Intermediate Representation in Model Based Recognition Using Straight Line and Ellipsoidal Arc Primitives

Sergiu Nedevschi, Tiberiu Marita, Daniela Puiu
Technical University of Cluj-Napoca, ROMANIA
Sergiu.Nedevschi@cs.utcluj.ro

Abstract

An intermediate representation suitable for the 2-D recognition of the 3-D objects, from a single intensity image is proposed. Determination of the intermediate representation from CAD models and from intensity images is presented. This representation uses as primitives line segments and ellipsoidal arcs. A complete technique for fitting contour lines with these primitives has been developed. The method has been proved very accurate and robust to noise, thus it is suited for a variety of applications such as matching and recognition of objects in real life images.
Index terms *– 2-D recognition, 3-D objects, intermediate representation, feature extraction, straight-line segment primitives, ellipsoidal arc primitives.*

1. Introduction

In model based recognition it is necessary to define a representation easy obtainable both from the image and the model. Such a representation is called "intermediate representation". The intermediate representation category includes the symbolic scene description [1], sensor-tuned representation [2], structural hierarchical description [3].

The symbolic scene description representation is based on features that can be immediately detected from the image. The sensor-tuned representation can also include more other significant features of the object or scene, especially geometrical features, which are not immediately detectable from the sensorial data. The structural hierarchic description is based on both features that can be immediately detected from the image and those resulted from a feature grouping process. While the features detection is a bottom-up process, the features grouping is a top down one. It makes assumptions about the scene geometry and gives more abstract feature hierarchies that quicken the matching process.

This paper presents an intermediate representation for 2-D model based recognition of a class of 3-D objects from a single intensity image. Most complex manufactured objects can be described as an aggregation of simple object types, like parallelepiped, cylinder, cone, polyeder. Therefore, these elementary objects and their parts generated by cuttings are taken into account in this work. In addition, we assume that the objects are bordered by planar, cylindrical or conical surfaces.

By consequence the primitives of the proposed intermediate representation are the 2-D straight-line segment and the ellipsoidal arc. These primitives and the intermediate representation based on them are obtained from the geometric model of the objects by edge projection and geometric inferences, and from the intensity image by edge extraction, segmentation, and a domain specific grouping process.

Many methods using polygonal approximation of the contour lines have been proposed [4,5]. This approximation of contour lines is simple but rarely effective in the case of cylindrical or conical parts. On the other hand, the approximation of contours with ellipses and circles remains open [6,7,8], especially when both precision and robustness to noise are required. The problems raised by the approximation of contours with ellipsoidal arcs are the detection of the contour regions corresponding to ellipsoidal arcs and the accurate detection of the arc parameters.

The Hough based methods do not require pre-segmentations of the image and are able to detect both the ellipsoidal arc's position and parameters. They are robust to noise and occlusions but are computationally expensive. The conic fitting techniques center on finding the set of parameters that minimize some distance measure between the data points and the conic. Their unsolved problem is the right estimation of the contour points involved in the fitting operation.

A combined method for contour approximation by straight lines and ellipsoidal arcs is proposed. A straight-line segmentation, based on corner points, is used for contour splitting in linear, convex or concave regions. A locally applied Hough transform is used to extract possible ellipsoidal arcs from these regions. This way the method becomes computationally efficient, and in the same time inherits only the advantages of the Hough based methods.

0-7695-1183-X/01 $10.00 © 2001 IEEE

2. Intermediate representation definition

The specification of the intermediate representation is based on the identification of some 2-D features and relations among these features that can be associated with 3-D shapes. The features corresponding to projection invariant properties of the 3-D shapes will be referred as index features. The other features will be referred as basic features. The primitives themselves define elementary features and by associating them based on relations of adjacency, parallelism, collinearity, and symmetry compound features are generated.

Table 1 shows the elementary objects with their 3-D index feature generator elements, and table 2 shows the 2-D index features generated by the 3-D index feature generator elements. The quantitative measure of the basic and index features is given by a set of specific identification and localization attributes (Tables 3 and 4).

A local coordinate system is associated with each feature. The position of a feature in the global coordinate system can be computed from the relation between the local coordinate system and the global one. In order to allow subsequent refinements, both index and basic features are kept in the intermediate representation.

