A MODEL-BASED TECHNIQUE WITH A NEW INDEXING MECHANISM FOR INDUSTRIAL OBJECT RECOGNITION

Peter L. Stanchev and Valery V. Vutov

Institute of Mathematics
Bulgarian Academy of Sciences
G. Bonchev Str., 8 - 1113 Sofia (Bulgaria)
telex 22628 BANMAT BG, fax 0359-2-752078

ABSTRACT

In this paper, we address the problem of industrial scene object recognition for the purposes of sensor-based robot assembly. The process of industrial scene object recognition is usually divided into a training and recognition phases.

We propose a technique in which the representation of the models (training phase) is performed by using a finite set of primitives and relations between them (object elements), characterized by a proper set of parameters. We describe a new index building mechanism, using both the recognised object primitives and the relations between the primitives. We obtain the characteristic set of primitives and relations for each model of a given industrial object set by eliminating the common (similar) object primitives and relations. The characteristic set will be refered to as Global Index.

The real scene object analysis (recognition phase) includes the recognition of scene primitives and relations between them, as well as their location in the Global Index. The proposed new index organisation gives the possibility to direct access to the particular object elements, significantly speeding up the hypot esis generation for the recognised scene object in comparisor ith other index tree supporting methods [1]. The generated hy thesis is verified, using the whole object representation, obta ed during the training phase.

I. INTRODUCTION

With the increasing emphasis on automated manufacturing .dustry, there has been a growing interest in interpreting im- containing 2D objects. A number of approaches have been posed for such types of scenes [2, 3, 4]. Most of them include aining and recognition phases; The training phase derives object representation and stores it in a model database, in form of recognised object primitives; The recognition phase the preliminary stored information to recognise objects parating in object scenes.

The most commonly used approach in the field is: first, ing the object boundary as a string of primitive structural s, and then utilising some form of string matching [5, 6].

Most of the existing methods are model driven. To analyze rtain scene, the scene objects are compared either with the ls in the model database or with the model sets of features nging to the model database. Some methods give the poslity to speed up the recognition phase by applying feature ex [7].

The technique developed in this study is also a model-riven one and is based on generation of hypothesis of the presence of one or more objects in a scene, retrieval of object representations from a database, and hypothesis verification. For that purpose we build special structures, referred to as Global Index and Table Index. The Global Index contains the specific features of each model (primitives and relations between them), recognised during the training phase. The Table Index includes the recognised relations and the model identifications, where the relations are located.

Our paper is organized as follows: in Section 2 we discuss the general description of the proposed technique. In Section 3 we give a procedural description of these technique. The advantages of the technique are discussed in Section 4. In Section 5 an example is given.

II. GENERAL DESCRIPTION OF THE TECHNIQUE

We view the process of industrial object scene recognition structured in the standard training and recognition phases. In this section, we will present schematically these phases.

II.I. Training phase

The training phase of our technique incorporates the following procedures:

- **Image Table construction procedure.** The input image is converted to a graph [3], which is a two dimensional line representation of the scene. The process of conversion includes: (a) edge detection, (b) line segment encoding, (c) line approximation and (d) relation recognition. The result is an Image Table, composed of the set of recognised primitives (with their positions into the image) and the set of relations between them;
- **Object Table construction procedure.** Using the Image Table and excluding the information for the specific positions of the primitives, we obtain the Object Table, which is stored in a model database;
- **Table Index construction procedure.** The Table Index consists of recognised relations and the Object Table identifications where these relations are located;
- **Global Index construction procedure.** The Global Index includes the object features (primitives and relations between them) that uniquely identify the objects.

The training phase is performed only once for all models in a chosen application domain.

II.II. Recognition phase

In the presented technique, the recognition of scene objects includes the following procedures:

- **Image Table construction procedure.** The procedure, described in the training phase, is performed on the scene of objects;
- **Global Index search procedure.** We search the recognised primitives and relations in the Global Index. If we find them, we use the specific Object Table and the Image Table to eliminate the object elements from the Image Table;
- **Table Index search procedure.** We search the recognised relations in the Table Index, constructed during the training phase. We try to find the corresponding Object Table among the tables that contain the specific relation value. If we find it, we eliminate all the object elements from the Image Table.

