# Recognition of Partially Occluded Objects Using B-tree Index Structure: An Efficient and Robust Approach

R. Dinesh and D.S. Guru
Dept. of Studies in Computer Science, University of Mysore, Manasagangotri, Mysore-570 006
Email: dinesh_r21@yahoo.com and guruds@lycos.com

## Abstract

*This paper presents a novel method for recognizing partially occluded objects. The proposed method uses corner points and their spatial relationship perceived through the application of Triangular Spatial Relationship (TSR) [5] by considering three successive corner points at a time. The perceived TSR among corner points are used to create a model object-base using B-tree, an efficient multilevel indexing scheme. The matched sequence is preserved in a two-dimensional matrix called status matrix. Experimental results, on real images of varying complexity of a reasonably large database of objects have established the robustness of the method.*

**Keywords:** Corner points, Corner labeling, Triangular spatial relationship, B-tree, Matching, Status matrix, Partially occluded object recognition

## 1. Introduction

Since 30 years, many attempts have been made in the recognition of objects in isolation with complete and well-defined boundaries. But in most of the situations, it is seldom possible to find complete boundary of an object due to presence of noise, which degrades parts of an object, inability in covering entire object in the field of view of a capturing media, etc. Indeed, in practice, objects are generally kept such that they are often occluded or articulated. Therefore, detection of objects in a cluttered environment is, day by day, receiving much attention of the research community.

A number of methods have been proposed for recognition of partially occluded objects. Price [12] has proposed a method, where each model segment is compared to every scene segment in terms of the lengths and the included angles between successive line segments of the approximated polygon of the objects and the matching information is stored in the disparity matrix. Although, the method proposed by Price [12] is simple, it is not computationally efficient since every entry of the disparity array has to be tested for the starting location of the longest sequence. Furthermore, the technique is sensitive to scale variation because the feature value, such as the length of a line segment, used in this technique is inherently scale dependent. Bhanu and Ming [2] have improved the price's approach by introducing the concept of K-means clustering, to cluster the matched line segments. Though, the improved method is much more efficient than the price's [12] method, it still remains impractical due to its iterative nature. To overcome the iterative nature of the method proposed by Bhanu and Ming [2], methods based on dynamic programming [4], [1], [18], [19] and distance transformation [9], [17] were proposed. Other interesting methods for recognizing partially occluded objects include fuzzy relaxation procedure [10], modified generalized hough transformation [15], neuro-fuzzy reasoning [14], markov model [3], probabilistic Attributed Relation Graph (ARG) matching [11].

All the aforementioned methods for recognizing partially occluded objects are model-based systems. In these systems, all known objects are precompiled and stored in some random order in the database. During recognition, model objects are retrieved in sequence from the database and are matched against the scene object one at a time until a correct match is found. Therefore, time complexity of these model-based systems is extremely high and is increased as the number of model objects in the database is increased.

Therefore, indexing methods seem to be potential approaches to solve the problem of partially occluded object recognition. In view of this few indexing methods were also proposed for recognizing occluded objects. Lamdan et al., [7], [8] have proposed methods for recognizing partially occluded objects using the concept of geometric hashing. The complexity of the algorithm depends on the number of model points in the scene. It is bounded from above by O(n4) and from below by O(n), where n is the number of scene points. However, the performance of this algorithm is dependent on the selection of quantization level of each bin. The quantization level is much more difficult to determine when the image is contaminated with noise. Moreover, large computer memory is required for this method.

To partially overcome the problems associated with the model-based methods for recognizing partially occluded objects, Tsai and Tsai [16], have proposed an artificial neural network (ANN) approach to determine the matching order of models in the database. This method is not complement to the model-based methods, in fact it is supplement to the model-based methods, The match between a model and a scene image is based on the rank of similarity of the model rather than its serial storage order in the database. The method is computationally expensive and not suitable for the real time applications. Rajpal et al., [13] have proposed an indexing method for recognizing partially occluded objects using neural network. Problems with the neural network based methods are that they require long time for training the neural networks and also insertion and deletion of objects into the knowledge base require retraining/reconstruction of the neural network.