Table 1. Elementary objects

Elementary object	3-D index feature generator element
Parallelepiped:	3-D parallelogram, vertex
Cylinder:	3-D circle, cylindrical surface, vertex
Cone:	3-D circle, conical surface, vertex
Polyeder:	3-D polygon, vertex

Table 2. 2-D index features

3-D index feature generator element	2-D index feature
3-D circle:	Circle, ellipse, circular&ellipsoidal arc
Cylindrical surface:	Parallel lines
Conical surface:	2 adjacent lines
3-D parallelogram:	2-D parallelogram
3-D polygon:	2-D polygon
Vertex:	Junction, angle

Table 3. Basic features

Basic feature	Attributes	
	Identification	Localization
Straight line segment (sline):	Length, medium gradient, end points coordinates, significance factor	Center coordinates, direction
Collinear lines:	Slines, lengths, distance, significance factor	Center coordinates, direction
Polygonal line:	Type, slines list, significance factor	

Table 4. Index feature

Index feature	Attributes	
	Identification	Localization
Parallel:	Slines, distance, median length, Slines length, significance factor	Center coordinates, direction
Angle:	Slines, angle, slines length, significance factor	Center coord., bisector dir.
Junction:	Slines, slines length, angles, significance factor	Center coordinates, direction
Parallelogram:	Parallels, significance factor	Center coordinate, direction
Ellipsoidal arc (e_arc) :	Axis lengths, begin and end ray angles, significance factor	Center coord., major axis dir.
E_arc-sline adjacency:	Sline, e_arc, angle, e_arc length, sline length, significance factor	Center coord., bisector dir.

The symbolic format proposed for the representation of the 2-D features allows explicit access to the attributes. The formats for the straight-line segment, ellipsoidal arc, and ellipsoidal arc straight-line adjacency are:

slinie (sline_Id, x_c, y_c, dir, lenght, gradient, x_b, y_b, x_e, y_e, significance_factor) (1)

e_arc (e_arc_Id, x_c, y_c, maj_axis_dir, min_axis_length, maj_axis_length, beg_ray_angle, end_ray_angle, significance_factor) (2)

e_arc_sline_adjacency (adj_Id, e_arc_Id, e_arc_end, sline_Id, sline_end, x_c, y_c, bisector_dir, e_arc_tang_dir, sline_dir, angle, e_arc_length,sline_length,significance_factor) (3)

The intermediate representation has the form of an attributed relational graph, with elementary features in nodes and compound features defining relational constraints between elementary features as arcs. To support the matching process, useful information about the features discrimination power is provided through significance factors [17].

3. Intermediate representation extraction from the model

The 3-D objects models are generated by a modeling system using the Boundary Representation scheme (B-Rep). The generation of the intermediate representation of an object has two steps [16]: the computation of the orthogonal or perspective projection along a given direction, extraction of features and computation of attributes.

After projecting the object along a direction, a hidden line analysis algorithm is employed to eliminate the hidden edges and surfaces. As a result, a set of lines is obtained, representing visible edge projections. The lines are depicted by the following relation:

Line (ModelId, EdgeId, SegmentId, LineId, (LineAttributes)) (4)
where *LineId* specifies the projected line identifier.

By associating the projections corresponding to the segments, belonging to the same face, closed contours (C_c) or closed polygonal lines are obtained, where:

$$C_c = \{ l \mid \forall l_i, l_j \in C_c, l_i \neq l_j, l_i.IdSeg.IdFace = l_j.IdSeg.IdFace \} \qquad (5)$$

From the closed contours corresponding to planar faces, features as parallelogram, angle or parallel can be extracted. From closed contours corresponding to cylindrical or conical faces features as parallel, angle, ellipsoidal arc can be extracted. For all extracted features, the corresponding attributes are computed.

4. Intermediate representation extraction from the intensity image

The extraction of the intermediate representation from the intensity image involves the following stages: edge detection followed by contour extraction, contour approximation by straight lines and ellipsoidal arcs, identification of the compound features, and computation of their attributes.

4.1. Edge detection and contour extraction

To detect the edges a robust and accurate zero crossing algorithm working at subpixel accuracy is used [17]. It has the following steps: filtering with a median and a Gaussian filter in order to eliminate the noise and to smooth the surfaces; the computation of the second order directional derivative along the gradient direction; determination of the zero crossings of the second order directional derivative with subpixel accuracy; contour extraction as a list of points; and contour closing.

4.2. Contour approximation by straight lines and ellipsoidal arcs

An algorithm for contour segmentation by straight lines and ellipsoidal arcs based on corner points and resulting segments classification in straight or curved is used [12,13].

