OBJECT RECOGNITION BASED ON INDEXING IN 2D ENVIRONMENT

S.K.Bose K.K.Biswas S.K.Gupta

Dept. of Computer Sc. & Eng.
IIT Delhi, New Delhi 110 016
INDIA
emai1: skbose@netearth.ernet.in

Abstract

Model-based object recognition has attracted considerable attention in the vision community over the last twenty years. This paper presents a model-based two dimensional object recognition system using feature indexing under a linear transformation through geometric hashing. The objects are represented by their dominant invariant features under a linear transformation in the normalised orthogonal coordinate frame. The goal of this paper is to solve the 2D recognition in the industrial environment in an efficient way and reduce the net computational overhead. The proposed algorithm is also able to identify a partially occluded object in cluttered environment. The paper shows how the symmetry of the object can be exploited to reduce the model data storage and to speed up the matching process. The matching is done through its feature set tuples. The feature set includes both the local and global features of the objects, which helps to recognise the object more accurately.

1. INTRODUCTION

Recognition of industrial parts and their pose determination in a factory environment is an important task whenever automatic part handling is involved. Most of such systems use a model-based approach as it is quite easy to extract the necessary geometric features from the known objects in advance. The earlier attempts to solve the problem of matching between the model and the scene data are based on a tree search, searching through all promising matches [1,2]. The computational complexity of such algorithms are of exponential order except for very few trivial scenes [3].

In a model-based object recognition system, the first task is to decide upon the scheme of representation of the objects. The representation should in general be reliable, flexible, accessible and computationally inexpensive to generate. It could either be based on global features, local features, relational structure of the parts or a combination thereof. The relative merits and demerits have been discussed in [4,5]. The features should be such that these could be extracted easily from the scene data. Object recognition is accomplished by a comparison of scene and model features through a matching process. The efficiency of matching can be increased by indexing the features and making use of a hash table [6,7].

In this paper, we consider the problem of recognizing rigid flat objects in different orientations. We exploit both the local and global features of the objects such as points of interest(e.g, corners), ratios of the length of line segments, angle between adjacent line segments and distances between centre of mass and the vertices. It is known that most of the industrial parts and tools have some symmetrical features. This property is also exploited by including the total number of skewed/reflectional symmetry axes in the feature set. The basic idea is to increase the efficiency of the matching process by operating on a small number of features (in view of, the symmetry) and thereby reducing the net computations. The models are computed in the normalised orthogonal coordinate frame to nullify the effect of scaling, rotation and translation [6].

To reduce the exponential complexity of the matching process, Huttenlocher and Ullman [8], proposed an alignment technique, where the object is first aligned with an image using a small number of pairs of model and image features, and then aligned model is compared directly against the image under a similarity transformation. Use of a small number of features to determine the position and orientation reduces the exponential complexity to order of 5th power of n, total number of feature points. An alternative approach is to use geometric hashing technique as suggested by Kalvin et.al.[9] and by Lamdan et.al. [6]. By using an appropriate indexing mechanism, the search space is reduced effectively. An improvement has been proposed by Stein and Medioni [7] by introducing the concept of gray coding which is a generalized quantization scheme. The schemes in general rely on the redundant representation of the model. We attempt to combine the local and global features of the objects(including symmetry) and obtain a similarity transformation by establishing a correspondence between a model basis pair and a scene basis pair.

Our proposed system is divided into two phases - model processing phase(off-line) and the recognition phase(on-line). In the model processing phase, we describe the object in terms of dominant features of the object. The models are stored into the database using indexing through some suitable hashing function and these models are organised in such a way that the deletion or insertion of a model is of constant time. In the recognition ph:;se, we match ant[3] verify the object with the model

TjTT-he~database to find an instance of this object, if any. The input scene is pre-processed in the first phase of the matching process to derive a representation similar to the model in the database and then generate a hypothesis by calculating maximum number of votes for the matched basis pairs of models in the model-base. The votes are calculated by finding a match between mode]. coordinates and scene coordinates. In the next phase, we verify the probable matched candidates with the scene candidates by comparing the other extracted features. The proposed recognition scheme would be suitable in the industrial environment because it is robust, fast, and computationally inexpensive, which is the most important ^requirement for an industrial vision system.