The output of the recognition phase is a list of objects, that have been recognised in the scene.

III. PROCEDURE DESCRIPTION OF THE TECHNIQUE

First we will describe the Image Table construction procedure, which is used in both training and recognition phases.

- **Image Table construction procedure.**

The input image is first converted to a graph, which is a two dimensional line representation of the scene. The graph is constructed by segments and nodes, representing the edges and vertices of the image. The procedure for the image-to-graph conversion is borrowed from [3]. It includes:

Edge detection. To find the edges of the image, the Sobel-operator [9] (a line edge detector) is applied. After that a peak filter is used, to detect the ridge pixels. The output of the peak filter is processed by a sceletonization algorithm [9].

Line segment encoding. The image is scanned, starting from the upper left corner, row by row from left to right. If a pixel is encountered, the line follower is started. The line follower steps along the line until a special pixel (end or node pixel) is discovered, and it is marked as the starting point. The line in the opposite direction is followed again, until a special pixel is found. The line is coded by the Chain-code[8].

Line approximation. In our technique we use high level primitives, such as straight line segment, arc and circle:

Segment $(S(l))$, *a line without bending, characterized by its length* (l);

Arc $(A(l_1, l_2))$, *an open line with a bending degree. The bending must not be null, and must be with a constant sign. It is characterized by its length* (l_1) *and the length of the cord* (l_2);

Circle $(C(r))$, *The circle is characterized by its radius* (r).

We will denote by

$$P_i(f_1(x), f_2(x), \ldots, f_{l_i}(x)), (i = 1, 2, \ldots, n)$$

the set of primitives (e.g. segment, arc, circle), where $f_q, (q = 1, 2, \ldots, l_i)$ *are the primitive attributes (e.g. length, angle, radius).*

We use the following set of relations – joint, intersection and facing:

Joint $(J(\alpha))$, *a relation between primitives with a common extremity. It is characterized by an angle* (α). *Each point of a closed arc is an extremity;*

Intersect $(I(\alpha))$, *a relation between primitives having one or more common points that are not extremities for both primitives. This relation is characterized by an angle* (α);

Facing $(F(d, \alpha, \beta))$, *a relation between primitives with no common points. It is characterized by the distance* (d) *between the primitives and two angles* α *and* β. *This distance is measured from the cord or segments medium point and from the circle centre.* α *and* β *are the corresponding angles between the primitives and the joining line on which we mesure the distance.*

We will denote by $R_k^{i,j}(g_1(x), g_2(x), \ldots, g_{s_k}(x)), (k = 1, 2, \ldots, m; i, j = 1, 2, \ldots, n)$ *the set of relations (e.g. joint, intersection, facing);* $g_q, (q = 1, 2, \ldots, s_k)$ *are the relation attributes (e.g. distance, angle, cord length).*

A fast sequential method [10] is used to approximate lines. The method uses a scan along technique where the approximation depends on the area deviation for each line segment. The algorithm outputs a new line segment when the area deviation per length unit of the current segment exceeds a prespecified value.

Next, we establish a relation between the segments and nodes. Nodes, within a certain distance from each other, are merged. The segments in the segment lists of the nodes are combined into one new list, removing duplicates. The node, that is not used, is deleted. Then corners between lines, that make angles very closed to 180°, are removed.

Now successive straight lines with small angles are checked to find arcs. If the maximum distance of the corners of these lines to the straight line, connecting the first and the last corner of the successive lines exeeds a given threshold, the lines form an arc. Otherwise the lines are substituted by a straight line. The small closed graphs are assumed to be holes in the objects and are approximated by circles.

Relation recognition. To recognise the relations between primitives, the following rules are applied:

- using the features related to the primitive position into the image, the joint and intersect relations are assigned. One or more disjoint sets of correlated entities are so obtained;
- facing relations are assigned between primitives (not necessarily of the same type) belonging to disjoint sets and having the least distance. If there is more than one node pair, having the same distance, the facing relation is assigned to the pair whose primitives have a greater global length. If this criterion is not sufficient, that is the two primitives have the same neighbor global length, the following hierarchy has to be used: closed arcs, arcs and segments.

