IAPR Workshop on CV - Special Hardware and Industrial Applications OCT.12-14, 1988, Tokyo

RECOGNITION OF OCCLUDED-OBJECTS BY LOCAL FEATURE SEQUENCING

Jianming SONG & Paul GAILLARD

U.A. CNRS No. 817. Université de Technologie de Compiègne
B.P. 649 60206 Compiègne FRANCE

ABSTRACT

We present a new method for 2-D occluded-object recognition. In this method, objects are segmented then represented by local and (rotation-translation) invariant feature-vectors. A decision tree is used to match similar local features, which allows an efficient and complete selection of potential models. In order to validate each potential model, we explore the geometric relationships between feature-vectors using a feature sequencing process. An example of a 2-D application is given to show the effectiveness of the method.

Key words : Occluded-objects recognition, Local feature extraction, Contour matching, Local feature sequencing.

I. INTRODUCTION

The task of recognition in computer vision consists of identifying and locating an unknown object in an image by matching the object with a set of possible objects, i.e. models.

The techniques of recognition can be classified as global-feature-based or local-feature-based[1-5]. The former approaches are only applicable to isolated-object recognition, while the latter ones are designed for occluded-object applications.

One of the difficulties in object recognition is caused by occlusion. Occlusion occurs when two or more objets touch or overlap one another. Since occlusion will be present in most industrial environments, the recognition of partially visible objects is of great importance for industrial applications.

Many methods have been proposed to solve the occluded-object problem. They can be classified according to either the choice of local-features or to the technique used to perform the matching.

The commonly used local features are subsets of the object's boundary, holes or corners[4,6].

The matching techniques can be divided into correlative techniques and structural ones. The correlative techniques, such as boundary matching, are based on the matching of similar local features[1,2], while the structural ones like relaxation make use of local features as well as their geometric relationships[4,5,7].

The structural techniques are computationally intensive because of the use of combinations of local features, while the correlative methods are often invalid because of the limited choice of available discriminant local features.

Attempting to overcome these difficulties, this paper proposes a method which performs the recognition process in two steps : a selection of potential models and a feature sequence matching.

By first matching similar local and invariant features, all potential models within the model set can be selected. A decision tree is used to speed up the selection process. A process of feature sequencing is then applied to evaluate the score of matching for each selected model. Finally, an estimation of transformation parameters allows to locate the recognized object in the scene.

The paper is organized as follows: Section II presents the principle of the method. Section III gives an experimental result in a 2-D application, which illustrates the effectiveness of the method. Section IV concludes with a discussion on further research directions.

II. PRINCIPLE OF THE METHOD

On the basis of previous discussion, we note that the keys which determinate the performance of a recognition method are the local features employed and the matching technique. Both the choice of local features and the matching technique are constrained by the specific applications, that is, the models set, the lighting condition, the computation time, and so on. Our task is therefore to find the optimal local features and an appropriate matching technique for a given application.

§II-1. Choice of local features:

We make use of the object's boundary for the recognition. In order to build a model or a scene description, the object's contours, after having been extracted from the image scene, are approximated by a group of linear segments. Such a polygonal approximation gives a compact description of an object's boundary and allows a considerable reduction of data size. The local features are then computed using the polygon obtained.

As an occlusion could occur at any portion of a contour, the local features chosen should be dispersed homogeneously over the whole contour. The feature value must be invariant with respect to the translation and rotation of objects in the scene.

For a given polygon of n segments ($S_1, S_2, \dots, S_n$), each segment being characterized by the length L and the direction D, we associate the ith segment S_i with a feature-vector of two components $\mathbf{P_i} = (L_i, A_i)$, where

$L_i = L_i$ is the length of the segment S_i ;

and

$A_i = | D_{i+1} - D_{di} |$ is the angle between two adjacent segments S_i and S_{i+1}.

We can easily verify the locality and the invariance of the features with respect to the rotation and to the translation of an object (Fig.1).

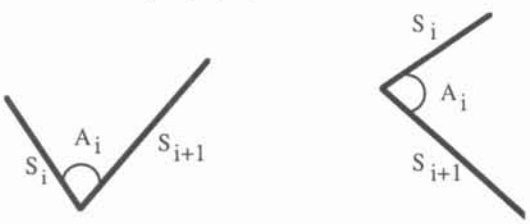


Fig. 1 Local feature

The above operation translates the polygonal description of an object contour into a sequence of local and rotation invariant feature-vectors.