2. FRAM13W0RK FOR RECOGNITION

In this section we will discuss the general framework for the recognition scheme. A block diagram of the system is given in fig.l. To reduce the complexity of the recognition system, it is divided into two phases. Each of the phases consists of separate module!;, which can be processed in parallel.

Model-base Processing Phase

For each model object

i) extract a set of features points by taking the end points of each line segment,

ii) find the total number of skewed/reflectional symmetry of the object and find the centre of mass of the object,

iii) transform the point set into normalised orthogonal coordinate frame with respect to a particular ordered basis pair,

iv) compute the angles between adjacent line segments, length ratios of the adjacent segments, and the distances between centre of mass and the vertices,

v) index the transformed point coordinates using some hashing function and store the point under a particular attribute triplet < model, basis-pair, symmetry> attached with the other features extracted from the object.

vi) if there is no symmetry, repeat steps iii) to v) for all possible pairs, else, repeat steps iii) to v) for the possible basis pair on one side of the symmetry axis.

Recognition Phase

Given an input scene

i) follow the step i) to iv) of the model pre-processing phase for a particular basis pair picked up from the scene,

ii) hash the point co-ordinates and looX for a match wi£h model ' point co-ordinates. If the match is found in the model-base, then vote the corresponding attribute triplet <model, basis-pair, symmetry>,

iii) select the model and basis-pair **based** on the maximum number of votes cast on that attribute triplet <model, basis-pair, symmetry>,

iv) verify the input scene with the model selected in step (iii) through their other extracted features about the object.

The complexity of the overall algorithm is decided by the complexity of the matching process. The complexity is linear in most of the cases except in very complicated scenes where no scene points match with the objects stored in model-base.

3. OBJECT RECOGNITION

The model-based object recognition problem can be defined as a problem of identifying the correspondence between a part of the object and a model in model-base. The major components of the system discussed in the previous section are **feature attraction, object representation and matching.**

3.1 Feature Extraction

In this section, we will discuss about the features required for representation and procedures for their extraction from the object. The aim is to extract geometric features of the objects which remain invariant under similarity transformation. The important features for our system are the line segments, angles between adjacent line segments, length ratios of two adjacent segments, distances between centre of mass and the contour points, and the total number of skewed/reflectional symmetry axes. Therefore, a five tuple feature set is needed to represent the object for this scheme. The feature tuple set exploits both the local and the global properties of the objects and helps in efficient and accurate recognition of the objects.

The line segments are extracted by grouping the edge points in the image, by appropriately merging and discarding wherever necessary. Once the line segments are computed, we find the adjacent segments for computing the angle between them and their length ratios.

To compute the distance between centre of mass and the vertices, we need to compute the moment of the object. The (p,q)th moment of the image function f(x,y) is defined as

$$m_{pq} = \sum_x \sum_y f(x,y) . x^p . y^q ,$$

Therefore, the centre of mass $(\bar{x}, \bar{y})$, will be

$$\bar{x} = m_{10} / m_{00} , \quad \bar{y} = m_{01} / m_{00} .$$

3.1.1 Symmetry Axis Computation

Symmetry is a prolific phenomenon in the world [10]. It may be defined in terms of three linear transformationsin n-dimensional Euclidean space E" : reflection, rotation and translation. Formally a subset S of E*is symmetric with respect to a linear transformation T if T(S) = S. The symmetries can be classified into two broad categories

: skewed/reflectional and rotational symmetry. Here we shall concentrate on reflectional symmetry.

The reflectional symmetry has a reflectional plane, for which the left half space is a mirror image of the right half. But the reflectional symmetry is actually generated from the skewed symmetry [11]. A "skewed symmetry" is a planar point pattern such that (x,y) exists, iff (-x,y) exists. The x-axis is called the "skewed transverse axis" t; whilst the y-axis is called the "skew symmetric axis" s (fig.2). If t and s are not orthogonal, then we have a skew symmetry. If they are orthogonal, then the skewed symmetry degenerates to a real (reflectional) symmetry, the reflectional axis of which is s.