III.1. Training phase

This phase involves the following procedures (Figure 1):

- **Image Table construction procedure.** The procedure is applied on the binary representation of the model.
- **Object Table construction procedure.** We construct the Object Table using the information, contained in the Image Table, ignoring the specific position of the object primitives into the image. The result is a relational symmetrical characteristic matrix.

The matrix rows and columns are labeled with the values of the recognised primitives. In the matrix element, placed at row

(1) $P_i(f_1(x_r), f_2(x_r), \ldots, f_{l_i}(x_r))$

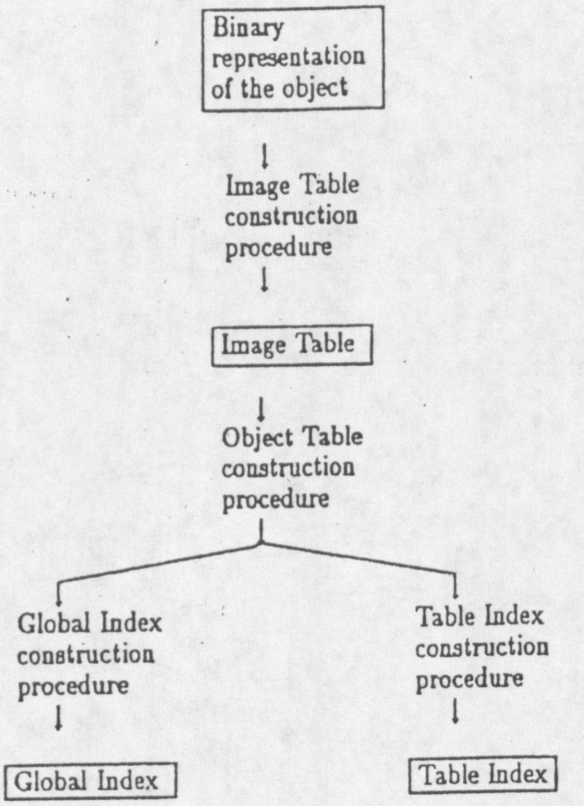

Figure 1. *Training phase scheme*

and at column

(2) $$P_j(f_1(x_c), f_2(x_c), \dots, f_{l_j}(x_c))$$

we put the relation value

(3) $$R_k^{i,j}(g_1(x_r,x_c), g_2(x_r,x_c), \dots, g_{s_k}(x_r,x_c))$$

if exists. The triplet ((1),(2),(3)) is referred to as object element.

- **Table Index construction procedure.** We modify the Table Index by adding new elements to it. Its elements are composed of two parts. The first one is only the part (3) from the triplet ((1), (2), (3)), recognised in the image between the primitives p_i and p_j. The second part is a list of Object Tables where this relation can be found.
- **Global Index construction procedure.** We add the new model matrix to the Global Index. The Global Index for the analysed model set can be represented by a 3-D Cartesian coordinate system Oxyz, where the axes Ox and Oy represent the ordered primitive values from the characteristic matrixes, and the axis Oz represents the corresponding relations between the primitives. We have a direct access to all the elements in this 3-D matrix. Every element saves the index of its native characteristic matrix. If there is a value "A" in the Global Index that is similar to the new added object element value "B", the object element "A" is deleted from the Global Index and the value "B" is rejected as a characterising model value.

$$R_k^{i,j}(g_1(x_r,x_c), g_2(x_r,x_c), \dots, g_{s_k}(x_r,x_c)) \sim$$

$$R_k^{i,j}(g_1(x_r',x_c'), g_2(x_r',x_c'), \dots, g_{s_k}(x_r',x_c'))$$

iff the difference between the primitive values

$$P_i(f_1(x_r), f_2(x_r), \dots, f_{l_i}(x_r))$$

and

$$P_i(f_1(x_r'), f_2(x_r'), \dots, f_{l_i}(x_r'))$$

and the difference between the relation values

$$R_k^{i,j}(g_1(x_r,x_c), g_2(x_r,x_c), \dots, g_{s_k}(x_r,x_c))$$

and

$$R_k^{i,j}(g_1(x_r',x_c'), g_2(x_r',x_c'), \dots, g_{s_k}(x_r',x_c'))$$

are smaller than some chosen value ϵ, i.e.