As a feature-vector is composed of two components, L and A, any feature-vector can be represented as an element in a 2-D matrix representing the feature space(see Fig.2a). In such a matrix, the matrix lines are associated to the different values of the feature L, and the columns correspond to those of the feature A. The matching of similar feature-vectors can therefore be carried out by locating the model feature-vector at the appropriate line and column in the matrix containing all of the reference feature-vectors. The resolution of the matrix is defined as the value interval within two adjacent lines (or columns).

One way of representing efficiently a 2-D matrix is the quadtree structure, which allows a compromise to the size and the resolution of the matrix represented. In a quadtree, the root stands for the whole region of the matrix, while each of its four nodes corresponds to one of the four subdivided regions. Fig.2b illustrates the use of a quadtree structure for representing model feature-vectors.

The advantage of using such a representation is that the same quadtree structure can be used as a decision tree to match similar local features.

§II-2. Model selection ;

This step performs a selection of potential models from all of the models in the data base.

We begin with the sequence of feature-vectors ($\mathbf{P_1}, \mathbf{P_2}, \dots, \mathbf{P_n}$) computed from the unknown object's contour. In order that all of the potential models be selected, we must search for every model feature-vector that matches one of the vectors among ($\mathbf{P_1}, \mathbf{P_2}, \dots, \mathbf{P_n}$).

To match feature-vectors, the quadtree structure containing all of the model feature-vectors is used as a decision tree. In each node, two thresholding values are stored for region subdivision (cf. Tl, Ta, Tl' and Ta' in Fig.2a).

The search for the similar feature-vectors in the model set is performed by proceeding a tree traversal for each feature-vector in the set ($\mathbf{P_1}, \mathbf{P_2}, \dots, \mathbf{P_n}$). The tree traversal starts from the root of the quadtree. It is carried out by a series of comparisons between the feature-vector $\mathbf{P_i}$ (i.e., L_i and A_i) and the thresholding values at each node visited during the traversal. The similar model feature(s) is (are) obtained when a terminal node(or leaf) is reached.

With the consideration of the feature variance, the path of the tree traversal is not unique. In fact, at each node visited, we calculate the difference between the feature value and the corresponding threshold. If the difference is less than the variance, we continue the traversal by taking simultaneously two or four

The model features found are used for potential model selection.

For each model selected, a match score is computed with the following formula :

$$\text{Score} := k_1*(N_m/N_t) + k_2*(N_s/N_m) + k_3*(L_m/L_t)$$

where N_t denotes the total number of the model's feature-vectors; N_m is the number of the matched feature-vectors; and N_s that of the successive feature-vectors; while L_t and L_m denote the length of N_t and N_m feature-vectors respectively; k_1, k_2 and k_3 are weighting factors with

$$k_1 + k_2 + k_3 = 1.0$$

Only the models with a high match score will be considered.

§II-3. Exploitation of geometric relationships ;

Since the matching based on local feature comparison provides no information about the geometric structure of the models, there could be some incoherences between the matched local feature(i.e. feature-vector) pairs. That is, one model feature could have been matched to several objects, and vice versa. In order to eliminate such incoherences, we make use of an criterion of continuity which provides information about geometric relationships between adjacent local features: By sequencing matched local features, both the matched features of the selected model and those of the object are ranged in the same order than that of theirs correspondent segments in the polygon. Sequences of successive features are then located, and the longest sequences found are validated by priority. In this way, any mismatching made to the pairs of local features within the sequences can be excluded. The sequencing proved to be a powerful criterion, since the probability of mismatching a feature sequence is much smaller than that of mismatching an isolated feature pair.

The match score defined by the formula(1) is again applied to estimate the matching quality between the reference model and the object analyzed. It is calculated after the newly validated feature pairs. If the match score is high enough, the object is considered to have been recognized. Otherwise, the object will be rejected.

§II-4. Estimation of transformation parameters:

The final step of the analysis processing is to estimate the transformation parameters between the reference model and the recognized object. Since each pair of matched feature allows to calculate a vector of transformation parameters (θ, tx, ty), the final estimation of the transformation can be obtained by a weighted-average algorithm.