One of the most important measure of the skewed/reflectional symmetry is

$$M = \begin{bmatrix} m_{10} & m_{11} \\ m_{11} & m_{02} \end{bmatrix}$$

where m&b = Z (^{x}K* » y_{k}^{b}) > summed over all points ^""(x^iY*,) € **object.**

The matrix of moments for a symmetric object is a diagonal matrix. The moment itijj is necessarily equal to zero for a symmetric object, since for every point (x,y) adding the quantity xy to the moment, there is another point (-x,y) adding -xy to the moment. The condition m^ = 0, is a necessary but not sufficient property of symmetric objects. The skewed symmetric axis may be defined as the locus of the midpoints of the mapping pairs of a symmetric object. The total number of skewed symmetric axts can be obtained through the following algorithm [12].

Algorithm

The input to the algorithm is the point set extracted from the scene.

1. oC •— 0; { the initial mapping direction parallel to the x-axis }

2. Clear the mid point array;

3. Rotate the point set by - oO about tht origin; { after the rotation; the mapping direction is parallel to x-axis }

4. For all lines L parallel to the x-axis do
 if two points u,v lies on the line L then find the mid point and store into midpoint array;
 endif;
 endfor;

5. Use a Hough transform to find straight lines in mid point array;

6. Increment *oO* ;

7. If dc > Tt , then exit else go to 2.

For &. curved object, the symmetry axis can be found by approximating the curve to a polygon.

3.2 Object Representation

As stated earlier we concentrate only on 2D object models. Our goal is to provide a similarity invariant representation of the model. Models are represented in a normalised orthogonal coordinate frame using the end points of a line segment in addition to other extracted features. The coordinate frame is defined uniquely by assigning the coordinates (0,0) and (1,0) to the first and second point respectively [6]. The vector joining the pairs is referred as the x-axis, and the y-axis is selected automatically as an axis perpendicular to the x-axis in the counterclockwise direction. We refer to this as a basis pair although it does not span the complete 2D plane. So in the model/image, with (0,0) and (1,0) chosen as a basis pair, the rest of the segments will be calculated with respect to this basis pair.

Once the line segments are computed with respect to a new basis, we index those segments using some hashing function and then store this information and other information related to the line segments into a record and store the record against the index key in the hash table. If the objects are not at all symmetric, we compute the model with respect to all possible basis pairs. But since most of the objects have some underlying symmetry, so the computation will be done only with respect to the possible basis pairs belonging to just one **side** of the symmetry axis. Therefore the total number of computation and storage is reduced significantly. Even if the objects are overall asymmetric, we try to partition the object to get some symmetry in the parts. The above representation is redundant, which helps in recognition of the partially occluded objects and the objects in cluttered environment. The worst case complexity, (where the objects are not at all symmetric) will be of the order of m-cube, where m is the total number of points. However, generally in most of the cases the complexity is less than the half of the worst case.

3.3 Matching

The general problem of matching can be defined as finding a set of features in the given image that approximately matches a model in the model-base. The choice of matching process is highly dependent on the type of model used for object representation. Here we combine both the local and global matching technique, to be able to recognise the object in a more efficient way. The local matching is done based on line segments, angles, and the length ratios whereas the global matching is done based on the symmetry information and the distances between centre of mass and the vertices.

In the first stage of matching, we extract the feature set from an input scene, as discussed in the last section. A line segment is picked up and end points are chosen as a basis pair and rest of the line segments ere recomputed with reference to this basis paiir. Now each of the line segment is matched with model line segments in the model-base. It will try to find a match with some: model segments for the input segments. If the match is found, then a vote is cast to the respective model basis pair. If no match is found, another basis pair is picked up and previous step is repeated. At the end, we select the basis pair and model based on the maximum number of votes as our probable matched candidate.

In the next phase we verify our matching through angle information, length ratio, distances between centre of mass and vertices, and from the number of symmetry axes information. Once the correspondence of basis pair is established in the model-base, the similarity transformation can be calculated from the correspondence of basis pair and also can find the mapping of line segments. In title matching process, most of the processing can be done in parallel since they are mutually exclusive. The matching time will be considered as actual recognition time. The worst case complexity of the sequential algorithm is of $O(n^3)$, n is the total number of end points. But in the average case, the complexity is op linear order.

