HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objects

NICHOLAS AYACHE AND OLIVIER D. FAUGERAS, MEMBER, IEEE

Abstract—**A new method has been designed to identify and locate objects lying on a flat surface. The merit of the approach is to provide strong robustness to partial occlusions (due for instance to uneven lighting conditions, shadows, highlights, touching and overlapping objects) thanks to a local and compact description of the objects boundaries and to a new fast recognition method involving generation and recursive evaluation of hypotheses named HYPER (HYpotheses Predicted and Evaluated Recursively). The method has been integrated within a vision system coupled to an indutrial robot arm, to provide automatic picking and repositioning of partially overlapping industrial parts.**

Index Terms—**Computer vision, occlusions, robotics, scene analysis, shape recognition.**

I. INTRODUCTION

COMPUTER VISION is an important field where roughly two somewhat conflicting tendencies can be identified. On the one hand, a very strong demand for applications implies that performant solutions to concrete problems have to be quickly developed. On the other hand, there is a very natural desire to understand human vision as a problem in itself, hoping that this will result in the development of a general methodology for solving computer vision related tasks.

One may argue that many applications are either not sufficiently representative of the whole set of vision problems or that the people who solved them did not bother identifying the general methods that could be used elsewhere. On the other side of the road, vision theoreticians can often be reproached not to always be enough concerned with the implementation of their findings on "reasonable" hardware executing "reasonable" code.

Three main problems can be identified in computer vision. The first is the construction from sensor output of a symbolic description where information necessary to solve the problem at hand is explicitely represented. The second is that of the representation of *a priori* knowledge. This "world model" is generally very complex and few things are known about ways of representing and organizing the corresponding database. The third problem is that of using these two structures to achieve the task.

Of course, there are many relationships between these three problems. Nonetheless, separating them allows us to identify a number of potential bottlenecks. Problem I is mostly a signal processing problem, problem II mostly a knowledge representation problem, and problem III mostly a control strategy problem. Their complexity can be defined in terms of a number of parameters such as signal quality of the sensor output, how many and how different are the objects or phenomena that can be observed, and what type of *a priori* information is available.

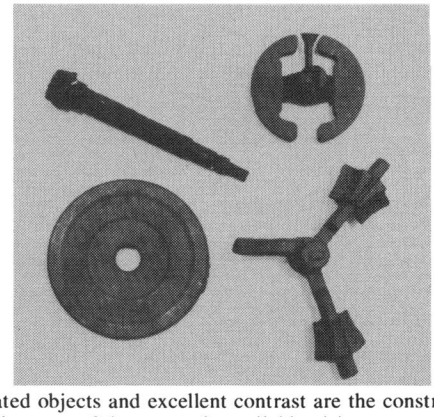


Fig. 1. Isolated objects and excellent contrast are the constraints required by most of the currently available vision systems.

We hope that if we fix one or several of these parameters and make the others vary in a controlled manner, we shall be able to outline a methodology for solving the corresponding problems in a large variety of situations. This has been our approach.

We present in this paper the methods we have developed in order to solve a very specific problem, that of analyzing scenes with randomly oriented and partially occulted industrial parts. These parts are assumed to be "flat," i.e., one of their dimensions is small compared to the other two. If we attempt to characterize this task in terms of the above parameters, it is clear that depending on signal quality, problem I may or may not be simple. On the other hand, the *a priori* information about the objects is of a quantitative geometric nature and can be made as accurate as needed; as a direct consequence problems II and III should be simpler.

This task has been tackled by several authors and is solved in a limited way by some commercially available systems. Such systems typically deal with isolated objects with excellent lighting conditions (see Fig. 1) making problem I very simple. Silhouettes of objects are usually extracted by a simple luminance thresholding followed by a connectivity analysis. Silhouettes are then represented

Manuscript received July 10, 1984; revised June 27, 1985. Recommended for acceptance by S. W. Zucker.

The authors are with the Computer Vision and Robotics Group, INRIA, 78153 Le Chesnay Cedex, France.

IEEE Log Number 8405790.

0162-8828/86/0100-0044$01.00 © 1986 IEEE

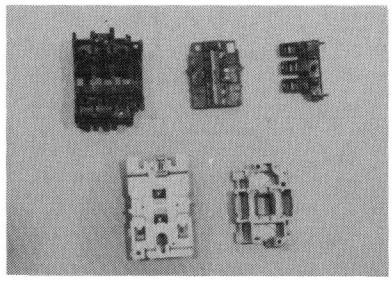


Fig. 2. Metallic and plastic objects often produce unpredictable highlights.

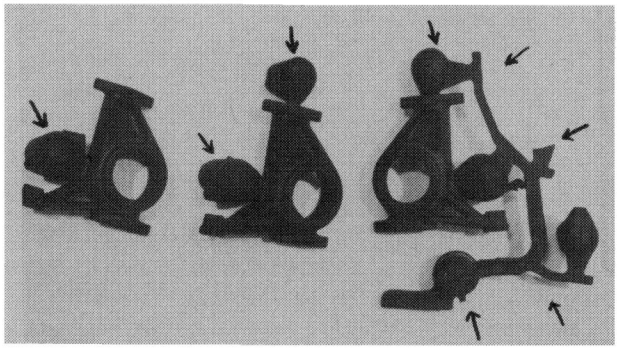


Fig. 3. Before their final process, foundary castings often have sprues and dead-heads (→) whose number and size are quite variable.

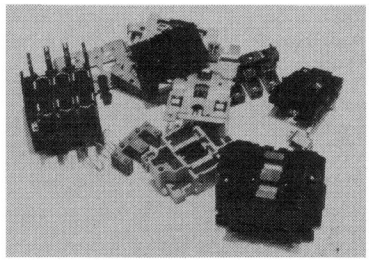


Fig. 4. Touching and overlapping objects.

by a few global numerical features making problem II also very simple, and the recognition and positioning problem is solved by nearest-neighbor techniques in feature space (for a good overview of existing industrial Vision Systems, the interested reader is referred to [1]).

A strong limitation of such systems is that they cannot handle partial alterations of the observed silhouettes which can be due, for instance,

1) to uneven lighting conditions including variations of contrast, shadows, or highlights (see Fig. 2),

2) to the occurence on the objects of sprues or dead-heads, whose sizes and shapes can vary much (see Fig. 3),

3) to the occurence of touching and overlapping objects (see Fig. 4).

More sophisticated systems can so far be found only in the laboratory. They can usually deal with objects under poor lighting conditions, thanks to sophisticated edge detection techniques, and tolerate partial occlusions by using structural representation of objects and more elaborate symbolic matching techniques. The systems developed by Perkins [2] and Dessimoz [3] are based upon cross-correlating the tangent angle or the curvature as functions of the curve length between the scene description and the database of models. They have produced good results on complex industrial scenes, but the preprocessing (segmentation) is expensive and both methods are not well suited to scale variations. Another approach is that of Rummel [4] and Hattich *et al.* [5] who have developed systems based on a representation of objects with line segments, corners, and circular holes. Model primitives are then matched with scene primitives with an A^* tree-search algorithm. Basic limitations of these systems are the combinatorial explosion when the number of primitives increases and the unability to deal with scale variations. A third approach is that of finding maximal cliques in representation graphs as pioneered by Ambler *et al.* [6] and further improved by Bolles and Cain [7]. A basic limitation here is the very large complexity of the clique finding problem. A last approach developed by Davis [8], Bhanu and Faugeras [9], and Ayache and Faugeras [10] is based on the use of relaxation techniques. Objects and scenes represented by relational graphs and subgraph isomorphisms are searched for. A basic limitation is the very large complexity of the relaxation algorithm. A fourth approach is that of the PVV system of Lux and Souvignier [11] which uses two modules implemented as coroutines: a description module extracts features in the image in a top-down or bottom-up mode and a prediction and verification module that interprets features produced by the other module in terms of a data base of models. More recently, Segen [12], Turney [13], and Grimson and Lozano-Pérez [14] proposed new approaches to the problem.