$$|P_i(f_1(x_r), f_2(x_r), \dots, f_{l_i}(x_r)) -$$

$$P_i(f_1(x_r'), f_2(x_r'), \dots, f_{l_i}(x_r'))| < \epsilon,$$

$$|P_j(f_1(x_r), f_2(x_r), \dots, f_{l_j}(x_r)) -$$

$$P_j(f_1(x_r'), f_2(x_r'), \dots, f_{l_j}(x_r'))| < \epsilon,$$

and

$$|R_k^{i,j}(g_1(x_r,x_c), g_2(x_r,x_c), \dots, g_{s_k}(x_r,x_c)) -$$

$$R_k^{i,j}(g_1(x_r',x_c'), g_2(x_r',x_c'), \dots, g_{s_k}(x_r',x_c'))| < \epsilon.$$

III.II. *Recognition phase*

The recognition phase involves the following procedures (Figure 2):

- **Image Table construction procedure.** The procedure is applied on the binary representation of the scene.
- **Global Index search procedure.** During this procedure we try to find the elements of the Image Table, starting from the upper left corner and going from left to right, in the Global Index. If we find some element $R_k^{i,j}(f_1(x_r,x_c), f_2(x_r,x_c), \dots, f_{s_k}(x_r,x_c))$, we do the Image Table object elimination procedure for it. If we reach the end of the Image Table, we apply Table Index search procedure.
- **Table Index search procedure.** Using the remaining elements in the Image Table, we look for them in the Table Index. If we find a value, we verify our hypothesis using the list of tables from the index. For the elements recognised the Image Table object elimination procedure is performed.

Here, we will describe the Image Table object elimination procedure, used during the recognition phase.