The steps of the contour approximation method are the following: corners detection; contour approximation by straight lines; identification of the linear, convex and concave regions of the contour; finding the ellipsoidal arcs in the convex and concave regions of the contour.

Contour filtering. The approximation of the contour by straight lines and ellipsoidal arcs could be affected by the existing noise in the contour lines. If the level of noise is to high, the straight lines and the ellipsoidal arcs will not be accurately detected on the contour. The purpose of the filtering is to smooth the contour lines. Because a digital image is a square grid of pixels and because the edge detection algorithm from the first step works at subpixel accuracy, the detected contour points are not equidistant (fig. 1). The distance between two adjacent points varies between 0 and √2 pixels in length.

In order to correctly apply a gaussian filtering on the contour an equidistant set of contour points is needed. Therefore, the initial set of points is transformed in an equidistant set of points, in which each point is at one pixel distance from each other.

Curvature computation and selection of corner points. The curvature *C* in a point *P(i)* can be approximated using the parametric functions *x(s), y(s)* and their first and second order derivatives [9,10] as follows:

$$C = \dot{x}\ddot{y} - \dot{y}\ddot{x} \qquad (6)$$

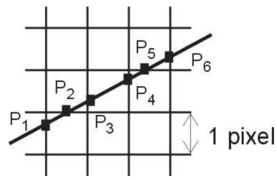


Figure 1. Contour points

A method that is robust to noise was selected to compute the derivatives. Instead of computing the first and second order derivatives using finite differences, the analytical expressions of the first and second order derivatives are deduced. The deduction is based on the interpolation of the current point *P(i)* of the contour and its neighbors *P(i-1)* and *P(i+1)* by a second order polynomial function passing through them, expressed by the following two equations:

$$x(s) = a_2 s^2 + a_1 s + a_0 , \qquad y(s) = b_2 s^2 + b_1 s + b_0 \qquad (7)$$

Replacing s with $i-1$, i and $i+1$ for the three successive contour points considered, two equation systems are obtained. Solving these equation systems, the coefficients a_2, a_1, a_0, b_2, b_1, b_0 are deduced. The first and second order derivatives for $x(s)$ and y(s) are obtained deriving the above equations.

The next operation is the selection of the corner points (the points with highest absolute local curvature value). This is done using a threshold value determined experimentally.

Contour approximation by straight-line segments. At this stage, the contour line is split by the corner points in regions. Next step is the region approximation by straight-line segments. A method based on the computation of the axis of least inertia of the sets of points of each region was chosen. If the approximation error is smaller than a specified threshold, a segment of the axis of least inertia is used as approximation of the region. The two ends of the segment are given by the projections of the end points of the corresponding contour region on axis. If the approximation error is higher than a specified threshold, the current region is split in two or more sub-regions until approximation error satisfies the threshold.

Finding concavities and convexities of the contour line. The concavity and convexity of contour line regions can be detected easily considering pairs of two consecutive straight-line segments obtained in the previous stage. These pairs of segments are rotated clockwise and translated, considering the first point (P_1) the base point (fig.2), until P_1 reaches the origin of the axis, and the last one (P_3) reaches the x axis. If $P'_2.y < -threshold$, then the region is concave, and the angle between segments P_1P_2 and P_2P_3 is considered negative. If $P'_2.y > threshold$, the region is convex, and the angle between segments P_1P_2 and P_2P_3 is considered positive.

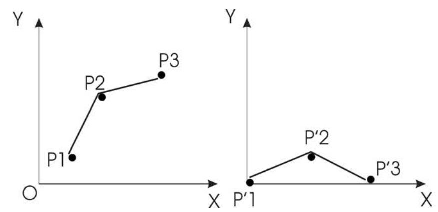


Figure 2. Geometric transforms for convexity / concavity detection

If $|P'_2.y| <= threshold$, then the convexity or concavity of region is uncertain (affected by noise or quasi-linear), and the angle between segments P_1P_2 and P_2P_3 is considered null. The threshold value is positive, and it is chosen according to the noise level on the contour line. This approach is more accurate than the analyses of the contour line itself because the use of the straight-line segments diminishes the noise influence.