The approach described in this paper is based upon matching simple descriptions of the scene and the models by a technique called HYPER (HYpotheses Predicted and Evaluated Recursively) of hypotheses generation and verification coupled with a recursive estimation of the model to scene transformation [15]–[17]. It is fast, accurate, robust to noise, and can deal with scale changes. It is also general in the sense that it is basically independent of the kinds of primitives used to represent the 2-D shapes and in the sense that it can be extended without too much difficulty to the corresponding 3-D problem [18].

In the next section we describe how models and scene descriptions are built, i.e., what kind of primitives are used in our representation and how we compute them from the input image. We then describe the matching process that identifies models in the scene description and estimates the corresponding geometric transformation. An analysis of the complexity of the corresponding algorithms is then presented, and we conclude with results obtained from a number of difficult scenes.

II. Building Models and Scene Descriptions

Our system is designed to handle objects with one dimension much smaller than the other two, that is flat or almost flat objects. Partial occultation is allowed, and no special care is taken of the illumination, i.e., the system is capable of working under poor lighting conditions. The acquisition device is a cheap standard Vidicon camera

connected to an image memory. The video signal is typically quantized using 256 grey levels and the image size is either 256*256 or 512*512.

One key feature of the system is that models and scenes are represented the same way. This makes life a lot easier for the matching procedures we describe later.

In order to build a model or a scene description, the following sequence of operations is applied to the picture of the isolated object or of the scene:

1) if the contrast is high enough (i.e., if the lighting conditions are perfectly controlled), threshold the image, smooth the resulting binary picture using erosions and dilations [19].

2) if the contrast is not high enough (general lighting conditions), find the edges by combining gradient and second order derivative information [20], [21]. A Sobel operator is first applied to the image and the result is thresholded yielding the major intensity discontinuities with the standard problems of contours which are not connected and of width larger than one pixel. Second, the picture is low-pass filtered with two filters of different sizes (in the current implementation we use 7*7 and 3*3 arithmetic averages). The results are subtracted and zero crossings detected. This produces a very accurate detection of all intensity discontinuities. Edges are connected and of width one pixel. By following edges in parallel in the two images, we can eliminate those corresponding to low contrast variations while keeping the connectivity high.

3) find the list of connected border points [22].

4) approximate the connected components with polygons [23].

Shapes of 2-D objects are therefore represented by polygonal approximations of their borders. This description has several advantages which are as follows.

1) It is local, meaning that different parts of the objects are described independently of each other, allowing for independent identification.

2) It is compact, meaning that most objects can be accurately described using a small number of line segments (typically less than 100).

3) It is general, meaning that it can be applied to any planar shape.

4) It is sensitive to variations in the position and orientation of the objects and allows to recover those parameters accurately.

5) It is simple, meaning that the operations used to go from the image to the description are straightforward and fast; most of them can be executed in fractions of a second on commercially available equipment.

Fig. 5 shows the silhouettes of two mechanical parts used in the French car industry. These parts are foundry castings. Fig. 6 shows the model description associated with these silhouettes and with their symmetric homologues. The number of segments involved in these description ranges between 39 and 50. The contrast conditions are very good and allow for the use of the first method.

Fig. 2 shows some of the parts of an electromechanical device made by TELEMECANIQUE; Fig. 7 shows the

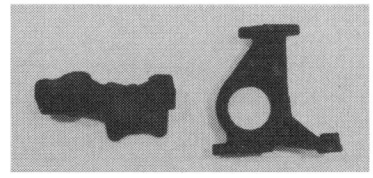


Fig. 5. Reference parts.

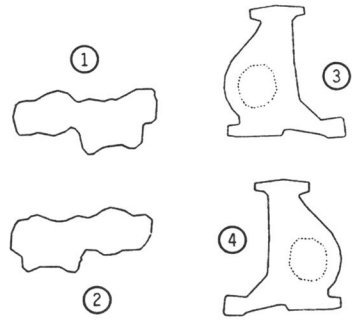


Fig. 6. Models associated with the parts in Figs. 5 and with their symmetric homologues.

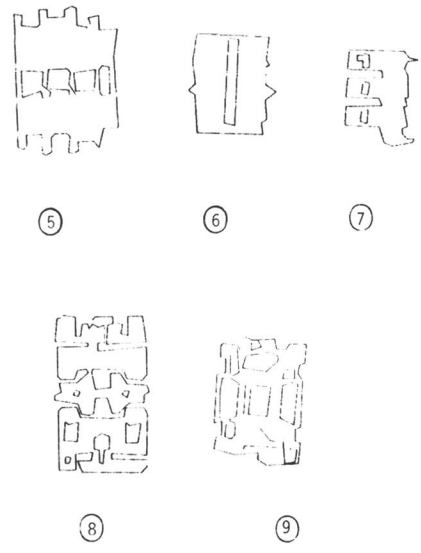


Fig. 7. Models associated with the parts in Fig. 2.

model descriptions associated with these silhouettes. (Symmetric homologues are not shown because their contours are too similar to the original ones).

The number of segments of these descriptions ranges between 22 and 129. The contrast conditions are poor and vary from one part to another, and the unpredictable presence of reflects (some parts are metallic, others are made of plastic) imposed the use of the second extraction method.

In the following, we assume that both the model and the scene descriptions are given by a set of linear segments, respectively, (M_i) and (S_j) of the form: $M_i = (x_i, y_i, l_i, a_i)$ and $S_j = (x_j', y_j', l_j', a_j')$ where x and y are the coordinates of the segment midpoint, l is the segment length, and a is the segment orientation measured relatively to the horizontal axis.

In addition, the model description will include a certain

number of privileged segments: in the current implementation, the privileged segments are the ten longest segments of the model description. (For justification, see Section IV-A.)

III. Matching Models and Scene Descriptions

A. Overview

The problem is to match in a scene one or several models while allowing for distorsion by partial occlusions and by a similarity transformation (the product of a translation, a rotation, and a scaling). The basic idea is, for each possible model, to generate (predict) and evaluate a number of hypotheses.

To generate a hypothesis is to predict the position of the model in the scene: this prediction is made by matching a privileged segment in the model description (M.D.) with a segment in the scene description (S.D.) by comparing local intrinsic features. Typically, a few hundred hypotheses are generated and ranked on the basis of a local criterion of merit.

To evaluate a hypothesis is to take advantage of the predicted position of the model to identify additional segments between the two descriptions, and also refine the predicted position of the model (by a Kalman filter). Only the best first hypotheses are evaluated (typically a few tens), and the result of each evaluation is a final position estimate and a quality measure which accounts for the relative length of the identified segments.

The matching ends when a sufficient number of hypotheses has been evaluated or when a very high quality measure is reached. The hypothesis with the highest quality score is then reexamined before being validated or rejected.

We shall now describe these different stages in detail.

