Clustering Based Recognition of Occluded Objects

Bir Bhanu and John C. Ming

Department of Computer Science, University of Utah
Salt Lake City, Utah 84112

Abstract - Clustering techniques have been used to perform image segmentation, to detect lines and curves in the images and to solve several other problems in pattern recognition and image analysis. In this paper we apply clustering methods to a new problem domain and present a new method based on a cluster-structure paradigm for the recognition of 2-D partially occluded objects. We also discuss the application of the clustering techniques to 3-D object recognition. In both cases, the cluster-structure paradigm entails the application of clustering concepts in a hierarchical manner. The amount of computational effort decreases as the recognition algorithm progresses. The implementation of the techniques discussed for the 2-D case have been completed and have been evaluated with respect to a large number of examples where several objects partially occlude one another. The method is able to tolerate a moderate change in scale and a significant amount of shape distortion arising as a result of segmentation and/or the polygonal approximation of the boundary of the object. A summary of the results is presented.

I. Introduction

Recognition of partially occluded objects is of prime importance for industrial machine vision applications and to solve real problems in military domain and factory automation because occlusion will be present in all but the most constrained environments. The problem of occlusion in a two-dimensional scene introduces errors into many existing vision algorithms which cannot be resolved. Occlusion occurs when two or more objects in a given image touch or overlap one another. In such situations vision techniques using global features to identify and locate an object fail because descriptors of part of a shape may not have any resemblance with the descriptors of the entire shape.

Several methods have been developed [1, 2, 4] to solve the occlusion problem. These approaches can be classified either as boundary based [2, 6] or local feature based [4]. Unfortunately, these techniques are computationally intensive. They cannot handle minor distortion in the shape, change in scale and do not give good matching results over a wide range of industrial objects.

In order to overcome the factors which caused many of these methods to fail, we have used a cluster-structure paradigm which allows the recognition of a given object and provides information about the orientation and position of the objects in the image. The paradigm applies the clustering concept in a hierarchical manner, which reduces the amount of computational effort as the recognition algorithm progresses. Basically the technique consists of three steps: clustering of segments, finding sequences of segments in appropriately chosen clusters and clustering of sequences. The length of each of these border segments as well as the angle between successive segments comprise the only information needed by the algorithm to recognize objects and find their position and orientation. The remainder of this paper describes the 2-D technique that has been implemented and summarizes the results obtained. We also discuss the applications of these methods in three dimensions.

II. Algorithm Description

As shown in Figure 1, the algorithm consists of the following main computational steps:

1) Disparity Matrix
2) Initial Clustering
3) Sequencing
4) Final Clustering
5) Transform Computation

Initially, two sets of data are assumed to be given. Each data set is merely the vertices which define the object boundary. The first set contains the model data, that is the object that we are searching for in the image. The second data set contains the description of the image that has been acquired.

Disparity Matrix: The first step of the algorithm consists of the formation of the disparity matrix. From the set of vertices for the object and the image, the algorithm determines the length of each segment and the angles between successive segments. Every segment in the object model is then compared with every segment in the image. If the segment lengths and successor angles are compatible, the rotational and translational disparity between the pair of segments is computed. These values are stored in the disparity matrix, indexing the values according to the segment numbers. After all segments have been compared, the disparity matrix values are normalized so that the rotation and translation values contribute equally in the clustering step. This matrix is similar in structure to the matrix used by Price [6]. However, in addition to the rotational offsets, we also place translational offsets in the disparity matrix.

Initial Clustering: Clustering, in its most general form, groups a set of objects into subsets where objects in a subset are more similar than the objects in other subsets. Clustering techniques are commonly used in pattern recognition and image processing. (For a recent review see [5].) A significant problem inherent in using the clustering techniques involves the choice of the number of clusters to be used at any given time. Fortunately, there are measures which can be used to find the intrinsic number of clusters [5]. These performance measures determine the scattering of the samples within each individual cluster as well as the distance between each of the cluster centers themselves. This information is held in a matrix form known as a scatter matrix. The scattering of the samples in a particular cluster is defined as within-cluster scatter matrix, S_w. The overall position of all clusters in relation to each other becomes the between-cluster scatter matrix, S_b. By definition, the β value for a certain clustering equals the trace of the within-cluster scatter matrix multiplied by the trace of the between-cluster scatter matrix, i.e., $\beta=Tr(S_w)Tr(S_b)$. As the number of clusters increases, the value of β will reach a maximum and then slope towards 0. The number of clusters at which the value of β is a maximum is the desired value and gives the best results. Several clustering techniques were evaluated and we selected the K-means method for use in our algorithm.

