Genetic Object Recognition Using Combinations of Views

George Bebis, *Member, IEEE*, Sushil Louis, Yaakov Varol, and Angelo Yfantis

***Abstract*—We investigate the application of genetic algorithms (GAs) for recognizing real two-dimensional (2-D) or three-dimensional (3-D) objects from 2-D intensity images, assuming that the viewpoint is arbitrary. Our approach is model-based (i.e., we assume a predefined set of models), while our recognition strategy lies on the recently proposed theory of algebraic functions of views. According to this theory, the variety of 2-D views depicting an object can be expressed as a combination of a small number of 2-D views of the object. This implies a simple and powerful strategy for object recognition: novel 2-D views of an object (2-D or 3-D) can be recognized by simply matching them to combinations of known 2-D views of the object. In other words, objects in a scene are recognized by "predicting" their appearance through the combination of known views of the objects. This is an important idea, which is also supported by psychophysical findings indicating that the human visual system works in a similar way. The main difficulty in implementing this idea is determining the parameters of the combination of views. This problem can be solved either in the space of feature matches among the views ("image space") or the space of parameters ("transformation space"). In general, both of these spaces are very large, making the search very time consuming. In this paper, we propose using GAs to search these spaces efficiently. To improve the efficiency of genetic search in the transformation space, we use singular value decomposition and interval arithmetic to restrict genetic search in the most feasible regions of the transformation space. The effectiveness of the GA approaches is shown on a set of increasingly complex real scenes where exact and near-exact matches are found reliably and quickly.**

***Index Terms*—Algebraic functions of views, genetic algorithms, object recognition.**

I. Introduction

THE PROBLEM of object recognition is fundamental in computer vision. Although a variety of approaches have been proposed to tackle the recognition problem during the last two decades [1], [2], building computer vision systems capable of recognizing relevant objects in their environment with accuracy and robustness has been a difficult and challenging task. Object recognition is difficult because the appearance of an object can have a large range of variation due to photometric effects, scene clutter, changes in shape (e.g, onrigid objects), and, most importantly, viewpoint changes. As a result, different views of the same object can give rise to widely different images. Accommodating variations due to viewpoint changes is a central problem in the design of any object recognition system.

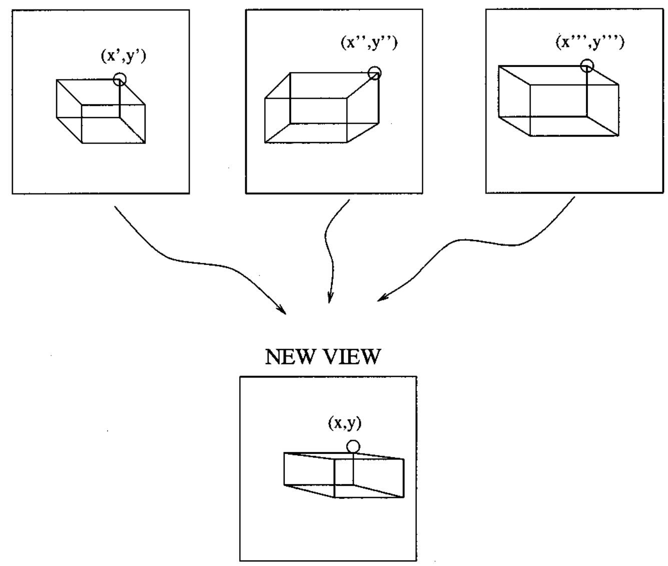


Fig. 1. New views can be obtained by combining three reference views.

Typical strategies for coping with the variable appearance of objects due to viewpoint changes include the use of invariants [3], explicit models [4], and multiple views [5]. According to the first strategy, invariant properties (i.e., properties that remain unchanged as viewing conditions change) are employed during recognition. The problem with this approach is that there areno general case invariants for three-dimensional (3-D) objects [6]. The second strategy employs explicit 3-D models. During recognition, a model of the image formation process is applied to the 3-D model objects in order to predict the objects' appearance and determine whether something of similar appearance can be found in the image. Methods based on this approach are not very practical since 3-D models are not always available. The last strategy models an object by a set of views showing how the object appears from various viewpoints. Systems based on this approach recognize the object in an image when they are able to match one of the reference views to some part of the image. The problem with this strategy is that it requires the storage of many views for each model.