B. Generating Hypotheses

The model position is defined by a transformation T, the product of a rotation, a scaling, and a translation. The transformation T is described by a parameter vector $v = (k \cdot \cos\theta, k \cdot \sin\theta, tx, ty)$, such that the image (x^*, y^*) of an arbitrary point (x, y) of the M.D. is given by the set of equations

$$x^* = tx + x \cdot k \cdot \cos\theta - y \cdot k \cdot \sin\theta \quad (1)$$

$$y^* = ty + x \cdot k \cdot \sin\theta + y \cdot k \cdot \cos\theta. \quad (2)$$

Given an M.D. and S.D., a hypothesis (i.e., a prediction of the position of the model in the scene) is generated by matching a privileged segment of the M.D. to a compatible segment of the S.D. Compatibility is locally defined as follows:

Let M_0 be a privileged segment of the M.D. A segment S_{j0} of the S.D. is defined as compatible with M_0 iff:

1) the angle A between M_0 and its preceding neighbor is close to the angle A' between S_{j0} and its preceding neighbor. (Close means that abs $(A - A')$ is lower than a threshold, typically 30 degrees).

2) the ratio r between the lengths of S_{j0} and M_0 is close to the *a priori* estimate k_0 of the scale factor, when this estimate is available (close means that abs $(r - k_0)$ is below a threshold, typically $0.3 * k_0$).

When a privileged segment M_0 is matched to a compatible scene segment S_{j0}, and if no *a priori* estimate k_0 of the scale factor is available, the parameter vector $v_0 = (k_0 \cos\theta_0, k_0 \sin\theta_0, t_{x0}, t_{y0})^T$ of T_0 is computed by resolving (1) and (2) for the two pairs of corresponding endpoints of M_0 and S_{j0} [see (3)–(6)]. In practice, one sometimes has a good *a priori* estimate k_0 of the scale factor k. In this case, the three remaining parameters θ_0, tx_0, and ty_0 of T_0 are computed by (4)–(6) only.

$$k_0 = l(S_{j0})/l(M_0) \quad (3)$$

$$\theta_0 = a(S_{j0}) - a(M_0) \quad (4)$$

$$tx_0 = x'_0 - k_0 \cdot (x_0 \cdot \cos\theta_0 - y_0 \cdot \sin\theta_0) \quad (5)$$

$$ty_0 = y'_0 - k_0 \cdot (x_0 \cdot \sin\theta_0 + y_0 \cdot \cos\theta_0) \quad (6)$$

where $l(S_{j0})$, $l(M_0)$, $a(S_{j0})$, and $a(M_0)$ denote, respectively, the lengths and orientations relative to the horizontal axis of S_{j0} and M_0, and where (x'_0, y'_0) are the coordinates of the midpoints of the segments S_{j0} and M_0, respectively.

Since the initial estimate T_0 of T is very likely in error, we introduce a measure of this error: S_0 is an error covariance matrix defined by

$$S_0 = E((v_0 - v) \cdot (v_0 - v)^t) \quad (7)$$

where v and v_0 are, respectively, the parameter vectors of the unknown transformation T and its estimate T_0. In practice, S_0 is initialized for each hypothesis with respect to the error variances s_k^2, s_a^2, s_x^2, and s_y^2 attached, respectively, to the initial estimates k_0, θ_0, t_{x0}, and t_{y0}. In the current implementation, these variances are heuristically estimated. Assuming that s_k^2 and s_a^2 are small compared to 1, the elements of S_0 are approximated by

$$S_0(1, 1) = k_0^2 \sin^2(\theta_0) \cdot s_a^2 + \cos^2(\theta_0) \cdot s_k^2 \quad (8)$$

$$S_0(2, 2) = k_0^2 \cos^2(\theta_0) \cdot s_a^2 + \sin^2(\theta_0) \cdot s_k^2 \quad (9)$$

$$S_0(1, 2) = S_0(2, 1) = \sin(\theta_0)\cos(\theta_0) \cdot (s_k^2 - k_0^2 \cdot s_a^2) \quad (10)$$

$$S_0(3, 3) = s_x^2 \quad (11)$$

$$S_0(4, 4) = s_y^2 \quad (12)$$

the other terms of S_{j0} being equal to zero.

When a given number of hypotheses has been generated (typically a few hundred), the hypotheses are ranked by measuring the compatibility between the pairs of matched segments (see above). Then the best hypotheses (usually a few tens) are evaluated.

C. Evaluating Hypotheses

After computing an initial estimate T_0 of the transformation, we match additional segments of the M.D. with segments of the S.D., while updating the estimate of the

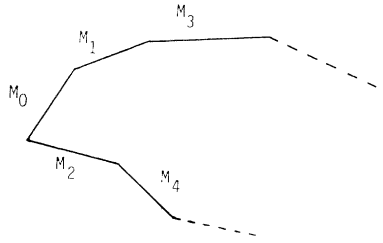


Fig. 8. Order of selection of the model segments.

position of the model in the scene and computing a quality measure of the resulting match. We now proceed to describe in more details those three points.

1) Matching Additional Segments: After having identified M_0 with S_{j0}, the program matches the other segments of the model by an iterative algorithm: at iteration i, the program selects among the non yet examined segments of the M.D., the segment M_i which is closest to M_0 (see Fig. 8). The choice of segments M_i close to M_0 is because if the initial estimate T_0 of T is inaccurate, then the error in position between the estimated image $T_0(M_i)$ of $T(M_i)$ increases with the distance $\|M_0M_i\|$. This segment M_i is transformed into a segment M_i^* by the current estimate T_{i-1} of the transformation T. Then a dissimilarity measure d_{ij} is computed between the image segment M_i^* and every segment S_j of the S.D. This dissimilarity measure is a weighted sum of three positive quantities which, respectively, account for

1) a_{ij} = the absolute value of the difference between orientations of M_i^* and S_j,

2) D_{ij} = the Euclidean distance between the midpoints of M_i^* and S_j,

3) l_{ij} = the absolute value of the relative difference between lengths of M_i^* and S_j: $l_{ij} = (l_i^* - l_j)/l_j$.

Each of these quantities is upper bounded by $a_{\max}$, $D_{\max}$, and $l_{\max}$, respectively. d_{ij} is then computed as follows:

- if a_{ij} or D_{ij} or l_{ij} is above its corresponding upper bound, then $d_{ij} = 1$.
- otherwise, d_{ij} is given by

$$d_{ij} = p \cdot a_{ij}/a_{\max} + q \cdot D_{ij}/D_{\max} + r \cdot l_{ij}/l_{\max} \quad (13)$$

where p, q, and r are associated positive weights which add up to one. In the current implementation, we chose $p = 0.6$, $q = 0.3$, and $r = 0.1$, values emphasizing the role of the segments orientation.

d_{ij} takes a minimum value of zero when M_i^* and S_j are just superimposed, and increases when the discrepancy between M_i^* and S_j increases: the maximum value of d_{ij} is 1, and this value is reached if and only if one of the quantities $a_{ij}/a_{\max}$, $D_{ij}/D_{\max}$, or $l_{ij}/l_{\max}$ is greater than or equal to one. In the current implementation we have $a_{\max} = 20$ degrees, $D_{\max} = 12$ pixels, and $l_{\max} = 70$ percent.

M_i is matched with the segment S_j of the S.D. such that d_{ij} is minimum and lower than one. Otherwise, M_i is matched with NIL, which means that M_i has no homologue in the S.D. with respect to the current hypothesis.

2) Updating the Model Position: When a segment M_i is matched with a segment S_{ji}, a recursive least square technique (Kalman filter) is used to update the estimate T_{i-1} of T. The new value of the parameter vector v_i is computed as follows.