Finding ellipsoidal arcs on the contour lines. At this stage, the contour line is approximated by a set of straight line-segments, and the concave and convex regions are detected. For each concave or convex contour region approximated by a set of consecutive straight-line segments P_1P_2, P_2P_3, … $P_{n-1}P_n$, the maximum length ellipsoidal arc among these segments is sought. The ellipse approximation algorithm used begins with the analyses of the entire set of point $P_{1\,\ldots}\,P_n$. If the identification of the ellipsoidal arc does not succeed on the entire set, the ellipse seeking process continues on $P_{2\,\ldots}\,P_n$, … , $P_{n-1\,\ldots}\,P_n$, $P_{1\,\ldots}\,P_{n-1}$, … , $P_{1\,\ldots}\,P_2$. If an ellipsoidal arc is found between points $P_{i\,\ldots}\,P_j$, the ellipse searching continues between points $P_{1\,\ldots}\,P_i$, and then between points $P_{j\,\ldots}\,P_n$. At the end of the searching process, each set of straight-line segments that could be approximated by ellipsoidal arcs is replaced by an ellipsoidal arc.

The process that checks out if a contour region $P_{i\,\ldots}\,P_j$ is an ellipsoidal arc consists of three steps: verification of the ellipsoidal arc existence conditions, computation of the ellipsoidal arc parameters, validation of the results.

The main condition, which an ellipsoidal arc has to satisfy, is that the centers of a set of parallel chords crossing the contour region must be collinear [11]. Based on this idea, the elongation directions of the centers of three sets of parallel chords crossing the contour region are computed. If the approximation error of these three axes is smaller than a threshold, the intersection points of these three axes are determined. The triangle given by these three intersection points delimits a set of candidate center points for the ellipsoidal arc.

The computation of the ellipsoidal arc's parameters is done using an optimized Hough transform. The Hough transform [14,15] is applied only on the set of contour points delimited between $P_{i\,\ldots}\,P_j$.. The ellipse parameters (major axis length A, the minor axis length B and the major axis orientation θ) are computed for each set of candidate center points (x_c , y_c) detected in the previous step. One or more candidate ellipsoidal arcs are obtained.

The last step is the validation of the candidate ellipsoidal arcs. This is done using two error metrics: the medium deviation and the maximum deviation. The medium deviation is the average distance between the contour points and the corresponding ellipse points measured on an axial direction. The maximum deviation is the greatest value of these distances. The ellipsoidal arc that is chosen corresponds to the smallest error values.

If all candidate ellipsoidal arcs that were found have bigger deviations then a specified threshold, then an ellipsoidal arc cannot approximate the contour region. The ellipsoidal arc detection algorithm is resumed on a subset of $P_{i\,\ldots}\,P_j$.

Refinement of the results. The first step consists of a better approximation of the ends of the ellipsoidal arc. An algorithm, which tries to extend the ends of the ellipsoidal arcs on their adjacent line-segments, is used. The algorithm splits the adjacent straight-line segment in sub-segments and finds the maximum length sub-segment that approximates better the ellipsoidal arc.

Next step is to find the intersection between the ellipsoidal arc and their adjacent line-segments. If the distance between the intersection point and the end of the line-segment is smaller then a threshold, then the intersection point is considered as the new endpoint of the line segment, and an ellipsoidal arc–straight line adjacency compound feature is created.

Finally, beginning and end ray angles are calculated for each detected ellipsoidal arc.

4.3. Identification of the compound features and their attributes computation

As a result of the contour interpolation the elementary features (basic or index) are obtained (line segments, polygonal lines and polygonal contours, ellipsoidal arcs, ellipses, circles).

In the third step the extraction of the features like angle, junction, parallel and parallelogram is performed together with their attributes computation. This is done according to the methods presented in [17].

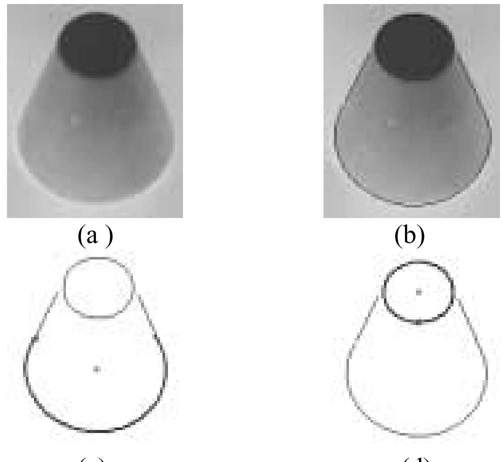
5. Results

In figure 3, the results of the segmentation of a simple conical object in a noisy image with a signal to noise ratio of 8.164 are shown. The generated intermediate representation consists of: two straight-line basic features, two ellipsoidal arc index features and two compound features corresponding to ellipsoidal arc – sline adjacency.