4. RECOGNITION UNDER AFFINE TRANSFORMATION

In the similarity transformation based model-base recognition scheme, the viewing angle of camera is the same for both the model and the image of a scene. This condition can be achieved in a factory environment where the viewing angle of a camera on a conveyer belt can be kept fixed. But the more general affine transformation can be approximated to the perspective projection where the viewing angle may vary. The features we have extracted for the similarity transformation, are . also invariant under affine transformation. In the affine domain, the recognition of objects are more accurate and it has greater flexibility on the object as well as on the viewing direction. The affine transformation is uniquely identified by three non-collinear points. If eoo, ejo , e^ be an ordered affine"basis triplet in the 2D plane, then the affine co-ordinates (x,y) of a point P are

$$P - x(e_{10}-e_{0c}) + y(e_{01}-e_{06}) + e_{06} .$$

Apply, the affine transformation T on P,

$$TP = x(Te_{10}-Te_{00}) + y(Te_{01}-Te_{00}) + Te_0o \bullet$$

Therefore TP has the same co-ordinate (x,y) in the basis triplet Te_{00}, Te_{10}, Te_{01}.

The complexity of both the phases will increase since three points are required to identify the affine transformation. If there are n^4number *of.* points, the complexity will be $\Theta(n)$.

5. EXPERIMENTAL RESULTS

Ths model-base is constructed in off-line phase. We assume that the models can be acquired under 'ideal' conditions (e.g., from a CAD model); hence the model-base processing step involves noiseless object scenes. To demonstrate the effective use of our method, here we have included some examples. At present, we have experimented only upon the line drawing objects (straight edges and curved edges both), see fig.3, fig.4. But our method is also equally useful for the gray scale images.

It has been observeAthat if the objects have at least one skewed symmetry, the total number of points computation with respect to the different ordered basis pair reduces significantly which helps to improve the recognition speed as well as the storage requirement for the model-base.

The above result is immediately verified by taking the example in fig.3 or fig.4. In fig.3, the total number of interest points are 6, therefore, according to the previous geometric hashing schemes [6,7], the total number of basis pairs will be 30 and a total of 180 points will be needed to be computed for representing the model. But in our approach, due to the two symmetry axes SP and AD, the maximum number of basis pairs will be 18 and total of 72 points will need to be computed to represent the model. The above representation is still redundant but the total number of computations required are less than the half of the previous method. A similar result also hold* for fig.4. This significant reduction of computation demonstrate the usefulness of the approach.

The total number of computation can still be reduced further, since the points of interest in this approach are the end points of a line segment, so, while selecting basis pair we can put a constraint (xj ,yj)>(xj ,yj)» in such a way the total number of computation again can reduce significantly.

The system can be made faster, if we assume that the objects are of convex type. In the convex domain, the partially occluded object may be identified very easily by identifying the concavity entrances of the objects. Experiments are supported by SUN 3/4 machine and it is done on the large class of objects. The scheme has shown successful results even in the case of partially occluded objects and this would be more suitable in the industrial environment, where the most of the objects handled are symmetric. The disadvantage of this approach is that, it relies on the low level processing and the quality of images. It may be improved upon by incorporating a domain knowledge of the objects.

6. CONCLUSION

This paper demonstrates the basic idea of indexed based object recognition in 2D environment. Here we have presented a method based on invariant geometric features under a similarity transformation. it also discusses about the problem in affine domain. The similarity transformation requires two points basis which is sufficient for 2D scene and it can allow rotation, translation, and scaling, but the -three points basis is required: for affine transformation which closely approximates the perspective projection. The advantage of this approach is that it is independent of scale, orientation, and position. It solves the problem in polynomial time and updates the data-base in constant time. The salient feature of the algorithm is the use of symmetry in recognition which significantly reduces the storage, computational cost, and also prune search space. Our algorithm is successfully tested for line drawing data and also the experiment on gray level images is going on.

REFERENCES

[1] T.O.Binford, "Survey of model-based image analysis systems," Int. J. Robo Research 1,1, 1982(spring), pp. 1-63.

[2] R.T.Chin and C.R.Dyer, " Model-based recognition in robot vision,"ACM Comput. Surveys." vol.18,1986, pp.67-108.

[3] W.E.L.Grimson and T.Lozano-perez, "Localizing overlaping parts by searching the interpretation tree," IEEE PAMI-9,1987,pp.469-482.

[4] T.F.Knoll and R.C.Jain,"Recognizing Partially visible objects using feature indexed hypothesis,"IEEE J. Rob, and Automat. RA-2,1986,pp.3-13.

[5] J.L.Turney, T.N.Mudget and R.A.Volz, "Recognizing partially occluded parts," IEEE PAMI-7,1985,PP.410-421.