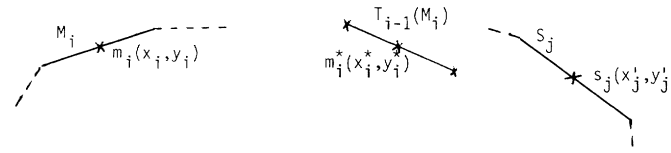


Fig. 9. Notations for matching.

Basic Method: Given a set of matches $\{(M_i, S_{ji})\}$, we look for the transformation T which minimizes the criterion

$$R = \sum_i \frac{l_i}{K} \Delta^2(T(m_i), s_{ji}) \quad (14)$$

where m_i and s_{ji} are the midpoints of segments M_i and S_{ji}, respectively, Δ is the usual Euclidean distance, and l_i is the length of segment M_i. The term l_i^2/K is here to emphasize the role of long segments which are less sensitive to noise. K is a constant whose value depends on the quality of the observed images (in our implementation, we simply computes $K = D_{\max} \cdot l_{\text{mean}}$ where l_{mean} is the average segment length and $D_{\max}$ is the quantity defined in Section III-C-1).

If we represent, as in Section III-B, the transformation T by the vector $v = (k \cos\theta, k \sin\theta, tx, ty)^t$, and the point s_{ji} of coordinates x_i', y_i' by the vector $Y_i = (x_i', y_i')^t$, we can rewrite (14) as

$$R = \sum_i (Y_i - C_i v)^t W_i^{-1} (Y_i - C_i v). \quad (15)$$

Matrix C_i is given by

$$C_i = \begin{pmatrix} x_i & -y_i & 1 & 0 \\ y_i & x_i & 0 & 1 \end{pmatrix}$$

where x_i and y_i are the coordinate of point m_i.

Matrix W_i is given by

$$W_i = \begin{pmatrix} w_i^2 & 0 \\ 0 & w_i^2 \end{pmatrix} \quad \text{with } w_i = K/l_i.$$

We would also like to control the variation of some of the parameters of the transformation T. This can be achieved by adding to R an extra term of the form $(v - v_0)^T S_0^{-1}(v - v_0)$ where v_0 corresponds to the initial hypothesis and S_0 is the matrix described in Section III-B. Finally, the criterion R is written as

$$R = \sum_i (Y_i - C_i v)^t W_i^{-1} (Y_i - C_i v)^t + (v - v_0)^t S_0^{-1} (v - v_0). \quad (16)$$

R is a quadratic criterion and can be minimized recursively by the standard following equations (cf. [24] for instance):

$$v_i = v_{i-1} + K_i \cdot [Y_i - C_i \cdot v_{i-1}] \quad (17)$$

$$K_i = S_{i-1} \cdot C_i^T \cdot [W_i + C_i \cdot S_{i-1} \cdot C_i^T]^{-1} \quad (18)$$

$$S_i = [I - K_i \cdot C_i] \cdot S_{i-1}. \quad (19)$$

These equations are initialized for a new hypothesis by S_0 and v_0 computed in Section III-B, and are recursively updated after each new match (M_i, S_{j_i}).

Refined Method: In the previous approach, the updating of the transformation T was done by trying to superimpose the centers of the matched segments. More accurate results were obtained by superimposing the center of each identified model segment on the straight line supporting its homologous scene segment. In this case, the determination of T is much less sensitive to the variations of the segment lengths, as it does not modify the position of the supporting straight lines.

In that case we simply minimize the criterion

$$R' = \sum_i \frac{l_i}{K} \Delta^2(T(m_i), S_{j_i}) \quad (14')$$

where $\Delta(T(M_i), S_{j_i})$ is the distance of the point $T(m_i)$ to the infinite line containing S_{j_i}. If its orientation is a_i' and if the coordinates of s_{j_i} are (x_i', y_i'), the criterion can be rewritten as

$$R' = \sum_i \frac{l_i}{K} ([-\sin(a_i') \cos(a_i')]\, C_i v + \delta_i')^2 \quad (15')$$

where $\delta_i' = x_i' \sin(a_i') - y_i' \cos(a_i')$. The minimization is performed exactly in the same way as before.

3) Computing a Quality Measure: The use of the quality measure is to discriminate between correct and wrong hypotheses. After each iteration i, $Q(i)$ measures the length of the identified model segments as a percentage of the total model length. $Q = Q(N)$ (N is the number of model segments) is upper bounded by 1; this maximum value is obtained whenever the model is perfectly and entirely identified in the scene. Q decreases in the presence of occlusions and nonrigid distorsions (noise, tilted objects, errors of segmentation, $\cdots$).

D. Ending the Matching Process

The matching ends when the number of hypotheses which have been evaluated is large enough (typically a few tens), or when a very high quality measure is reached by an hypothesis. In each case the hypothesis with the highest quality measure is reexamined before being validated or rejected: the reexamination consists in evaluating a last hypothesis, whose *a priori* parameters are the *a posteriori* estimate and covariance matrix of the best hypothesis. This reexamination is to check whether some additional model segments could be matched with a more accurate initial estimate of T. When this is the case, the process is repeated until it converges. The reexamined hypothesis is then definitely validated if its quality measure is above a prespecified threshold, and rejected otherwise.

IV. Complexity Analysis

A. Computing Time

The average computing time required to match a model description with a scene description is equal to the number of generated hypotheses multiplied by the average evaluation time of a hypothesis.

The number of generated hypotheses is reduced by having a small number of discriminant model segments selected to be used as privileged segments M_0. The choice of the long segments is for two reasons. First, long segments are usually less numerous and therefore more discriminant. Second, the initial estimate of the transformation T is more accurate with long segments. Of course at least one of the privileged segments has to be visible (e.g., occluded length <30 percent of segment length) in the scene for the model to be identified. It appeared that the choice of the 10 longest model segments as privileged segments never prevented the recognition of reasonably occulted objects (e.g., total occluded length <60 percent of model length) in our experiments. This is probably due to the fact that, in this case, the probability of having all the priviliged segments occluded *simultaneously* is very small.

In addition, each privileged segment M_0 of the M.D. is identified only with compatible segment S_{j0} of the S.D. (compatibility is defined in Section V-B). Typically, the number of scene segments compatible with a privileged model segment is about 10 percent of the total number of scene segments (allowing a scale variation of about 30 percent). Therefore, if there are 10 privileged model segments, the number of generated hypotheses is usually close to the number of scene segments.

The evaluation time is reduced mainly by three techniques. First, a branch-and-bound technique is used: during the evaluation of a hypothesis and at each iteration i, the program computes an upper bound $Q_{\max}$ on the final quality measure $Q(N)$: this upper bound is computed by adding to the current partial quality measure $Q(i)$ the normalized length of the model contours which have not been examined yet (therefore assuming this remaining part will be perfectly matched). As $Q(i)$ is a decreasing function of i, the evaluation of the current hypothesis is aborted early (and the hypothesis rejected) as soon as $Q_{\max}$ happens to be lower than the quality measure attached to a previously evaluated hypothesis.

Second, the evaluation process is significantly accelerated by having the scene segment orientations a_j initially sorted: in this case, when searching for the best match S_j of an image segment M_i^* (cf. Section III-C-1), the scene segments S_j whose orientation a_j is compatible with the orientation of M_i^* (i.e., such that abs $(a_i^* - a_j < a_{\max})$ are selected by a binary search in logarithmic time. One could also compute square buckets on the S.D. to have fast access to the scene segments close to a predicted location.

Third, all segments whose length is below a fixed limit (typically 8 pixels for images of size 256 * 256) are removed from both the M.D. and the S.D. before processing.

