.

[15] W.H. Li, A.M. Zhang, L. Kleeman, Global reflectional symmetry detection for robotic grasping and visual tracking, ACRA 05, Australasian Conference on Robotics and Automation, 2005. http://www.visionbib.com/bibliography/journal/acr.html#ACRA05

[16] T. Varady, R. Martin, J. Cox, Reverse engineering of geometric models - an introduction, Computer-Aided Design 29, 4, 1997, pp. 255–68.

[17] M. Pratt, B. Anderson, T. Ranger, Towards the standardized exchange of parameterized feature-based CAD models. Computer-Aided Design 37, 12, 2005, pp. 1251–65.
[18] B. Mills, F. Langbein, A. Marshall, R. Martin, Approximate symmetry detection for reverse engineering. In: Proc. 6th ACM symp. solid and physical modeling, 2001, pp. 241–8.
[19] M. Li, F. Langbein, R. Martin, Detecting approximate symmetries of discrete point subsets, Computer-Aided Design 40, 1, 2008, pp. 76-93.
[20] V. Niola, G. Quaremba. Pattern recognition and feature extraction: a comparative study. Proc. of the 6th WSEAS Int. Conf. on Neural Networks, Lisbon, Portugal, 2005, pp. 109-114.
[21] Kyunghee Lee, Segmentation and recognition of hand-written digits using OSSA neural network. Proc. of the 7th WSEAS Int. Conf. on Applied Computer& Applied Computational Science, Hangzhou, China, 2008, pp. 282-287.
[22] A. Perez-Mendez, E. Torres-Rivas, F. Rivas-Echeverria, R. Maldonado-Rodriguez, A methological approach for pattern recognition system using discriminant analysis and artificial neural network. Proc. of the 6th WSEAS Int. Conf. on Neural Networks, Lisbon, Portugal, 2005, pp. 244-249.
[23] H. K. Sulehria, Y. Zhang, Vehicle logo recognition using mathematical morphology. Proc. of the 6th WSEAS Int. Conf. on Telecommunications and Informatics, Dallas, Texas, USA, 2007, pp. 95-98.