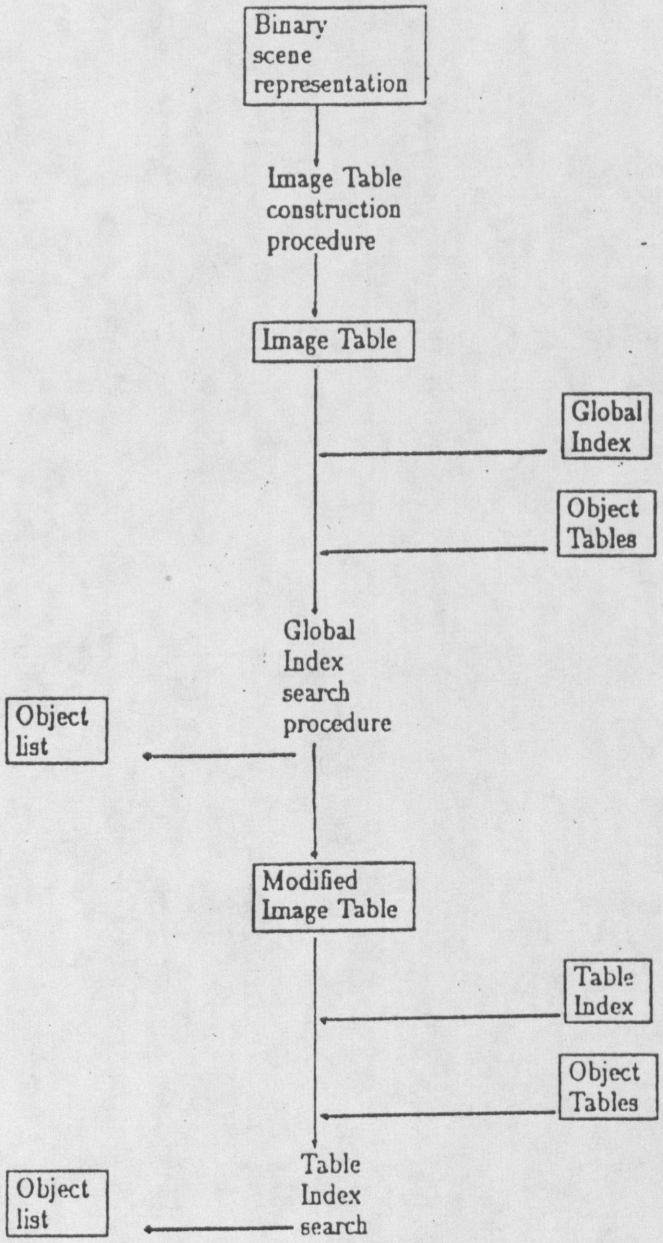

Figure 2. *Recognition phase scheme*

- **Image Table object elimination procedure.** The procedure includes:
  - recognised object reconstruction. Using the corresponding Object Table and the coordinates of the recognised primitives, we reconstruct the object in the scene.
  - recognised object elimination. We delete the elements, belonging to the reconstructed object from the Image Table. These elements are relations between primitives (they may differ in the length parameter from the Object Table primitives).

The main advantages of the proposed technique are:

- The training phase, which is more time comsuming, is only once performed for each model included in the model database.
- The mechanism proposed assures the recognition of the model iff only one characteristic feature of the model is found in the scene. It is an efficient mechanism to find where and in which models, the extracted primitive values and the relation between them exist.
- The proposed technique efficiently supports the insertion and deletion of models. This is handled by a simple Global Index and Table Index modifications.
- The proposed scheme allows an easy way for the insertion and deletion of models. To delete a model, one must search the Global Index and the Table Index for object elements belonging to this model, and eliminate them. We must delete the corresponding model matrix too. To insert a new model one must perform the training phase for it.

V. AN EXAMPLE

Suppose the models in Figure 3 have been entered by a scanner device. The resulting matrix from the analysis of the model presented in Figure 3a is shown in Figure 4. The upper index presents the model identification letter. The Global Index is given in Figure 5. As a result of the analysis of the scene, shown in Figure 6, the table shown in Figure 7 is obtained.

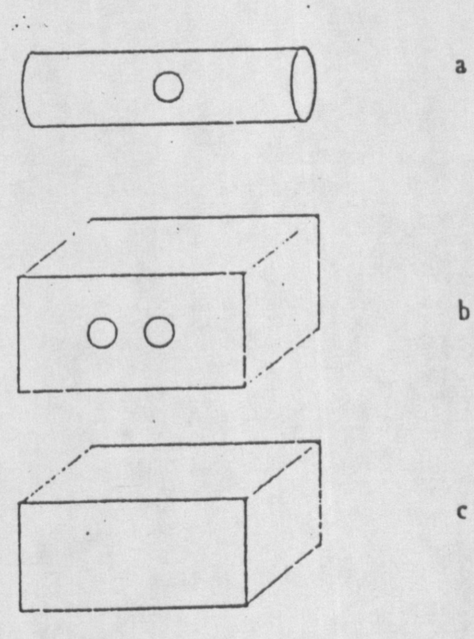

Figure 3. *Model images*

	S^a (4.5)	S^a (4.5)	A^a(1.3, 1.5)	A^a(1.3, 1.5)	A^a(1.3, 1.5)	C^a (0.25)
S^a(4.5)			J(90)	J(90)	J(90)	F(0.6, 90,90)
S^a(4.5)			J(90)	J(90)	J(90)	
A^a(1.3, 1.5)	J(90)	J(90)		J(0)		
A^a(1.3, 1.5)	J(90)	J(90)	J(0)			
A^a(1.3, 1.5)	J(90)	J(90)				
C^a(0.25)	F(0.6, 90,90)					

Figure 4. *Model matrix for the object in Figure 3a.*

Facing	C^a(0.25)	C^b(0.25)	C^b(0.25)
S^a(4.5)	F(0.6,90,90)		
C^b(0.25)		F(1,0,0)	
S^b(3.9)			F(1,60,0)
S^b(3.9)			F(1,60,0)

Joint	S^a (4.5)	S^a (4.5)	A^a (1.3, 1.5)	A^a (1.3, 1.5)	A^a (1.3, 1.5)
S^a(4.5)			J(90)	J(90)	J(90)
S^a(4.5)			J(90)	J(90)	J(90)
A^a(1.3, 1.5)	J(90)	J(90)		J(0)	
A^a(1.3, 1.5)	J(90)	J(90)	J(0)		
A^a(1.3, 1.5)	J(90)	J(90)			