To conclude, one could notice the possibility of generating and evaluating all hypotheses independently of each other. This property allows for an execution of the program on parallel hardware to still reduce the global computing time. If several models must be located in the same scene, they can also be processed in parallel.

B. Storage Requirements

The storage requirements are small and are a linear function of the data size: one has to store essentially the M.D. and the S.D., i.e., the vertices coordinates of two polygonal approximations (usually a few hundred points). Also, and in order to speed up the evaluation process (cf. above), one can store the orientations of the segments of both descriptions.

V. Results

The recognition method described in this article has been integrated within a vision system and tested on a large number of different scenes. The vision system has also been coupled to an industrial robot arm to achieve picking and repositioning of unoriented partially overlapping industrial parts [25]. We present here some typical results which illustrate the capacities of the vision system.

Except for Example 6, programs are written in Fortran and run on a minicomputer Perkin Elmer 3240. Also, computing times refer to the matching process only excluding the image segmentation process; in effect, the segmentation process is totally independent of the matching process and should be performed in a fraction of a second on dedicated hardware.

A. Example 1: Illustrative Example

We first present a simple didactic example to illustrate the major steps of the recognition procedure. Fig. 10 illustrates the generation and evaluation of a correct hypothesis, while Fig. 11 illustrates the discrimination between correct and wrong hypotheses.

Let us consider the two left-most drawings of the first row of Fig. 10; they show, respectively, the model description associated to a car shock absorber and the scene description associated to the image of a similar part rotated, translated, and partially occluded by another part. Both descriptions have been obtained by the method described in Section I (good lighting conditions).

The task of the recognition program is to match and locate the model within the scene. Among all the hypotheses generated by the program, a correct hypothesis is obtained when the privileged model segment represented by a solid line in the M.D. is matched to the compatible scene segment represented by a solid line in the S.D. (first row of Fig. 10). In this situation, the program determines an a priori estimate of the model position which is shown in the third column of this first row.

Rows 2, 3, and 4 of Fig. 10 correspond to some steps of the evaluation process. Columns 1 and 2 show, respectively, in solid lines the segments which are identified between the M.D. and the S.D. Column 3 shows the current estimate of the model position. Successive estimates are

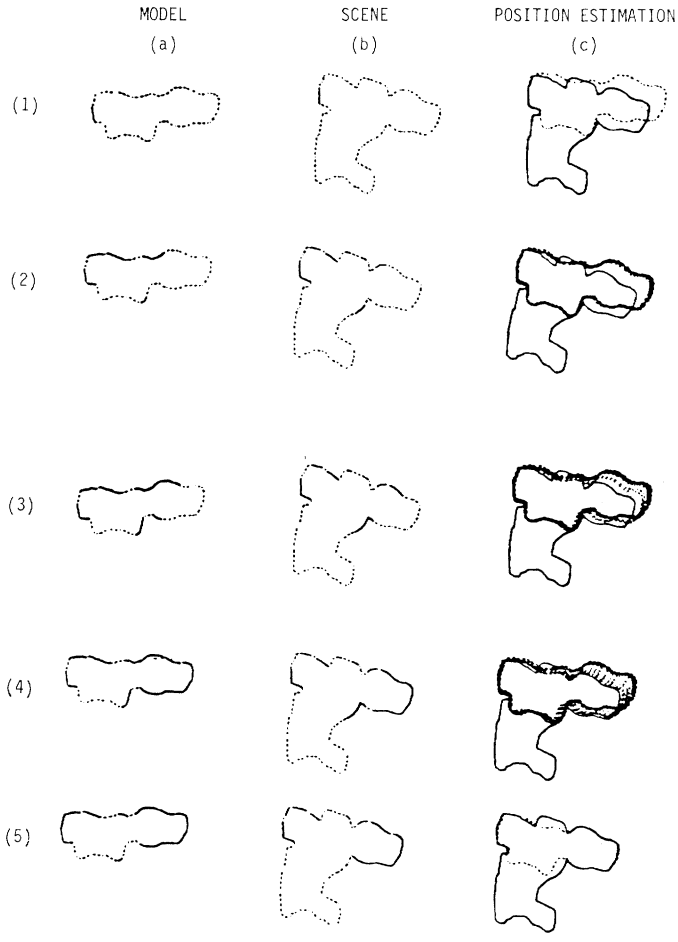


Fig. 10. Generation and evaluation of a correct hypothesis (see text).

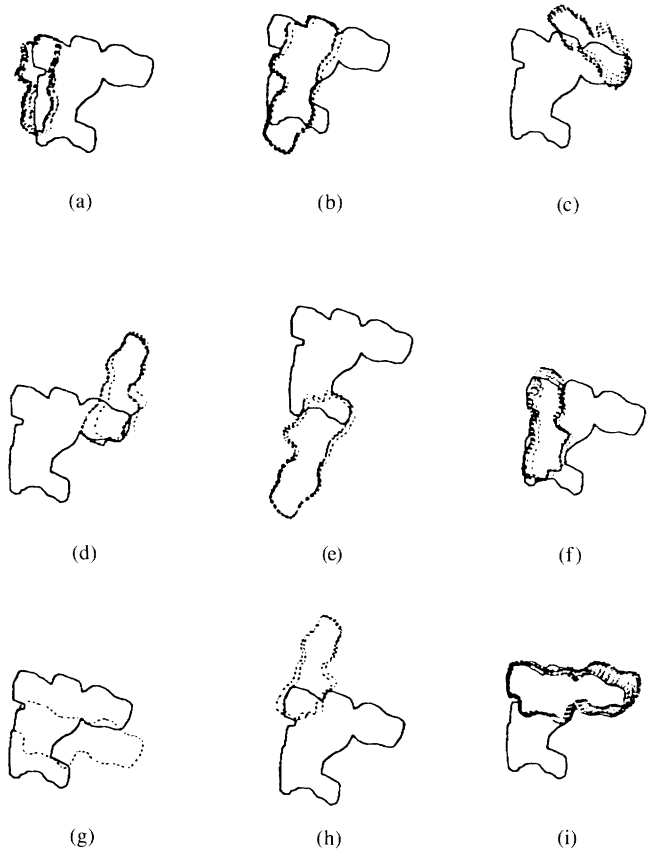


Fig. 11. Discrimination between wrong and correct hypotheses (see text).