The theory of algebraic functions of views (AFoVs) [7]–[13] provides a powerful foundation for tackling variations in the

Manuscript received July 7, 2000; revised June 6, 2001. This work was supported in part by the Office of Naval Research under Grant N00014-01-0781 and in part by the National Science Foundation under Grant 9624130. This paper was presented in part at the 11th International Conference on Computer Applications in Industry and Engineering, Las Vegas, NV, November 1998, in part at the International Conference on Imaging Science, Systems, and Technology, Las Vegas, NV, July 1998, and in part at the Seventh Annual Conference on Evolutionary Programming, San Diego, CA, March 1998.

G. Bebis, S. Louis, and Y. Varol are with the Department of Computer Science, University of Nevada, Reno, NV 89557 USA (e-mail: bebis@cs.unr.edu; sushil@cs.unr.edu; varol@cs.unr.edu).

A. Yfantis is with the Department of Computer Science, University of Nevada, Las Vegas, NV 89154 USA (e-mail: Yfantis@cs.unlv.edu)

Publisher Item Identifier S 1089-778X(02)04104-8.

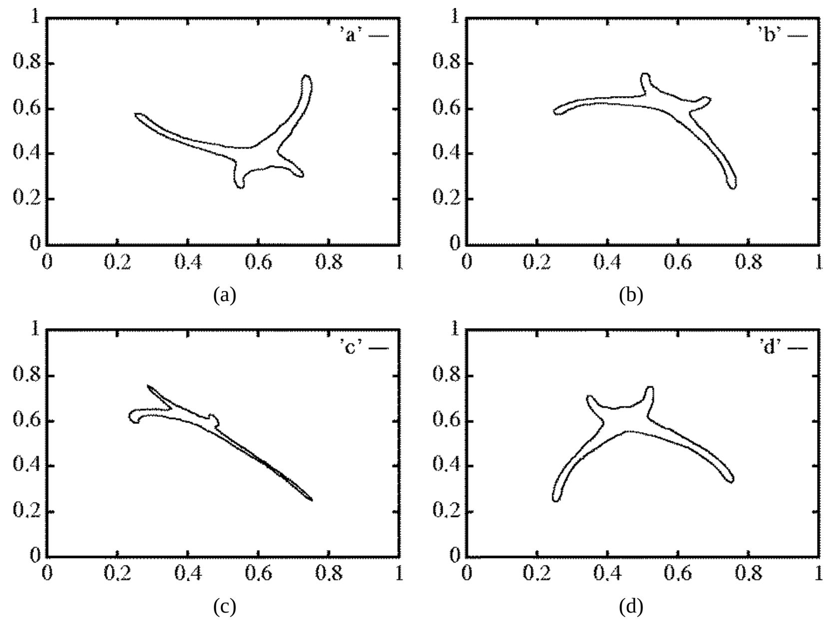


Fig. 2. (a) Known view of a planar object (b)-(d) new views of the same object obtained using (5) and (6). The x axis corresponds to the x coordinates of the contour and the y axis to the y coordinates.

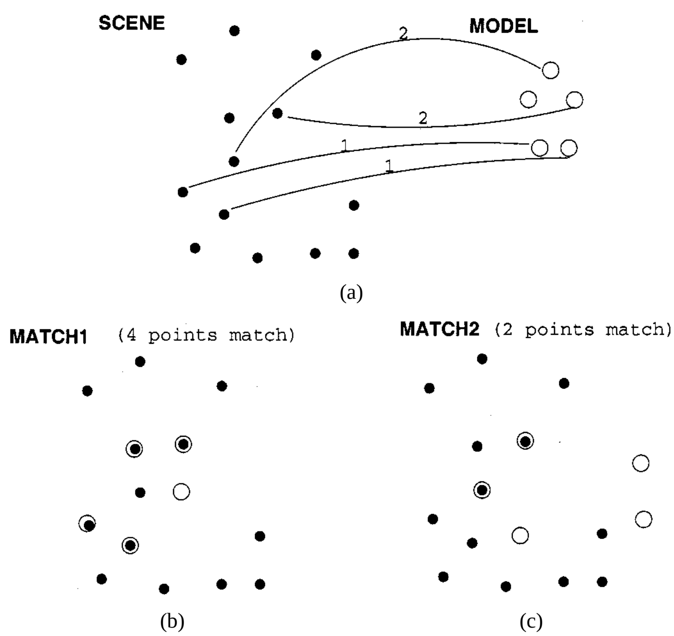


Fig. 3. (a) Two different hypothetical matches of two-point correspondences between the model and the scene. (b) *Match1* produces more matches (good match), while (c) *Match2* fails to produce more matches (bad match).