From the above discussion it is clear that the aforementioned methods suffer either due to heavy computation load or due to poor recognition results, and hence they are of little use in the real pragmatic applications. In view of this in this paper we propose a novel method for recognizing partially occluded objects. The proposed method uses corner points and their spatial relationship perceived through the application of Triangular Spatial Relationship (TSR) (Guru and Nagabhushan, 2001) by considering three successive corner points at a time. The perceived TSR among corner points are used to create a model object base using B-tree an efficient multilevel indexing scheme. Since the TSR is preserved by the use of quadruples and B-tree does not support multidimensional data (4-dimension in our case) a distinct and unique key is computed for each of the distinct quadruple and then the computed keys are stored in B-tree as representatives of the corresponding quadruples. Each key value stored in B-tree is attached with a list of objects, which have the corresponding quadruple as one of their associated quadruples. The matched sequence is preserved in a two-dimensional matrix called status matrix. The object for which there is a longest substring of 1s in the status matrix is declared to be the identified object. Experiments conducted on a set of industrial tools reveal the superiority of the proposed method.

## 2. The concept of Triangular Spatial Relationship: An overview

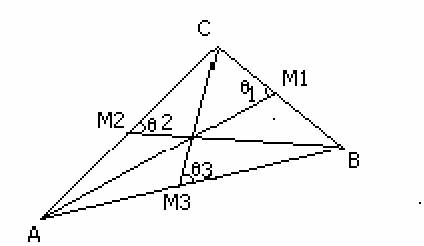


Fig. 1 Triangular spatial relationship

Let A, B, and C be any three non-collinear components of a symbolic image. Let La, Lb and Lc be the labels of A, B and C respectively. Connecting the centroids of these components mutually forms a triangle as shown in Fig.1. Let M1, M2, and M3 be the midpoints of the sides of the triangle as shown in Fig.1. Let $\theta 1$, $\theta 2$, and $\theta 3$ be the smaller angles subtended at M1, M2, and M3 respectively and are shown in Fig.1. The triangular spatial relationship among the components A, B and C is represented by a set of quadruples $\{(L_a, L_b, L_c, \theta_3), (L_a, L_c, L_b, \theta_2), (L_b, L_a, L_c, \theta_3), (L_b, L_c, L_a, \theta_1), (L_c, L_a, L_b, \theta_2), (L_c, L_b, L_a, \theta_1)\}$. Since representing all six possible quadruples for every three non-collinear components is unwieldy, it is recommended to choose only one of those based n a set of defined criteria [5]. The TSR is claimed to be invariant to image transformations for more details on TSR, the readers are directed to refer [5].

## 3. Proposed Model

In this section we propose a novel method of representing objects by the use of local descriptors and also the corresponding recognition algorithm based on the proposed representation scheme for the purpose of recognizing partially occluded objects. Thus the proposed model has the following two major stages.

### 3.1 Representation of objects

The representation scheme identifies the local descriptors called corner points and then perceives the spatial relationship among the corner points considering three consecutive corner points at a time by the use of TSR based on their labels and then preserve the perceived TSR in B-Tree, a multilevel indexing structure.

#### 3.1.1 Corner extraction and labelling

The images of the objects to be represented are preprocessed to extract the boundary curves. The corner detection algorithm [6] is employed to locate all corner points present in the extracted boundary curves. The method [6] is specifically chosen because it is non-parametric in addition to being adaptive and robust to image transformations such as scaling, translation and rotation. Based on the curvature information and region of support of each corner point on a boundary, the corner point is either classified into a convex or concave corner point. By the use of curvature value at a corner point due to its left arm and right arm [6], the corner points are further categorised into say *l* more sub classes in each convex and concave classes. Hence there are $2*l = m$ number of distinct corner labels. Thus, the boundary of an object can now be represented as a sequence of class labels of corner points present on the boundary curve along with their coordinates, when the boundary is traversed in counter clockwise direction. Let B be the boundary curve of an object, with b number of corner points in it. Let $C_1, C_2, \ldots, C_b$ be the sequence of the labels of b corner points of B, when B is traversed in counter clockwise direction, that is, $B = \{ C_i \mid C_i = (x_i, y_i) \in Z^2, i = 1,2,\ldots,b\}$ where $C_{i+1}$ is the label of the next corner point of B visited after the corner point whose label is $C_i$ (mod b) visited when B is traversed in counter clockwise direction.