Provided that there are m pairs of valid local feature-vectors, with (θ_i, tx_i, ty_i) denoting the transformation parameters for ith pair, the final estimation for rotation angle is defined as :

$$\theta = k_L \times \theta_L + k_S \times \theta_S \qquad (2)$$

with $0<k_L<1$, $0<k_S<1$ and $k_L+k_S=1$ as weighting factors.

where $$\theta_L = \frac{\sum_{i=1}^{m} \theta_i L_i}{\sum_{i=1}^{m} L_i}$$

represents the term taking account of the feature segment length,

and $$\theta_S = \frac{\sum_{i=1}^{m} \theta_i C_i}{\sum_{i=1}^{m} C_i}$$

is the term in favour of the successive features, with C_i the score assigned to a feature sequence of i successive features.

The estimation for tx and ty is made in the same way.

III. EXPERIMENTAL RESULTS

The system described above was implemented on a VAX-11/750 computer. A GRINNELL-270 image processing system was used for image acquisition, preprocessing and image storage. In the system, each image contained 512x512 pixels of 256 grey levels.

The system's input is a real image of 256 grey levels, with dark objects placed on a bright background. The following sequence of operations is applied to the original image:

- If the contrast of the input image is not high enough, a Sobel operator is first applied. The resulting gradient image is then thresholded. A thinning procedure [8] is finally applied yielding the object thin contour.

- If the contrast is high enough, a global thresholding algorithm can be directly used for contour extraction. The thresholding value is computed from the histogram distribution of original image. The result is a binary image, with black objects on a white background.

- The previous contour image (or binary image) is followed using an eight-neighbor chain code (Freeman's code).

- The coded contour is approximated by a sequence of linear segments from which we calculate the corresponding local features.

- The obtained local features are analyzed using the proposed method.

About ten miscellaneous objects were used as models for testing the proposed method. Some of them are shown in Fig. 3. Fig.4 shows a scene containing occluded-objets. The object contour extracted is illustrated in Fig. 5. And the recognition result is shown in Fig. 6.

V. CONCLUSIONS

We have presented a method for recognizing 2-D occluded objects. The principle is based on invariant local features matching between reference models and the analyzing object(s). The recognizing process is performed in two steps: (1) A selection of potential models is made by comparing compatible local features. A quadtree structure is employed to store model features and to accomplish feature comparison. (2) After the model selection, each candidate model is compared to the analyzing object(s) by matching the feature sequences. A continuity criterion is used in favor of successive feature sequences. Finally, all matched (model-object) features pairs are used to estimate the transformation parameters.

The proposed method is universal, since the local-feature-based selection gives a wide choice of potential models. It allows to generate all possible hypotheses. The method is efficient, as it takes account of the order of local features in the sequence. The advantages are shown by the developed vision system.

In any case, the algorithm of polygonal approximation remains the key factor for a successful recognition.

Further research consists of improving the prototype system and extending the method to the recognition of objets with variable scales.

Although the method is proposed for 2-D object recognition. It can be extended to 3-D recognition by assuming that each 3-D object is placed in a few stable attitudes, with each stable attitude being considered as a distinct model.

REFERENCES

[1]. W.A.Perkins "A Model-Based Vision System for Industrial Parts" IEEE. Trans. on Computers, Vol. C-27, No.2, Feb. 1978.

[2]. J.L.Turney "Recognizing Partially Occluded Parts" IEEE. Trans. on PAMI. Vol. PAMI-7, No. 4, July, 1985.

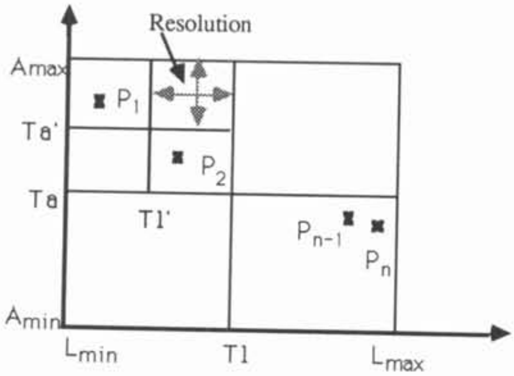
[3]. N.Ayache "HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objects" IEEE. Trans. on PAMI. Vol. 8, No.1, Jan., 1986.

[4]. T.F.Knoll "Recognizing Partially Visible Objects Using Feature Indexed Hypotheses" IEEE Journal of Robotics & Automation, Vol.2,No.1, March,1986.