appearance of an object's shape due to viewpoint changes. According to this theory, the variety of two-dimensional (2-D) views depicting an object (2- or 3-D) can be expressed as a combination of a small number of 2-D views of the object [7], [8], in the case of 2-D objects, for viewpoint. In this case, the combination scheme is equivalent to an affine transformation of the image coordinates from the known view. Various results exist for 3-D objects. Under the assumption of orthographic projection and 3-D rigid transformations, e.g., three views are enough to represent any view from the same aspect [7]. These results suggest a simple but powerful framework for object recognition: novel views of an object (2-D or 3-D) can be recognized by simply matching them to combinations of a small number of stored views (reference views) of the object. This result is also supported by psychophysical findings indicating that the human visual system works in a similar way [14], [15].

TABLE I
COMPUTED VALUE INTERVALS FOR THE 2-D OBJECT

	Ranges of values		
	a1 range	a2 range	a3 range
original	[-2.953, 2.953]	[-2.89, 2.89]	[-1.662, 2.662]
preconditioned	[-0.408, 0.408]	[-0.391, 0.391]	[0.0, 1.0]

Employing AFoVs for recognition has several advantages. First, it is more practical than methods requiring explicit 3-D models. In fact, a sparse set of 2-D views is required to represent a 3-D object, however, the scheme is as powerful as using 3-D models. Second, it is more efficient since it stores and manipulates 2-D views only. In contrast to multiview approaches, however, novel views are compared to "predicted" views (i.e., combinations of reference views) rather than to the reference views themselves. Since the predicted views can be very different from the reference views, recognition does not depend on the similarity between novel and reference views, as it is the case with multiview approaches. Finally, it is more general since the above theoretical results hold true for various projection models and transformations (see Section II), giving rise to a "family" of methods.

Given a novel view of an object, recognition based on AFoVs implies "predicting" the novel view by combining together known views of the object. The main difficulty in implementing this idea is choosing the parameters of the combination scheme. As discussed in [7], the parameters of the combination can be recovered either by: 1) identifying a set of features from the novel view that approximately match a set of features from the known views [image space (IS)] or 2) searching the space of parameters [transformation space (TS)] explicitly. In