It should be noticed here that the labels $C_1, C_2, \ldots, C_b$ are not necessarily distinct. However, the associated (x, y) are different.

Now, the problem of recognizing an object in occlusion is reduced to the problem of finding out the common longest subsequence of corner labels of the model object and the test scene object, with the same spatial scattering. This method is not advised since the process of substring matching takes non-deterministic polynomial time complexity. In view of this, in this work we recommend to perceive the triangular spatial relationship existing among the corner points assuming that they are the components of the image and then to preserve the perceived TSR in a B-tree as explained in the next section.

### 3.1.2 Representation of objects in B-tree through the perception of TSR.

Let $B_1,B_2,\ldots,B_n$ be the boundary curves respectively of the objects $O_1,O_2,\ldots,O_n$ to be learnt and represented in a object database. Let $C_1, C_2,\ldots,C_m$ be m distinct corner labels. Here, each $C_i$ is nonzero positive numerical value less than or equal to m. Each boundary $B_i$ is said to have $b_i$ number of corner points. In order to make the representation scheme invariant to image transformations, we recommend to perceive the TSR existing among the corner points considering three consecutive corner points at a time, and then to preserve the TSR by the use of quadruples as explained in the previous section. Hence, there are $b_i$ number of quadruples computed for the boundary curve $B_i$. Thus, the problem of object representation is reduced to the problem of storing the computed quadruples such that the recognition task becomes effective and efficient. Hence, it is advised to store quadruples in the object database through creation of B-tree, an efficient multilevel indexing structure, which outperforms any method based on neural networks and geometric hashing technique. Another advantage of B-tree is that it supports easy insertion and deletion. However, B-tree does not support storage of multivalued data such as quadruples. Thus, a unique and distinct number called key is generated for each of the distinct quadruples to be stored in B-tree and then the generated keys are stored in the B-tree as representatives of the corresponding quadruples. If $(C_p,C_q,C_r,\theta)$ is the quadruple to be stored in B-tree then the key K corresponding to the quadruple is computed as,

$$K = D_\theta (C_p-1)m^2 + D_\theta(C_q-1)m + D_\theta(C_r-1)+(C_\theta -1) \ldots\textbf{(1)}$$

where $D_\theta$ is the number of slices/classes, the continuous interval type domain [0..90] associated with $\theta$ is split into and $C_\theta$ is the class number to which a specific value of $\theta$ belongs. The discritization of the continuous domain of $\theta$ is suggested to take care of the possible errors that can occur during the computation of $\theta$ value due to the limitation of the computing system in handling floating point numbers. It can be noticed that the associated keys for any two different quadruples are distinct and unique.

Let N be the total number of distinct quadruples generated due to all n objects and let $\{K_1,K_2,\ldots,K_N\}$ be the set of corresponding keys. All these N TSR keys are stored in a B-tree. Each key value is then attached with a list of object indices. The object indices that are attached to a key value (K) are the indices of objects which have the key K as one of the keys in the corresponding key set. Therefore, following is the algorithm proposed for representing objects in the object database.

**Algorithm**: Proposed representation scheme.
Input: $O_1,O_2,\ldots,O_N$ images of objects.
Output: Object database (B-tree)
Method:
Step 1: For each object $O_i$ do
- (i) Extract the boundary curve.
- (ii) Employ the corner detection method (Guru and Dinesh, 2004)
- (iii) Label each detected corner point.
- (iv) Apply TSR and obtain a set of quadruples preserving TSR among the corner points by considering three consecutive corner points at a time.
- (v) For each obtained quadruple, compute a unique key using equation (1) and update the list of object indices associated with that key by inserting i as a new index.