To accurately compute the features of interest the following input parameters were considered:
σ (gaussian parameter) =1, minimum gradient = 4, minimum contour length= 5, ε (approximation error) = 4, σ_c (contour gaussian parameter) = 0.6.

The features extracted are presented using the symbolic format proposed in (1), (2) and (3):

sline(1,161.2, 88.1, 113.7, 41.4, 7,152.9, 69.1, 169.5, 107, 0.004)
sline(2, 69.4, 87.7,-115.5,47.1,10, 59.2, 9.0, 79.5, 66.5,0.007)
e_arc (1, 14.74, 138.53, 4, 58, 64, -143.46, -26.21, 0.2)
e_arc (2, 16.67, 63.00, 0, 28, 32, 96.79, 91.71, 0.1)
e_arc_sline_adjacency (1, 1, B, 1, E, 169.0, 107.3,
-160.4, 121.95, 113.71, 171.77, 256.2, 41.4, 0.768)
e_arc_sline_adjacency (2, 1, E, 2, B, 59.2, 105.0,
-23.12, 59.68, -115.52, -175.19, 256.2, 47.1, 0.860)

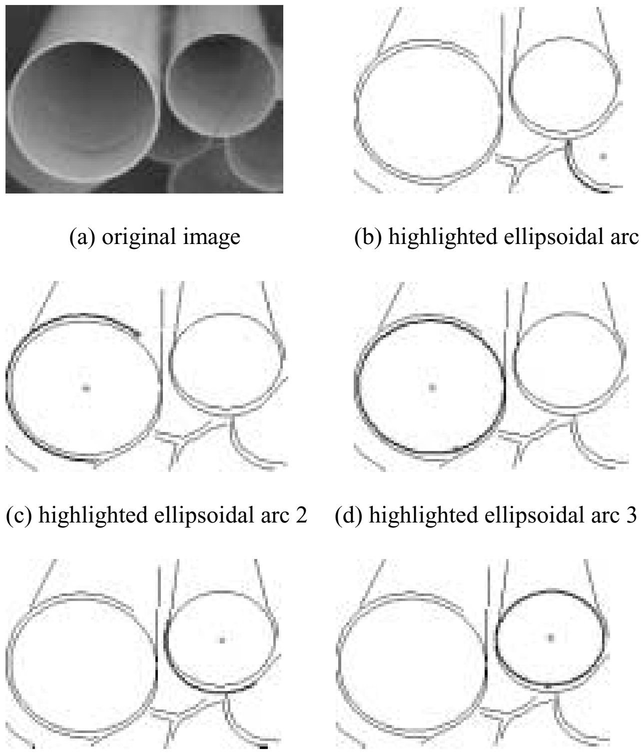


(a) (b)

(c) (d)

Fig. 3. Segmentation of a conical object: (a) Original image. (b) Original image plus extracted features. (c,d) Found ellipsoidal arcs

In figure 4, the results of the segmentation of some industrial objects are shown. The same input parameters were considered, as in the previous case.



(a) original image (b) highlighted ellipsoidal arc

(c) highlighted ellipsoidal arc 2 (d) highlighted ellipsoidal arc 3

(e) highlighted ellipsoidal arc 4 (d) highlighted ellipsoidal arc 5

Figure 4. Extraction of ellipsoidal arc primitives from an industrial object.

The ellipsoidal primitives extracted are presented using the symbolic format (2):

```
e_arc ( 1, 203.12, 155.26,  44, 24, 28,  40.65,  153.72, 0.0)
e_arc ( 2,  77.96, 123.22,  20, 52, 56,  66.79, -111.34, 0.1)
e_arc ( 3,  77.31, 123.03,   8, 48, 52,  65.02,   63.55, 0.2)
e_arc ( 4, 175.23, 105.06, 172, 38, 40, -62.94,   28.17, 0.0)
e_arc ( 5,  74.65, 103.81, 176, 34, 38, -99.03,  -96.71, 0.1)
```

6. Conclusions

The aim of this paper was to present the definition and the implementation of an intermediate representation for 2D and 3D objects recognition from intensity images. The representation is based on local features and uses as primitives the straight-line segment and the ellipsoidal arc.

By aggregating basic features, 2-D index features associated with 3-D objects were defined. Explicit identification and localization attributes and significance factors were associated to the features.