Thus, after all of the normalized values have been placed into the disparity matrix in the previous step, the algorithm clusters these values where the feature vector is merely the normalized rotational and translational offsets for each of the pairs of line segments. The initial number of clusters is set equal to one and in the application of the K-means algorithm the first sample becomes the first cluster center. At each step of the K-means method, all of the samples are clustered, the value of the new cluster centers are recomputed, and this process continues until none of the cluster centers change their positions. Now for the current cluster results, scatter matrices are computed and the value of β is determined. The algorithm then compares the current β value with the last β value. If the value has decreased, then the previous β value and the number of clusters become the final result of this processing step.

After the intrinsic number of clusters have been determined and the results are known for that particular value, the program selects the cluster with the largest number of samples. The data in this cluster will be used by the rest of steps in the algorithm to determine the location and orientation of the model in the image. However, since some of the other clusters may contain approximately the same number of samples as the largest one, the program also uses any cluster which is within 20% of the largest cluster. Each cluster is considered separately and the final transform comes from the cluster which yields the highest confidence level. The program now passes each cluster that has been selected to the following algorithm steps, one at a time.

Sequencing: Since the clustering results provide no information concerning the physical structure of the model, this information must be provided at this time. Using the samples in the current cluster, the program finds all sequences in these samples in one pass over the data. For instance, if the previous sample indicates that segment 1 in the model matches segment 27 in the image (represented by the notation [1,27]), the program then searches for the pair [2,28], since this pair should logically follow the first pair on the borders of the model and the image, respectively. Since there may be some missing and extra segments in the model and the image as a result of segmentation, polygonal approximation, and various other reasons, we allow up to 2 extra or 2 missing segments when finding the sequences. This procedure continues until all possible sequences have been located in the data of the current cluster. This step provides the only structural information within the algorithm and cannot be omitted.

The final task to be accomplished at this step of the algorithm is to compute the rotational and translational average of each sequence that has been located. These averages are merely the averages of all of the samples that are present in each sequence. These sequence averages will be used in the final clustering step of the program.

Final Clustering: Using the sequence averages obtained from the previous step, the algorithm clusters theses values to find those sequences which lead to the same rotational and translational results. Note that in this application of clustering, we are clustering sequence average rotations and translations. In other words, we are now working at a higher symbolic level of recognition. As with the initial clustering, the program uses the iterative technique of clustering, evaluating, clustering, etc. After the value of β has reached its maximum, the program again selects the cluster which contains the largest number of sequences and passes this cluster to the final program step. While the initial clustering step had to deal with a large number of samples present in the disparity matrix, this step is much faster since we have eliminated so much data in the earlier steps.

Transformation Computation: After all clusters which were selected have been sequenced and clustered a second time, the program computes the confidence level of the transformation determined by each cluster. The confidence level is the sum of the matched segment lengths divided by the model boundary. The cluster with the highest confidence level is selected as the final transformation cluster. The program assembles the set of matched segments included in the sequences in this cluster. The final output of the program is the rotation and the vertical and horizontal translation necessary to locate the object model within the image.

III. 2-D Results

Image Acquisition & Polygonal Approximation: In order to determine the ability of the program to find objects in an occluded scene, a set of 14 models was obtained and used in the matching algorithm. The models consist of a set of tools such as a hammer, screwdriver, pliers, wrench, and so on. The model for each of these tools was obtained using a commercially available digitizer. Once the models have been obtained, they are transferred to a Vax 11/780 for the remainder of the processing. The program finds the border of each tool using a simple border follow algorithm. The polygonal approximation of the tools are then found using a curvature maxima and split-merge algorithm. The task of obtaining images to be processed proceeds in exactly the same manner as in the model acquisition. 20 images were collected for this experiment. Some examples of the images collected in this test run appear in Figure 2.

Model Based Recognition: When the clustering program was run on the 20 images that were collected, the results were very good. Of the 56 models present in these images, 47 (84%) models were correctly matched. 4 of the 56 models were mistakenly matched to a different model. The remaining 5 model instances could not be matched due to severe occlusion in the image. Figure 3 shows the matching results for the three images that were shown in Figure 2. Solid lines show the polygonal approximation of the images using the split-merge algorithm. The dashed and dotted lines show the polygonal approximation of the model at its matched location in the image. Dashed lines indicate the segments which were matched while the dotted lines show the segments which did not contribute to the matching.