For end.
Step 2: Obtain the set of all keys computed in step 1.
Step 3: Create a B-tree of rank r, containing all the keys obtained in Step 2.
Step 4: Attach each key in the B-tree, with the respective list of object indices.
**Algorithm ends.**

## 3.2 Recognition of objects

There are two kinds of object recognition: recognition of an object in isolation and recognition of an object in occlusion, with the former a special case of the latter. In case of occluded object recognition, the object may be occluded either by a foreign object or by another object in the training set. For a given scene image, the boundary detection procedure is performed to obtain the boundary curve and then aforementioned corner detection algorithm is employed to locate all corner points present in the extracted boundary curve. The detected corner points are then labelled as explained in the previous subsection. Subsequently, the TSR existing among labelled corner points by considering three consecutive of them at a time are perceived and preserved by the use of quadruples. For each of the quadruple a unique and distinct key is computed using Equation (1). Let $K_1,K_2,\ldots,K_T$ be the sequence of the TSR keys generated in counter clockwise direction, where T is the total number of quadruples (number of corner points) generated for the scene image. Since the quadruples are not necessarily distinct, these T keys are not necessarily distinct. For each key K, the B-tree is accessed through to retrieve the list of object indices associated with K. The key K may or may not be present in the B-tree. In case, K is not present in the B-tree, then the list associated with K is assumed to be empty. This situation shall occur even if the scene objects are the trained objects, because the occlusion of objects create new corner points with different TSR. Indeed, this knowledge would be of some use for finding out the location where occlusion has been taken place and based on that, the orientation of the object in occlusion can easily be estimated. Once the list of object indices are extracted for each key $K_j$, a matrix of size n x T called status matrix, is constructed where $(i, j)^{th}$ entry will be set to high (value 1) if the object $O_i$ is retrieved because of $j^{th}$ key $K_j$, otherwise it is set to zero. Assuming each row as a binary string, we then look for a row with a longest substring containing only 1s. The object corresponding to that row is declared to be detected in the scene. Therefore, the following is the designed algorithm to recognize an object in occlusion.

**Algorithm**: Recognition.
Input: A scene image, I
Output: the index of object(s) present in I.
Method:

Step 1: Employ a boundary extraction procedure on I to extract the boundary curve B

Step 2: Detect corners on B and label them. Let T be the number of corner points in B.

Step 3: Compute TSR key for each corner point as explained already.

Step 4: Access through the B-tree in search of each key and extract the list of object indices associated with that key and then construct the status matrix of size n x T, as explained above.

Step 5: Declare the index of the row, which has the longest substring with only 1s as the index of the object seen in the given test image (I).

**Algorithm ends.**

Since the proposed method for recognizing partially occluded objects, is based on B-tree, the algorithm has logarithmic time complexity.

## 4. Experimentation

An experiment on a set of twelve hand tools has been conducted to corroborate the success of the proposed methodology. In this section we present the results obtained for the subset of the set of hand tools considered as an instance. Fig. 2 shows the images of four hand tools out of twelve. The extracted boundary curves along with the labels of the corners detected are given in Fig. 3.The proposed method of representation is employed on these boundary curves and the obtained quadruples along with their corresponding keys are given in Table-1. The keys are inserted into the B-tree and then the associated indices list of objects is attached to each key (See Fig. 4). Throughout the experiment value of m is set to 8.

The test image shown in Fig. 5 is considered for recognition and the status matrix obtained is given in Table-2. It is clear from Table-2 that the identified object is the hammer (longest substring of 1s is in the row corresponding to hammer).

Table-3 is the status matrix obtained for the test scene shown in Fig. 6. It can be noticed that the rows corresponding to the cutter and spanner have got longest substrings containing only 1s.

The proposed recognition method is employed on the test scene shown in Fig. 7 also and the obtained status matrix is given in Table-4. It can be observed from table-4 that the scene has hammer, spanner. The presence of 1s in the row corresponding to wrench signifies the presence of wrench also. However that decision is not always acceptable. Even the human vision system fails to identify the presence of wrench in Fig. 7.