A method to extract the straight-line segment and the ellipsoidal arc primitives was presented. First, a segmentation process that extracts contour lines with subpixel accuracy was implemented. Then, the corner points of the contour were detected and the contour was approximated with straight-line segments between these points. Then, the convex or concave polygonal lines were approximated with a maximum length ellipsoidal arc using a locally applied Hough transform. For each ellipsoidal arc found, some refinements were done by extending the ends of the arc over its adjacent straight-line segments, as long it was possible. The proposed method was proved robust and computationally efficient, inheriting only the advantages of the Hough based methods.

The experimental results proved that the technique is extremely robust to noise and usable in many applications, especially in industrial vision.

The main advantage of the proposed intermediate representation are the improved robustness in the case of manufactured objects with cylindrical or conical parts and the facilitation of the matching process between the model and the scene by reducing of the searching space. The elements that improve the matching process are:

- Data driven selection possibility of the model by using index features.
- Reduction of the number of hypotheses by using compound index features with higher discrimination power.
- Usage of explicit identification attributes to easy comparison of features of the same type.
- Usage of explicit localization attributes which allow easy inference of the positional relationships between the features.

7. References

[1] R. T. Chin, C. R. Dyer, “Model-Based Recognition in Robot Vision”, *Computing Surveys*, vol. 18, no. 1, pp. 67-108, 1986

[2] A. C. Kak, A. J. Vayda, R. L. Cromwell, W. Y. Kim, C. H. Chen, “Knowledge-based robotics”, *Int. J. Prod. Res.*, vol.26, no.5, pp. 707-734, 1988

[3] R. Horaud, T. Skordas, "Finding Geometric and Relational Structures in an Image", *Proc. First European Conference on Computer Vision*, Antibes, France, pp. 374-384, 1990.

[4] P.I. Rosin, “Techniques for Assessing Polygonal Approximations of Curves”, *Proceedings, British Machine Vision Conference,* pp. 153-162, 1996

[5] B. K. Ray, K. S. Ray, “A new split and merge technique for polygonal approximation of chain coded curves”, *Pattern Recognition Letters*, vol. 16, no. 2, pp.161-169, 1995.

[6] P.L. Rosin, G.A.W. West, “Nonparametric Segmentation of Curves into Various Representations”, *IEEE Trans. PAMI,* vol. 17, pp. 1140-1153, 1995.

[9] C. Ichoku, B. Deffontanies, J. Chorowitz, “Segmentation of Digital Plane Curves: A Dynamic Focusing Approach”, *Pattern Recognition Letters,* vol. 17, pp. 741-750, 1996.

[10] A. Fitzgibbon, M. Pilu, R. B. Fisher, “Direct Least Square Fitting of Ellipses”, *IEEE Trans.PAMI,* vol.21, pp.476-480, 1999.

[11] A. Ratarangshi, R. T. Chin, “Scale-Based Detection of Corners of Planar Curves”, *IEEE Trans. PAMI*, vol.14, pp. 430-449, 1994.

[12] H. T. Sheu, W. C. Hu, “A Rotationally Invariant Two Phase Scheme for Corner Detection”, *Pattern Recognition*, vol. 29, pp. 819-828, 1996.

[13] I. Weiss, A. Rosenfeld, “A simple method of ellipse detection”, *Center for Automatic Research, University of Maryland*, CS-TR-3717, 1996.

[14] T. Lindenberg, M. X. Li, “Segmentation and classification of edges using minimum description length approximation and complementary junction cues”, *Proc. 9th Scandinavian Conference on Image Processing*, Uppsala, June, pp. 767-776, 1995.

[15] H. T. Sheu, W. C. Hu, “Multiprimitive segmentation Of Planar Curves – A Two Level Breakpoint Classification and Tuning Approach”, *IEEE Trans.PAMI*, vol.21, pp.791-797, 1999.

[16] V.F. Leavers, “Shape Detection in Computer Vision Using Hough Transform”, New York, Spriinger-Verlag, 1992.

[17] H.K. Yuen, J. Illingworth, “Shape Using Volumetric Primitives”, *Image and Vison Computing*, vol. 7, pp. 30-37, 1989.

[18] S. Nedevschi, M. Marton, “Recognition Oriented Modeling Environment for Robust 3D Object Identification and Positioning from a Single Intensity Image”, *9th DAAAM International Symposium*, Technical University Cluj, 1998.

[19] S. Nedevschi, Image Processing and Pattern Recognition, *Editura Microinformatica*, Cluj-Napoca, 1997.