IV. 3-D Applications

We have identified several areas in three dimensional scene analysis which can readily be solved using clustering techniques. The recognition method described in the previous sections can readily be extended into three dimensions. In addition, we have investigated applications of clustering on range data to find planar patches. As with the 2-D algorithm, the clustering techniques are applied in a hierarchical fashion to reduce the amount of data computation. Each of these areas will now be described briefly.

Identification of planar patches or faces on the surface of a three dimensional object using range data is one of the current areas of interest. This range data is available from several sources, including range information from a laser range finder as well as data points obtained from a CAD model of the object. The first step of this process is to compute the surface normals for every point in our data set. We first identify all the neighbors of each point and then use a least-squared method to represent the plane which approximates the local surface patch. The surface normal is then obtained from this plane. Once the surface normals are known, the use of clustering allows surface points with similar normal values to be easily grouped together. The K-means algorithm along with the performance measures previously discussed are again used in the clustering process. In this step of the algorithm, we cluster the normals of each point. Once the initial clustering has been completed, we perform an additional clustering on each of these clusters separately. Since the first step only considered the orientation of the surface normals, it is possible that non-adjacent surface patches that have similar or identical orientations on the object have been included in the same cluster. We now apply the clustering methods on the x, y, and z values of each of the data points and separate any samples in these clusters that are not spatially adjacent. Since we use the performance measures to find the inherent number of clusters, the initial clusters that contain more than one planar face are split up while the clusters that contain only one face are left intact. The algorithm also uses the least-squared procedure on each of the final clusters to determine the theoretical position and orientation of each of the planar faces on the surface of the object.

Once we know the position and orientation of the planar faces on the surface of the object, we can use this information to aid in the recognition process when we are given only one view or an occluded view of the object and a 3-D model having a similar representation. The clustering method discussed earlier can be naturally extended into three dimensions. The line length and angle between successive segments in 2-D now become the planar area and angles between adjacent planes. The disparity matrix is then formed by comparing all planes in the known model with the planes in the image. If compatible, the three dimensional rotation and translation disparities are entered in the disparity matrix.

Clustering can then be used to group sets of planar faces that have similar transformations. Sequencing in two dimensions corresponds to planar region growing in the 3-D algorithm, providing the necessary structural information. In each of the initial clusters, we mark all compatible groups of adjacent planes. We then determine the average transformation determined by each of these groups. In the second clustering stage, we cluster these average transformations to obtain the groups of planar regions that yield the same rotation and translation. Finally, we select the matching that leads to the best confidence level. The development of the 3-D algorithm is under investigation and will be reported in the future.

V. Conclusions

Based on the results presented in this paper, we conclude that the cluster-structure paradigm is a robust approach to solve the occlusion problem in 2-D. The data and the amount of computation reduce in a systematic and hierarchical manner. Since the technique does not limit itself to a single sequence of line segments on the border of an object, it can locate all of the matched segments of the model, which accounts for the high success rate. The program was not highly successful in the instances of severe occlusion, where a given model has only about 5% of the total number of segments visible in the image. In those situations, even an expert would have problems locating a model within the image. Further details about the 2-D technique can be found in [3].

In addition to the success of the 2-D work, we also believe that the application of the clustering methods to the three dimensional domain also holds good promise of success. Note that in 3-D, both matching and symbolic representation can be obtained using the same clustering techniques used in the cluster-structure paradigm. Work completed to date suggests that the conversion to 3-D should provide reliable results.

[1]This work was supported in part by NSF Grants DCR-8506393, DMC-8502115, ECS-8307483 and MCS-8221750

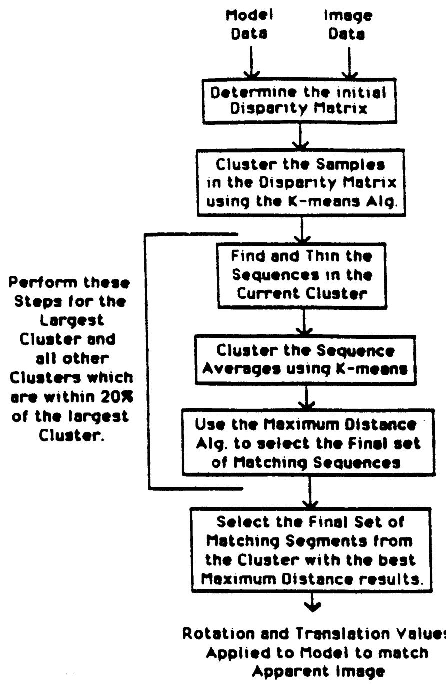

Fig. 1 Block diagram of the clustering based occlusion algorithm.