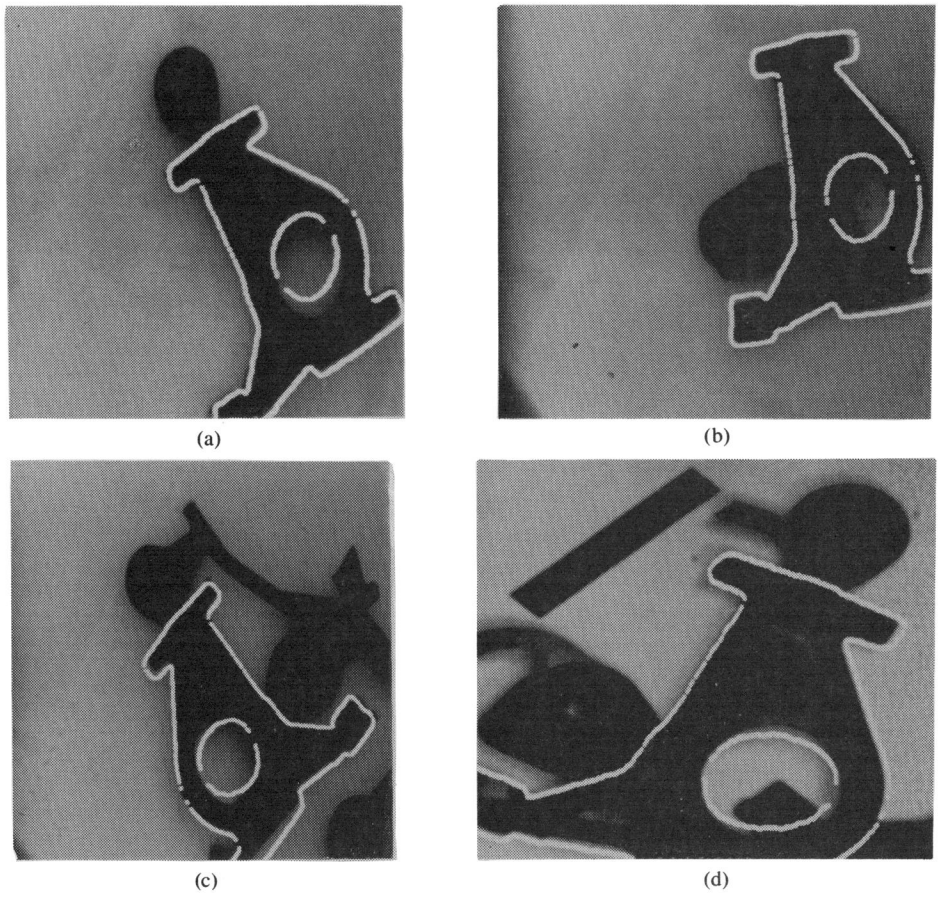


Fig. 12. Models 3 and 4 of Fig. 6 are identified and located (in white) in four scenes. Notice the presence of sprues and dead-heads on the castings, and the presence of scaling between model and scene descriptions of 0.68 in scenes (a), (b), and (c), and 1.06 in scene (d).

superimposed to better exhibit the convergence result, and one can visualize the recursive update of the model position. Numerically speaking, the parameters (θ, k, tx, ty) of the transformation T vary from an *a priori* estimate ($-81.7°$, 1.13, -49 pixels, 281 pixels) to a final estimate ($-73.86°$, 1.015, -15.67 pixels, 237.52 pixels).

The last row of Fig. 10 shows the final result obtained after the reexamination of this hypothesis: one can notice the correction of some matching errors which were initially due to the inacurrate *a priori* estimate of the model position. The parameters of the final estimate of T after reexamination are ($-74.07°$, 1.00, -9.55 pixels, 236.34 pixels).

Among all the hypotheses generated by the program, we see in Fig. 11 nine hypotheses generated when the same privileged segment is identified with nine different compatible scene segments. All these hypotheses (except for the last one) have a quality measure lower than 0.25 and are rejected; the last hypothesis has a quality measure greater than 0.60, and is validated.

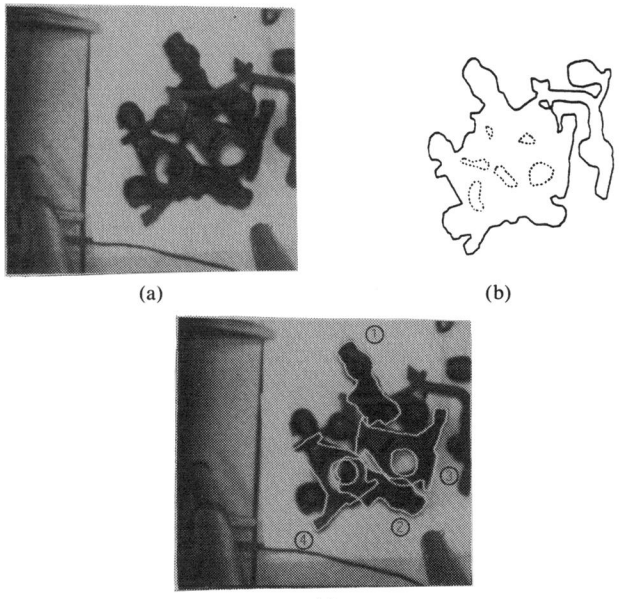
B. Example 2: Castings with Dead-Heads, Variable Scale Factor

Fig. 12 shows the result of the identification and positioning of models 3 and 4 of Fig. 6 in four different scenes containing similar parts in arbitrary planar positions. The parts are observed just after the casting process and they have sprues and dead-heads attached to them. In addition, there is a scaling between models and scenes of 0.68 in Fig. 12(a), (b), and (c), and of 1.06 between models and (d). This scaling is taken into account by using (3) to estimate k_0.

The pictures are segmented as described in Section II (good lighting conditions). The models are correctly detected and located, and the computed positions are superimposed in the pictures. The quality scores vary between 65 and 85 percent, and the computing time is of the order of one second per model.

C. Example 3: Partially Overlapping Castings

Fig. 13 shows the identification and positioning of models 1, 2, 3, and 4 of Fig. 6 in a scene containing similar parts in arbitrary positions [Fig. 13(a)]. In addition, some of the parts have large sprues and dead-heads attached to them, and the parts are partially occluding each other. The picture is segmented as described in Section I (good lighting conditions) and the resulting scene description is shown in Fig. 13(b) (280 segments). The result of the matching is shown in Fig. 13(c), where the models have been superimposed in white on the scene at the location determined by the program. We show in Table I the main parameters of the solution.



(a) (b) (c)

Fig. 13. (a) Original scene containing four partially overlapping castings with sprues and dead-heads. (b) Segmented scene description (280 linear segments). (c) Models of Fig. 6 are identified and located (in white) by the program in the scene.

TABLE I
RECOGNITION OF MODELS 1, 2, 3, AND 4

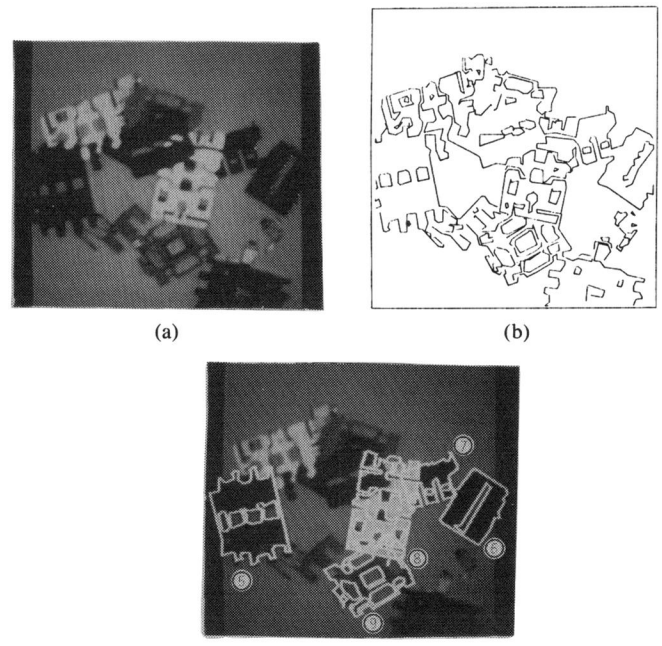
Model	Number of Segments	Quality Measure	Computing Time
1	39	54 percent	0.7 s
2	41	55 percent	1.25 s
3	50	45 percent	1.25 s
4	50	40 percent	2 s

D. Example 4: Partially Overlapping Electromechanical Parts

Fig. 14(a) shows a scene with several overlapping parts of an electromechanical device observed under bad lighting conditions (this image corresponds to the scene shown in Fig. 3). The scene is segmented by the second method described in Section I and the resulting scene description (759 segments) is shown in Fig. 14(b). Models 5, 6, 7, 8, and 9 of Fig. 7 are successfully identified and located in the scene; the result of the matching is shown in Fig. 14(c) where the models have been superimposed in white on the scene at the location determined by the program. Note that when there are several occurences of a model in a scene, the program simply selects the hypothesis with the highest quality measure, which usually corresponds to the most visible occurence. A minor modification in the progam would allow for the recognition of all the occurences of a model corresponding to hypotheses whose quality measure is above a determined threshold. Table II shows the main parameters of the result.