Figure 5. *Global Index matrizes*

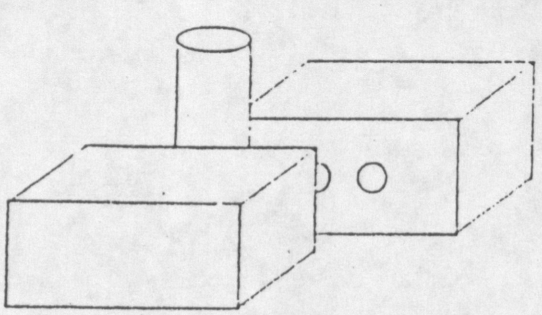

Figure 6. *Scene image*

Model	Rotation
a	90
b	0
c	0

Figure 7. *The result of the example scene analysis*

VI. CONCLUSIONS AND FUTURE WORK

The described technique applys a data-driven indexing mechanism for model retrieval (hypothesis generation) as well as a standard model-driven approach to hypothesis verification. The proposed indexing mechanism is more efficient than the tree indexing mechanism, because it gives the possibility to directly access the object elements. The technique proposed is embeded into a system which is under development on IBM PC computer. It is written in C and Assembler. It makes use of Canon IX12 scanner. An experience with image database management, obtained during the development of the IM_DBMS system [11], has been followed.

REFERENCES

[1] R. Mehrotra and W. Grosky, "Efficient Object Recognition Using a Data-Driven Model-Based Technique", Proceedings of Sixth Annual Int. Phoenix Conference on Computers and Communications, Scottsdale, Arizona, Feb. 1987, pp. 508-512.

[2] Y. Lamdan and H. Wolfson, "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme", Proceedings of Second Int. Conference on Computer Vision, Tampa, Florida, Dec. 1988, pp. 238-249.

[3] J. Buurma, G. Weller, F. Groen, G. Meyer and L. Hertzberger, "Graph-Construction and Matching for 3-D Object Recognition and Pose Estimation", in "Artificial Inteligence and Information-Control Systems of Robots-89", I. Plander (edt.), North-Holland, 1989, pp.383-389.

[4] Y. Liu and S. Ma, "3-D Object Recognition by Graph Matching", Proceedings of Int. Conference on Intelligent Autonomous Systems, Amsterdam, Dec. 1989, pp. 918-921.

[5] L. Bahl and F. Jelinek, "Decoding for Channels with Insertion Deletions, and Substitutions with Applications to Speech Recognition", IEEE Transaction on Information Theory, vol. IT-21, 1975, pp. 404-441.

[6] R. Lowrance and R. Wagner, "An Extension of String-to-String Correction Problem", Journal of ACM, vol. 22, 1975, pp. 177-183.

[7] T. Knoll and R. Jain, "Recognising Partially Visiable Object Using Feature Indexed Hypotheses", IEEE Journal of Robotics and Automation, vol. RA-2, 1986, pp. 3-13.

[8] H. Freeman, "Computer Processing of Line Drawing Images", ACM Comp. Surv. 6, 1, March 1974, pp. 57-97.

[9] A. Smeulders, "An Introduction to Image Processing and Computer Vision", University of Amsterdam, 1990.

[10] W. Karin and P. Danielsson, "A Fast Sequential Method for Polygonal Approximation of Digitized Curves", Computer Vision, Graphics, and Image Processing 28, 1984, pp. 220-227.

[11] P. Stanchev and F. Rabitti, "An Image Database Management System. Functional Description", in "Artificial Inteligence and Information-Control Systems of Robots-89", I. Plander (edt.), North-Holland, 1989, pp. 389-396.

(64.) Stanchev P., Vutov V., "A Model - Based Technique with a New Indexing Mechanism for Industrial Object Recognition" ***9-th 1990 IEEE/CHMT International Electronics Manufacturing Technology Symposium,*** Washington DC, USA, October 1-3, 1990, 56-60.