It can also be observed in all the test scenes that the objects are in orientation which is different from the original orientation and hence the method is invariant to image transformations.

## 5. Discussion and conclusion

A method for identifying partially occluded objects in a digital scene has been presented in this paper. The proposed method being based on B-tree representation outperforms all the existing model-based methodologies. As the B-tree allows easy insertion and deletion the proposed method outperforms other existing neural network based indexing schemes also for recognition of objects in occlusion. The proposed method is invariant to image transformations.
Extension of the method to estimate the orientation and scaling factors of the detected objects and strengthening the method to establish subsequence match by the use of predecessor keys are the targets of our future research.

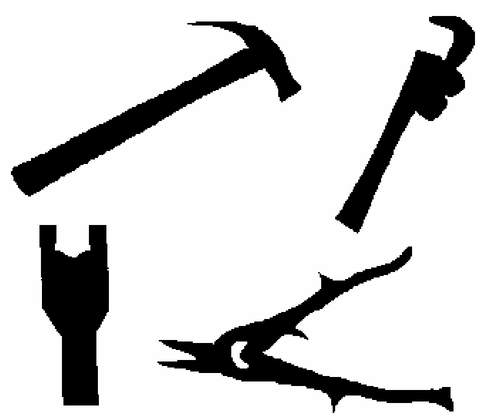

Fig. 2: Images of four hand tools

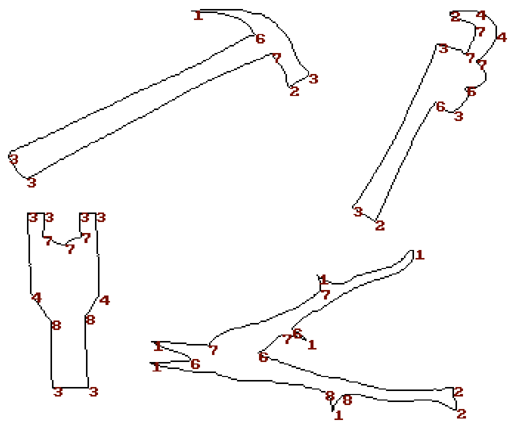

Fig. 3: Extracted boundary along with the labels of the corners detected.

Table 1: Quadruples along with the generated keys for four hand tools

| Hammer | | Wrench | | Spanner | | Cutter | |
|---|---|---|---|---|---|---|---|
| Quadruple s | Keys | Quadruples | Keys | Quadruples | Keys | Quadruples | Keys |
| 6 3 1 11.53 | 6311 | 7 4 4 83.16 | 7446 | 4 3 3 21.45 | 4332 | 6 1 1 58.42 | 6114 |
| 6 3 1 28.67 | 6312 | 7 6 4 8.98 | 7641 | 7 3 3 65.68 | 7335 | 6 1 1 13.94 | 6111 |
| 6 3 3 12.71 | 6331 | 7 6 3 4.21 | 7631 | 7 7 3 50.19 | 7734 | 7 6 1 55.38 | 7614 |
| 7 3 3 12.49 | 7331 | 6 6 3 69.19 | 6635 | 7 7 7 78.45 | 7776 | 7 6 1 33.14 | 7613 |
| 7 3 2 15.99 | 7322 | 6 3 2 89.41 | 6326 | 7 7 3 40.64 | 7733 | 7 6 2 53.34 | 7624 |
| 7 3 2 60.77 | 7325 | 6 3 2 21.23 | 6322 | 7 3 3 72.35 | 7335 | 6 2 2 13.05 | 6221 |
| 3 2 1 69.55 | 3215 | 3 3 2 14.70 | 3321 | 4 3 3 20.75 | 4332 | 8 2 2 24.17 | 8222 |
| | | 7 3 3 16.64 | 7332 | 8 4 3 30.59 | 8433 | 8 2 1 14.90 | 8211 |
| | | 7 7 3 63.13 | 7735 | 8 4 3 30.30 | 8433 | 8 8 1 80.53 | 8816 |
| | | 7 7 2 53.59 | 7724 | 8 3 3 49.49 | 8334 | 8 1 1 10.85 | 8111 |
| | | 7 4 2 73.61 | 7425 | 8 3 3 53.11 | 8334 | 8 6 1 6.71 | 8611 |
| | | 4 4 2 32.54 | 4423 | 8 4 3 27.13 | 8432 | 6 1 1 57.83 | 6114 |
| | | | | 8 4 3 27.45 | 8432 | 7 6 1 53.84 | 7614 |
| | | | | | | 7 7 1 15.14 | 7712 |
| | | | | | | 7 7 1 12.97 | 7711 |
| | | | | | | 7 1 1 18.74 | 7112 |