E. Example 5: Coupling with a Robot Arm

The vision system has been coupled to a robot arm to achieve automatic picking and placing of overlapping workpieces. In this system, the modeling of objects includes a list of potential grasping locations, and the recognition procedure includes, when a model is identified, the selection of an accessible grasping location among the potential grasping locations.



(a) (b) (c)

Fig. 14. (a) Original scene containing parts of an electromechanical device (this is the observed image of the scene in Fig. 4). (b) Segmented scene description (759 linear segments). (c) Models of Fig. 7 are identified and located (in white) by the program in the scene.

TABLE II
RECOGNITION OF MODELS 5, 6, 7, 8, AND 9

Model	Number of Segments	Quality Measure	Computing Time
5	73	39 percent	7 s
6	22	36 percent	4 s
7	48	40 percent	2 s
8	129	66 percent	1 s
9	89	38 percent	3.5 s

This robot system is described within details in another article [25] and we shall only present results in the sequel. Fig. 15 shows the potential grasping locations attached to models 8 and 9 of Fig. 7. Fig. 16 shows the accessible grasping locations selected after the recognition of these models, and Fig. 17 shows the actual picking and repositioning of the corresponding objects.

The result of this coupling has been to provide a more realistic testbed for the vision system, and also to demonstrate the feasability of the automatic picking and repositioning of partially overlapping workpieces lying on a flat surface using our vision system.

F. Example 6: Precision Test

It is difficult to compute the accuracy of the determination of the transformation T in general because it depends on many factors such as the nature of the model, the quality of the viewing conditions, and the degree of

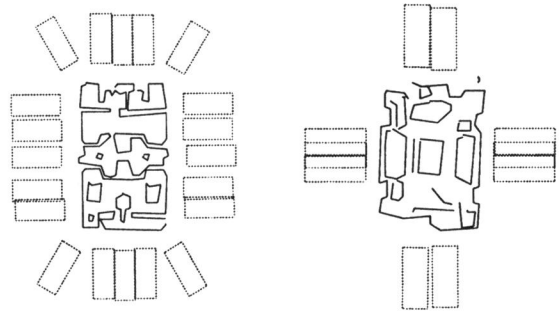


Fig. 15. Grasping locations attached to models 8 and 9 of Fig. 7. The symmetric rectangles are the vertical projections of the gripper fingers.

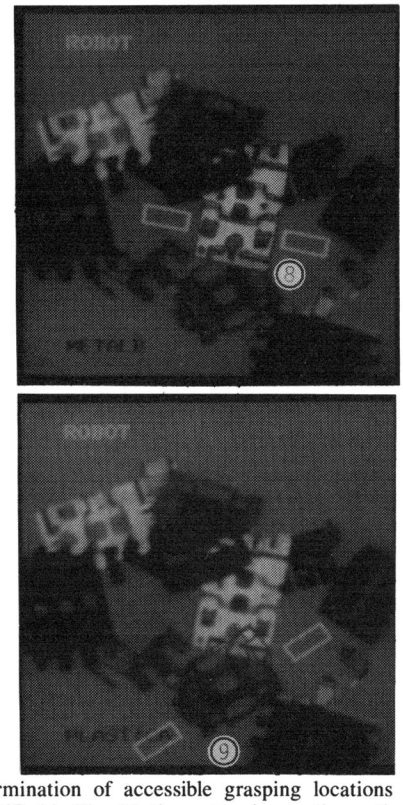


Fig. 16. Determination of accessible grasping locations for two of the models identified in Fig. 14; the rectangles are the vertical projections of the gripper fingers.

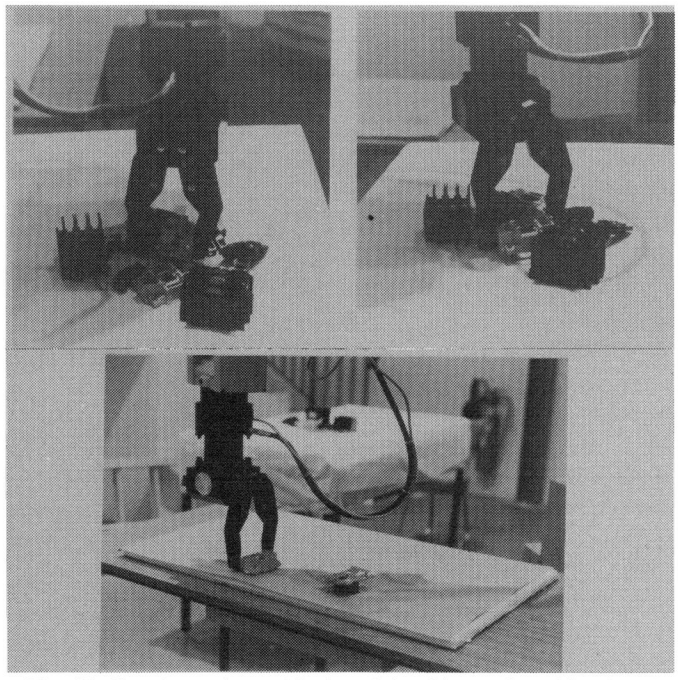


Fig. 17. Grasping and repositioning of the objects selected in Fig. 16.

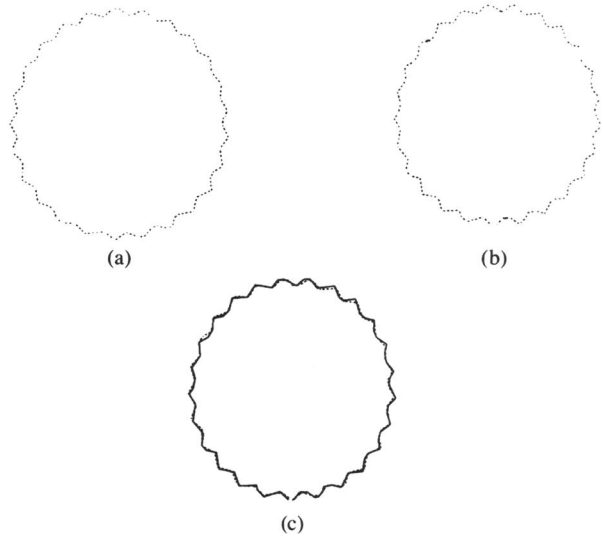


Fig. 18. Descriptions associated with the gears used in the precision test. (a) Reference gear. (b) Gear rotated by an angle of 7°. (c) Superimposition of (a) and (b) by the program.

occultation of the observed objects. At the least, a qualitative estimate was derived by having the robot arm safely picking and repositioning several different objects in many different situations.

However, a quantitative estimate of the accuracy of the determined rotation angle θ was made on images of mechanical gears. The polygonal segmentations extracted from two different images of a gear are shown in Fig. 18. One can notice some local alterations of the contours which are mainly due to unpredictable metallic reflections.