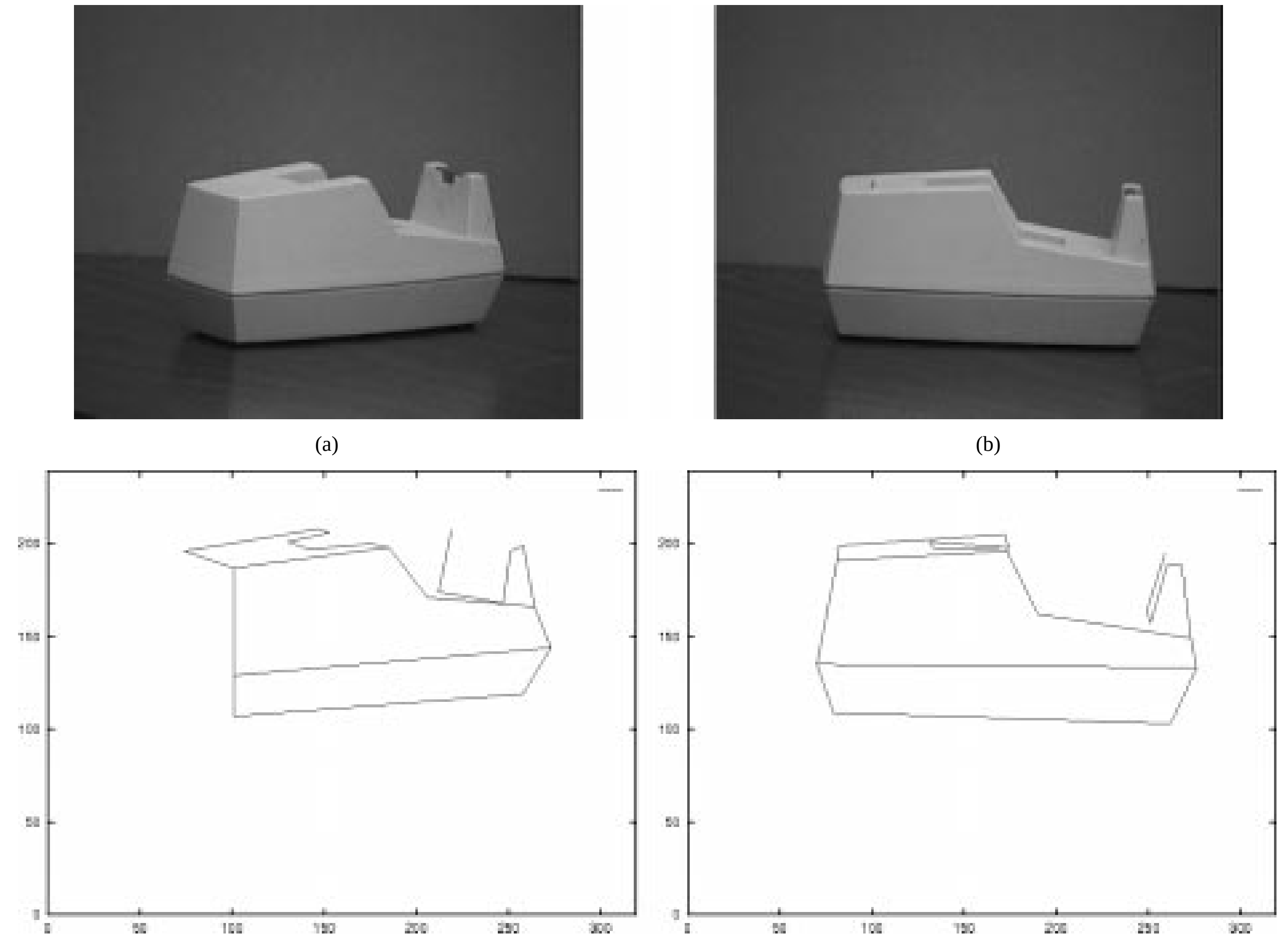


Fig. 4. One of the 3-D objects and the reference views used in our experiments. (a) First reference view. (b) Second reference view. (c) Interest points from the first reference view (lines connecting the corners have been added to enable visualization purposes only). (d) Interest points from the second reference view (lines connecting the corners have been added to enable visualization purposes only). The x axis corresponds to the x coordinates of the contour and the y axis to the y coordinates.

the IS, one has to compute the transformation that aligns the model with the scene by solving a system of linear equations. The main problem with this approach is that the number of model-scene feature matches grows exponentially as the number of scene features increases. Searching the TS avoids the feature matching step. However, this can be very time consuming due to the large number of possible transformations.

In this paper, we propose using genetic algorithms (GAs) for searching these spaces efficiently [16]–[18]. Our goal is to find if an instance of a given model appears in a scene or not. Two different approaches are considered: genetic algorithm in the image space (GA-IS) and genetic algorithm in the transformation space (GA-TS). GAs are search procedures that have been shown to perform well when the space to be searched is very large [16], [17].To facilitate GA-TS, we employ singular value decomposition (SVD) [19] and interval arithmetic (IA) [20], [21] to estimate the interval of values that the parameters of the combination scheme can assume [22]–[24]. Thus, genetic search is confined to only the most feasible regions of the TS.