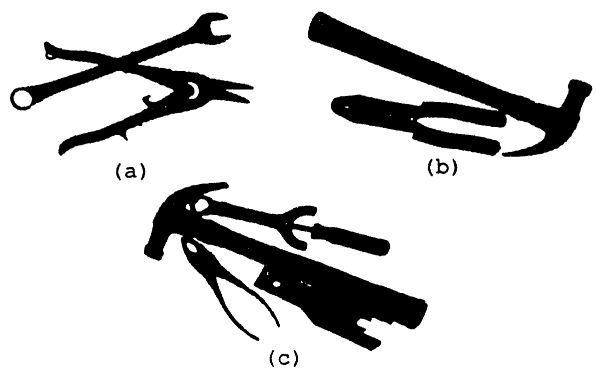

Fig. 2 Images of the occluded objects (a to c).

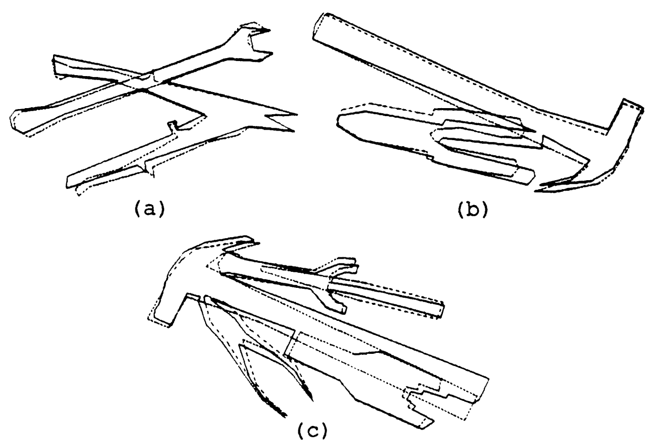

Fig. 3 Results of matching for the occluded images shown in Fig. 2.

References

[1] N. Ayache. A Model-Based Vision System to Identify and Locate Partial Visible Industrial Parts. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 492-494. June, 1983.

[2] B. Bhanu and O.D. Faugeras. Shape Matching of Two-Dimensional Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6(2):137-155, March, 1984.

[3] B. Bhanu and J.C. Ming. Recognition of Occluded Objects: A Cluster Structure Paradigm. In *IEEE International Conference on Robotics & Automation*, pages 776-781. April, 1986.

[4] R.C. Bolles and R.A. Cain. Recognizing and Locating Partially Visible Objects: The local-Feature-Focus Method. *The International Journal of Robotics Research* 1(3):57-82, Fall 1982.

[5] K.S. Fu and T.Y. Young, Eds. *Handbook of Pattern Recognition & Image Processing*. Academic Press, 1985. Cluster Analysis, Chapter by A.K. Jain.

[6] K.E. Price. Matching Closed Contours. In *Proc. 7th Int. Conf. on Pattern Recognition*, pages 990-991. July-August, 1984. Also in Tech. Report 104, Intelligent Systems Group, University of Southern California, Los Angeles, October 19, 1983, pp. 29-37.

IAPR afcet

EIGHTH INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION

PARIS, FRANCE
OCTOBER 27-31, 1986

PROCEEDINGS

Sponsor : IAPR, International Association for Pattern Recognition
Organizer : AFCET, Association Française pour la Cybernétique Economique et Technique

Volume 2

IEEE Computer Society Order Number 742
Library of Congress Number 86-81419
IEEE Catalog Number 86CH2342-4
ISBN 0-8186-0742-4

COMPUTER SOCIETY PRESS

IAPR

Bir Bhanu

afcet

EIGHTH INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION

PARIS, FRANCE
OCTOBER 27-31, 1986

PROCEEDINGS

Sponsor : IAPR, International Association for Pattern Recognition
Organizer : AFCET, Association Française pour la Cybernétique Economique et Technique