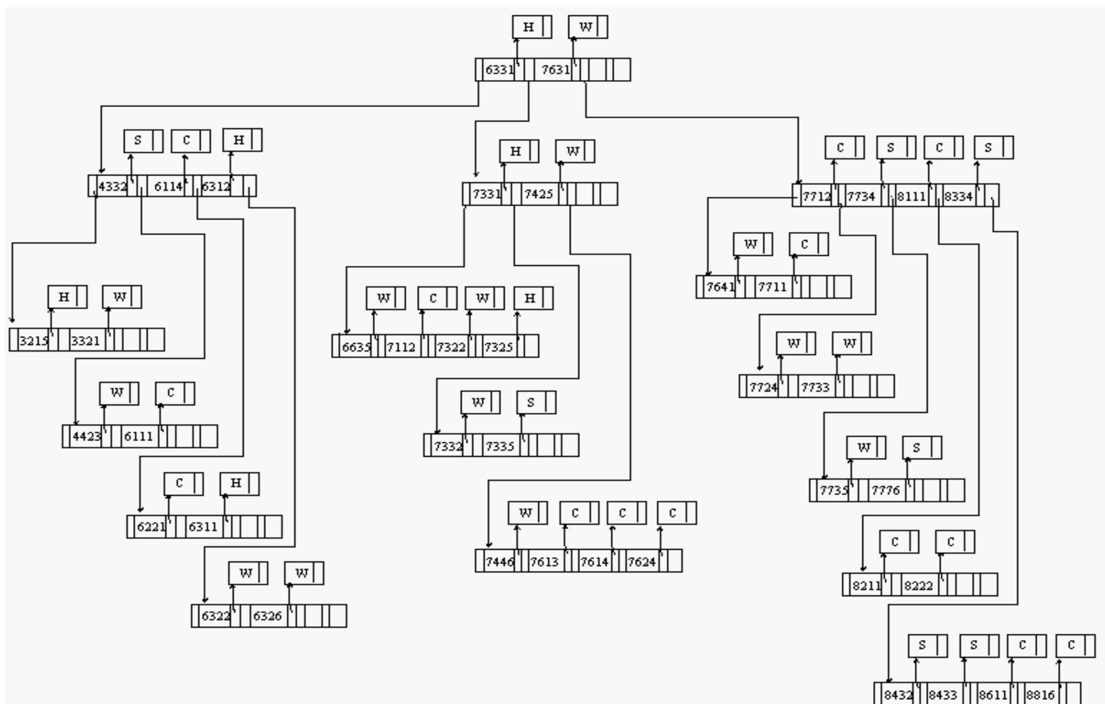


Fig. 4: B-tree representation

Fig. 5: Test image, hammer is occluded by foreign object

Fig. 7: Test image with hammer, spanner and wrench in occlusion

Fig. 6: Test image with cutter and spanner in occlusion

Table-2: Status matrix for test image in Fig. 5

| | |
|---|---|
| Hammer | 1 1 1 0 0 0 0 0 0 0 0 0 0 0 |
| Wrench | 0 0 0 0 0 0 1 0 0 0 0 0 0 0 |
| Spanner | 0 0 0 0 1 0 0 0 0 0 0 0 0 0 |
| Cutter | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |

Table-3: Status matrix for test image in Fig. 6

| | |
|---|---|
| Hammer | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |
| Wrench | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |
| Spanner | 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 |
| Cutter | 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 |

Table-4: Status matrix for test image in Fig. 7

| | |
|---|---|
| Hammer | 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 |
| Wrench | 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 |
| Spanner | 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 |
| Cutter | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |

## Reference:


[1] Ansari N and E.J. Delp, (1990) Partial shape recognition: A landmark-based approach, IEEE Trans. PAMI, Vol. 12, No. 5, pp. 470-483.