GAs have been used to solve various problems in computer vision. For example, they have been used for feature selection [25], target recognition [26], and face detection/verification [27]–[29]. There have also been several attempts to apply GAs in object recognition [30], [31]. In [30], GAs were used to solve the object recognition problem, formulated as a graph matching problem. The results reported in [30] were encouraging, but were based on artificial objects only. Extending this approach to 3-D object recognition is not easy since it requires that the 3-D representation of the objects is available. Our approach, on the other hand, relies on 2-D representations only. In [31], point patterns were matched using GA-IS, assuming similarity transformations (i.e., camera perpendicular to the image plane). That work has similarities with our GA-IS approach, however, it does not generalize efficiently to 3-D object recognition (i.e., it would require either 3-D models or a very large number of 2-D views). Our approach, on the other hand, is based on a powerful recognition framework which can handle both 2-D and 3-D objects, i.e., AFoVs. Also, we used real objects in our experiments whereas the results reported in [31] are based on artificialdata only. In a different application, GAs were used for image registration [32]. The main idea in that work was applying GA-TS to register a set of images. This approach has similarities with our GA-TS approach; however, the problem considered here is different. Also, our search is more efficient since the GA searches the most feasible regions of the TS.

The remainder of this paper is organized as follows. Section II presents an overview of the theory of AFoVs. A brief review of GAs is provided in Section III. Section IV discusses the two solution spaces: the IS and the TS. The methodology used to estimate the value intervals for the parameters of the combination scheme is described in Section V. This section also presents the genetic search approaches. Specifically, we present the encoding mechanism, the selection scheme, genetic operators, and fitness function used. Section VI includes our experimental results and finally, our conclusions are given in Section VII.

II. Background on Algebraic Functions of Views

AFoVs were first introduced by Ullman and Basri [7]. In particular, it was shown that if we let an object undergo 3-D rigid transformations, (i.e., rotations and translations in space), and we assume that the images of an object are obtained by orthographic projection followed by a uniform scaling, then any novel view of the object can be expressed as a linear combination of three other views of the same object. Specifically, consider three reference views of the same object V_1, V_2 and V_3, which have been obtained by applying different rigid transformations, each view represented by a set of "interest" points (e.g., corners and junctions). Take now three points $p' = (x', y')$, $p'' = (x'', y'')$, and $p''' = (x''', y''')$, one from each view, which are in correspondence. If V is a novel view of the same object, obtained by applying a different rigid transformation, and $p = (x, y)$ is a point that is in correspondence with p', p'', and p''', then the coordinates of p' can be expressed in terms of the coordinates of p', p'', and p''' as the following:

$$x = a_1x' + a_2x'' + a_3x''' + a_4 \quad (1)$$
$$y = b_1y' + b_2y'' + b_3y''' + b_4 \quad (2)$$

where the parameters $a_j, b_j, j = 1, \ldots, 4$ are the same for all the points that are in correspondence across the four views. This idea is illustrated in Fig. 1. The parameters a_j and b_j can be recovered by solving a linear system of equations, given that we know at least four point correspondences across the views.

The above result can be simplified if we assume that the object undergoes a 3-D linear transformation in space (i.e., remove the orthonormality constraint associated with the rotation matrix). In this case, the AFoVs are simpler and involve only two reference views. Specifically, consider two reference views V_1 and V_2 of the same object which have been obtained by applying different linear transformations and two points $p' = (x', y')$, $p'' = (x'', y'')$, one from each view, which are in correspondence. Then, given a novel view V of the same object, obtained by applying another linear transformation, and a point $p = (x, y)$, which is in correspondence with points p' and p'', the coordinates of p can be expressed as a linear combination of the coordinates of p' and p'' as the following:

$$x = a_1x' + a_2y' + a_3x'' + a_4 \quad (3)$$
$$y = b_1x' + b_2y' + b_3x'' + b_4 \quad (4)$$

where the parameters a_j, b_j, $j = 1, \ldots, 4$ are the same for all the points that are in correspondence across the three views. It should be noted that (3) and (4) can be rewritten using the y coordinates of the second reference view instead.

TABLE II
Computed Value Intervals for the 3-D Object

Ranges of values			
a1 range	a2 range	a3 range	a4 range
[-0.4193, 0.4193]	[-0.3623, 0.3623]	[-0.4292, 0.4292]	[0 1]

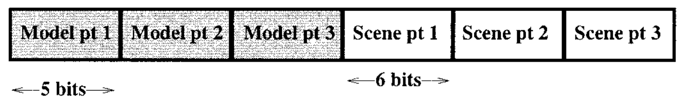


Fig. 5. Chromosome contains the binary encoded points to be matched between the model and scene.