The precision experiment consisted in extracting the description of a gear in a reference position. This description was taken as a model description. Then, the same gear was rotated by a precisely measured angle, and the corresponding extracted description was taken as a scene description. For 300 successive measures, the maximum deviation between the estimate and the actual angle was 0.15 degrees. The measured standard deviation was 0.07 degrees.

For this industrial application, the program was transported to a Motorola-68000 based microcomputer, and partially translated to machine code. The number of generated hypotheses was 50, and only the 8 best hypotheses were evaluated. The maximum length of non identified segments was upper-bounded by 30 percent of the total model length, while the average length of nonidentified segments was 20 percent (due to the lighting conditions and the lack of stability of the segmentation algorithm). The computing time is lower than one second, including the segmentation process which is done on dedicated hardware.

VI. Conclusion

We have described a new method for the recognition and positioning of 2-D objects. This method uses segmented descriptions of the object contours to generate and recursively evaluate a number of selected hypotheses. The main features of this method are its robustness to lighting conditions, partial occlusions (up to 60 percent typically) and scale variations (20–40 percent typically), its accuracy in locating objects, its high degree of parallelism (hypotheses can be generated in parallel), and its small storage requirements (essentially the storage of the segments endpoints). The method has been experimented on a large number of different scenes, and some typical examples have been presented, including the coupling of the vision system to a robot arm.

Returning to problems I, II, and III of the Introduction, we can say that the existing work in computer vision allows us to generate quickly and accurately the outlines of flat objects even under difficult viewing conditions. Boundary representations can then easily be built by functional approximation techniques. First degree polynomials were used in this paper but nothing (except the computing time) would have prevented us from using other functions. Problem I is therefore solved by algorithms which can be implemented by very fast programs or hardware.

We have only scratched the surface of problem II. We represent our models the same way as our scenes, i.e., with polygonal approximations. The corresponding database is simply a sequence of such models and no ways are provided for smarter model indexing than a simple linear scan of that database. A lot remains to be done in this area.

Problem III, that of matching models with scene descriptions, has been solved in a simple way by exploiting the very important constraint of rigidity. This allows us to work with the well-known group of similarity transformations and to drastically prune our search tree. By coupling the search algorithm with a recursive estimation of the transformation, we have achieved a high positional accuracy. We believe that these two features (exploiting rigidity and recursive parameters estimation) are basic in many related applications. In that sense we think our work is also a contribution to establishing the lacking methodology we were referring to in the Introduction.

References

[1] N. J. Zimmerman, G. J. R. Van Boven, and A. Oosterlinck, "Overview of industrial vision systems," in *Industrial Applications of Image Analysis*. Pijnacker, The Netherlands: D. E. B., 1983.

[2] W. A. Perkins, "A model based vision system for industrial parts," *IEEE Trans. Comput.*, vol. C-27, pp. 126–143, Feb. 1978.

[3] J. D. Dessimoz *et al.*, "Recognition and handling of overlapping industrial parts," in *Proc. 9th Symp. Industrial Robots*, Washington, DC, 1979.

[4] P. Rummel and W. Beutel, "A model based image analysis system for workpiece recognition," in *Proc. 6th Int. Conf. Pattern Recognition*, Munich, Oct. 1982, pp. 1014–1017.

[5] W. Hattich *et al.*, "Experience with two hybrid systems for the recognition of overlapping workpieces," in *Proc. 6th Int. Conf. Pattern Recognition*, Munich, Oct. 1982, pp. 670–673.

[6] A. P. Ambler *et al.*, " A versatile computer controlled assembly system," in *Proc. IJCAI*, Stanford CA, 1973, pp. 298–307.

[7] R. C. Bolles and R. A. Cain, "Recognizing and locating partially visible workpieces," in *Proc. IEEE Comput. Soc. Conf. Pattern Recognition and Image Processing*, Las Vegas, NV, June 1982, pp. 498–503.

[8] L. Davis, "Shape matching using relaxation techniques," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, pp. 60–72, Jan. 1979.

[9] B. Bhanu and O. D. Faugeras, "Shape matching of two-dimensional objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, Mar. 1984.

[10] N. J. Ayache and O. D. Faugeras, "Recognition of partially visible planar shapes," in *Proc. 6th Int. Conf. Pattern Recognition*, Munich, Sept. 1982.

[11] A. Lux and V. Souvignier, "PVV - Un système de vision appliquant une stratégie de Prédiction-Vérification" (in French), in *4eme Congrès Rec. Formes et Intell. Artificielle*, Paris, 1984, pp. 223–234.

[12] J. Segen, "Unsupervised feature selection for object recognition," in *Proc. SPIE Int. Soc. Opt. Eng.*, vol 360, 1982, pp. 132–135.

[13] J. L. Turney *et al.*, "Recognizing partially hidden objects," in *Proc. SPIE Int. Soc. Opt. Eng.* vol. 521, 1984, pp. 108–113.

[14] W. E. L. Grimson and T. Lozano-Pérez, "Recognition and localization of overlapping parts from sparse data in two and three dimensions," presented at the IEEE Comput. Soc. Int. Conf. Robotics, St. Louis, MO, Mar. 1985; see also M.I.T. AI Lab. Memo. 841.

[15] N. Ayache, "Un systeme de vision bidimensionnelle en robotique industrielle" (in French), INRIA Tech. Rep. ISBN 2-7261-0345-6, 1983.

[16] N. Ayache and C. Darmon, "Reconnaissance récursive et localisation de formes planes partiellement visibles dans une image" (in French), in *Proc. 9ème Colloque le traitement du Signal et Ses Applications*, Nice, France, May 1983, pp. 611–617.

[17] N. Ayache, "A model based vision system to identify and locate partially visible planar shapes," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, Washington, DC, June 1983, pp. 492–494.

[18] O. D. Faugeras and M. Hebert, "A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces," in *Proc. 8th Int. Conf. Artificial Intell.*, Karlsruhe, West Germany, Aug. 1983, pp. 996–1002.

[19] J. Serra, *Image Analysis and Mathematical Morphology*. London: Academic, 1982.

[20] D. Marr, *Vision*. San Francisco, CA: Freeman, 1982.

[21] N. Keskes *et al.*, "Application of image analysis techniques to seismic data," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Paris, 1982, pp. 855–857.

[22] J. P. Chieze and P. Germain, "Logiciel d'analyse de contours" (in French), INRIA Tech. Rep., 1979.

[23] T. Pavlidis, *Structural Pattern Recognition*. New York: Springer-Verlag 1977.

[24] B. D. Anderson and J. D. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[25] N. Ayache, J. D. Boissonnat, B. Bollack, and B. Faverjon, "Automatic handling of overlapping workpieces," precented at ICPR'84, Montreal, P. Q., Canada.

Nicholas Ayache was born in Paris, France, on November 1, 1958. He graduated from the Ecole Nationale supérieure des Mines de Saint Etienne in 1980, received the M.S. degree from the University of California, Los Angeles, in 1981, and the Docteur-Ingénieur degree in computer science from the University of Paris XI, in 1983.

Since 1981, he has been a researcher in the Computer Vision and Robotics Group of INRIA (Institut National de Recherche en Informatique et Automatique) in Le Chesnay, France. His current research interests are in computer vision and related fields.

Olivier D. Faugeras (S'76–M'76) is the Scientific Director of the Computer Vision and Robotics group at INRIA (Institut National de Recherche en Informatique et Automatique), Le Chesnay, France, and a Senior Lecturer at the University of Paris XI and Ecole Polytechnique. His interests are in computer vision, graphics, robotics, pattern recognition, and artificial intelligence.

Mr. Faugeras is a member of the Association for Computing Machinery.