[2] Bhanu B. and J. C. Ming, (1987). Recognition of occluded objects: A cluster-structure algorithm. Pattern Recognition, 20(2), pp. 199-211.

[3] Chan C.J and S.Y Chen, (2002). Recognizing partially occluded objects using markov model, Int. Journal of Pattern Recognition and Artificial Intelligence, Vol. 16, No. 2, pp. 161-191.

[4] Gorman J.W, O Robert Mitchell and Frank P. Kuhl, (1988). Partial shape recognition using dynamic programming, IEEE Trans. PAMI, Vol. 10, No. 2, pp. 257-266.

[5] Guru D. S and P. Nagabhushan, (2001). Triangular Spatial Relationship: A New Approach for Spatial Knowledge Representation, Pattern Recognition Letters 22, pp. 999-1006.

[6] Guru D.S and R. Dinesh, (2004). Non-parametric adaptive region of support useful for corner detection: a novel approach, Pattern Recognition, Vol. 37, No. 1, pp. 165-168.

[7] Lamdan Y, J.T. Schwartz and H.J. Wolfson, (1988 a). On recognition of 3-D objects from 2-D images, in Proc. IEEE Int. Conf. Robotics and Automation, Philadelphia, PA, pp. 1407-1413.

[8] Lamdan Y, J.T. Schwartz and H.J. Wolfson, (1988). Object recognition by affine invariant matching, in Proc. IEEE Conf. Computer Vision and Pattern Recognition, Ann Arbor, MI, June 5-9, pp. 335-344.

[9] Liu H.C and M.D Srinath, (1990). Partial shape classification using contour matching in distance transformation, IEEE Trans. PAMI, Vol. 12, No. 11, pp. 1072-1079.

[10] Ogawa H, (1994) A fuzzy relaxation technique for partial shape matching, Pattern Recognition Letters, Vol. 15, pp. 349-355.

[11] Park B.G, M.L Kyoung, U.L Sang and J.H Lee, (2003). Recognition of partially occluded objects using probabilistic ARG (Attributed Relation Graph)-based matching, Computer Vision and Image Understanding, Vol. 90, pp. 217-241.

[12] Price K.E, (1984). Matching closed contours, in Proc. Seventh Int. Conf. Pattern Recognition, Montreal, P.Q., Canada, July 30-Aug. 2, pp 990-992.

[13] Rajpal N., S. Chaudury, S. Banerjee, (1999). Recognition of partially occluded objects using neural network based indexing. Pattern Recognition Letters, 32, pp 1737-1749.

[14] Ray K.S, Jayati Ghoshal, (1998). Neuro-fuzzy reasoning for occluded object recognition, Fuzzy Sets and Systems, Vol. 94, pp. 1-28.

[15] Tsai D. M., (1997). An improved generalized hough transform for the recognition of overlapping objects. Image and Vision computing, 15, pp 877-888.

[16] Tsai D.M and R.Y Tsai, (1996). Use neural networks to determine matching order for recognizing overlapping objects. Pattern Recognition Letters 17, pp 1077-1088.

[17] Tsang P.W.M, P.C. Yuen and F.K. Lam, (1994). Classification of partially occluded objects using 3-poing matching and distance transformation, Pattern Recognition, Vol. 27, No. 1, pp. 27-40.

[18] Yuen P.C, W.M.P Tsang and F.K Lam, (1994). Robust matching process a dominant point approach, Pattern Recognition Letters, Vol. 15, pp. 1223-1233.

[19] Yuen P.C, W.M.P. Tsang (1995). Local feature detector using 3-point matching and dynamic programming, Image and Vision Computing, Vol. 13, No. 3, pp. 179-187.