[6] Y.Lamdan, J.T.Schwartz and H. Wolf son, "Affine Invariant Model-based Object Recognition,"IEEE Trans. on Robo. & Automat., vol.6,No.5,1990,pp.578-589.

[7] F.Stein and G.Medioni,"Efficient two dimensional object recognition,"in Proc. 10th Int. Conf. on Pattern Recognition," June.1990,PP.13-17.

[8] D.P.Huttenlocher and S.Ullman,"Object recognition using alignment,"in Proc. 1st ICCV(London),1987,pp.102-111.

[9] A.Kalvin, E.Schonberg, J.T.Schwartz and M.Sharir,"Two dimensional, model-based, boundary matching using footprints,"Inti J. of Robo.Res.. 5(4),1986,pp.38-55.

[10] K.A.Stevens,"Surface Perception from local analysis of texture,"AI-Tech, Rep-512, AI.Lab.,MIT, Cambridge,1980.

[11] S.A.Friedberg,"Finding axes of skewed symmetry,"CVGIP,34,1986,pp.138-155.

[12] S.Y.K.Yuen,"Shape from Contour using symmetries," Cognitive Sc. Res. Paper, Sl.no.141, University of Sussex.

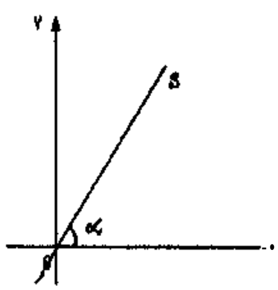

Fig.2. The skewed symmetric axis s passes throgh 0 (considered as centre of mass). The skewed transverse axis t is the x axis

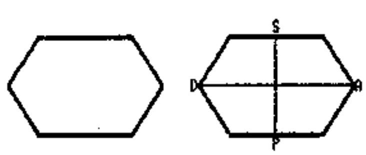

Fig.3. Hexagon with two symmetry axes.

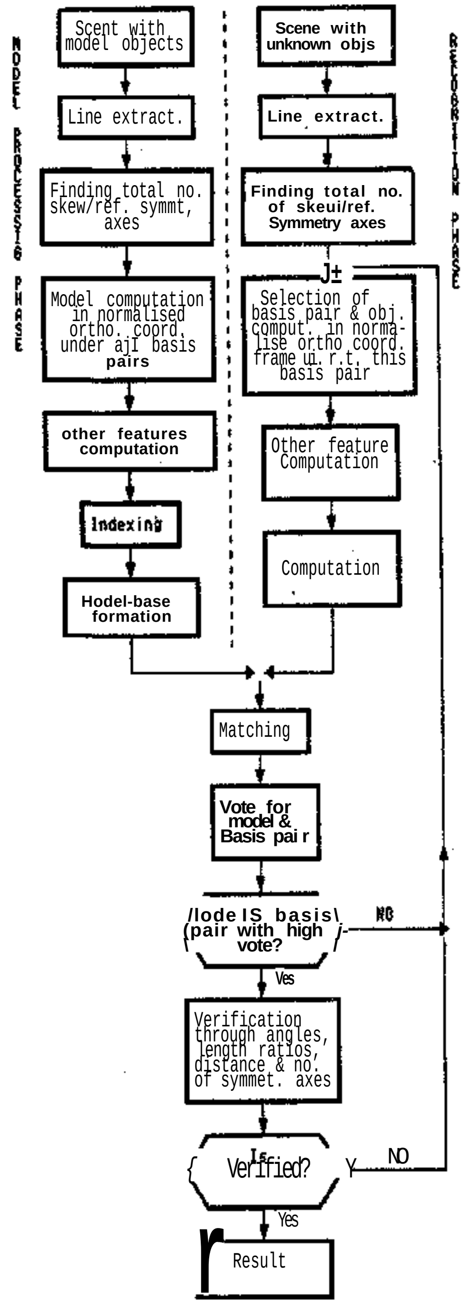

Fig.1. General Indexed Based Recognition Paradigm.

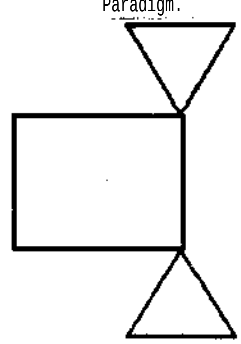

Fig.4. scene of composite objects consists of 3 parts. Each is symmetric.