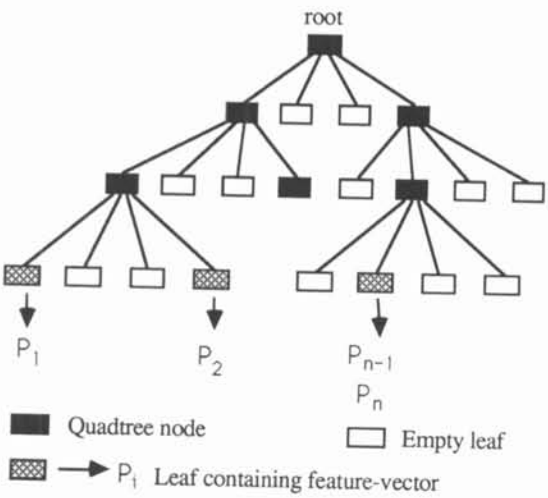
[5]. R.Freytag et al. "Development Tools for a Model Directed Workpiece Recognition System" Pattern Recognition Vol.19, No.4, pp.267-278, 1986.

[6]. R.C.Bolles "Recognizing and Locating Partially Visible Workpieces" Proc. PRIP 1982. 498-503.

[7]. B.Bhanu and O.D.Faugeras "Shape Matching of Two-Dimensional Objects" IEEE. Trans. on PAMI. Vol. PAMI-6, No. 2, March, 1984.
[8]. N.J.Naccache and R.Shinghal "SPTA: A Proposed Algorithm for Thinning Binary Patterns" IEEE. Trans. on SMC.Vol.14,No.3,May/June,1984.



(a). The feature space



(b). The quadtree structure

Fig. 2

Fig. 3

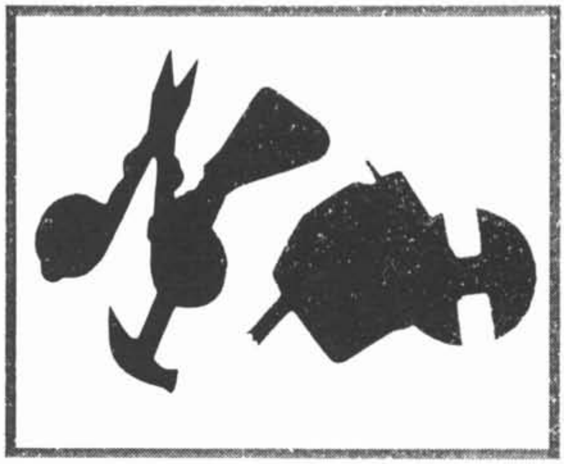


Fig. 4

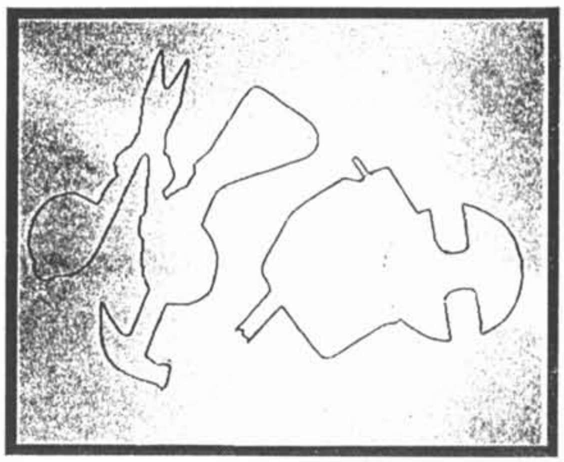


Fig. 5

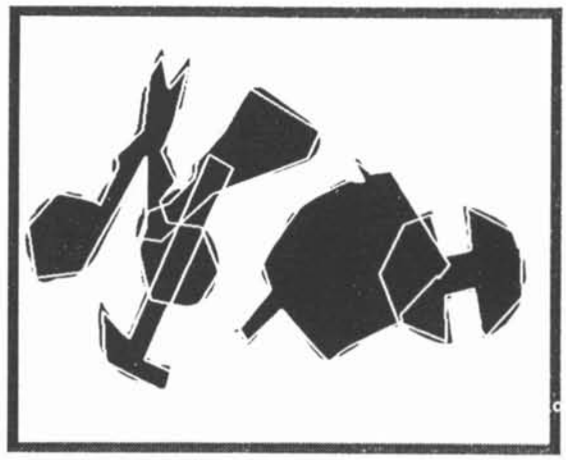


Fig. 6