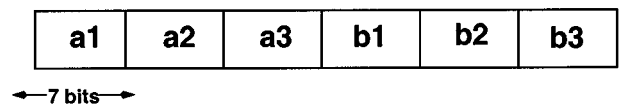


Fig. 6. Chromosome contains the binary encoded parameters.

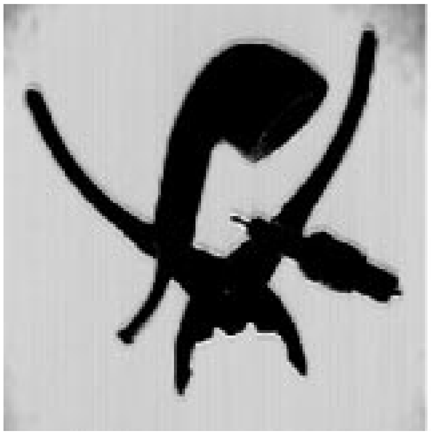
Although it was not explicitly discussed in [7], algebraic functions, involving one reference view, also exist in the case of planar objects. This is because in the case of planar objects, scaled orthographic projection is equivalent to a 2-D affine transformation [4]. Consider a reference view V_1 and a point $p' = (x', y')$. Given a novel view V and a point $p = (x, y)$ that is in correspondence with p', the coordinates of p are related to the coordinates of p' as the following:

$$x = a_1x' + a_2y' + a_3 \quad (5)$$
$$y = b_1x' + b_2y' + b_3. \quad (6)$$

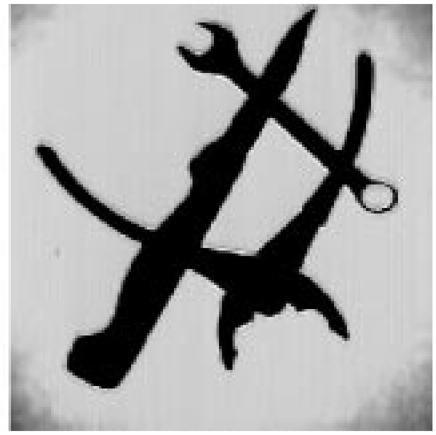
Fig. 2(b)-(d) shows new views of the object in Fig. 2(a), obtained using (5) and (6).

The extension of AFoVs in the case of perspective projection has been considered by several authors [8], [12]. In particular, it has been shown that three perspective views of an object satisfy a trilinear function. Moreover, Shashua [8] has shown that a simpler and more practical pair of algebraic functions exist when the reference views have been obtained under scaled orthographic projection (one perspective and two orthographic views satisfy a bilinear function). More results exist that support the validity of the theory of AFoVs under various projection models, transformations, and object types (paraperspective projection [10], rigid or nonrigid objects [9], and objects with smooth or nonsmooth surfaces [13]).

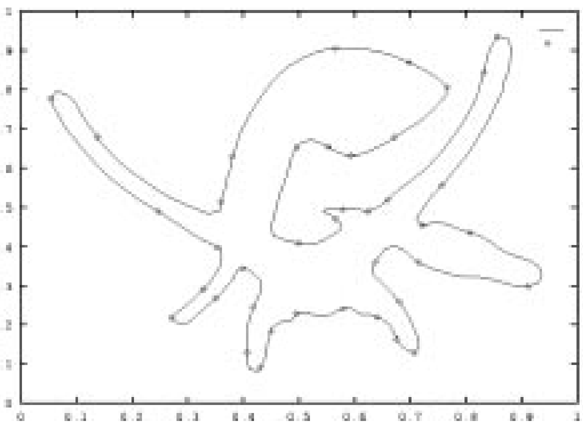
In this paper, we have used (5) and (6) for 2-D recognition and (3) and (4) for 3-D recognition. In general, (3) and (4) will not give satisfactory results when perspective distortions are large to be neglected. This is the case when the objects are close to the camera. When the distance of an object from the camera is large compared to its depth, however, it is well known that perspective projection can be approximated by scaled orthographic projection [4], [39]. In this case, (3) and (4) will give good results as is also verified by our experimental results.



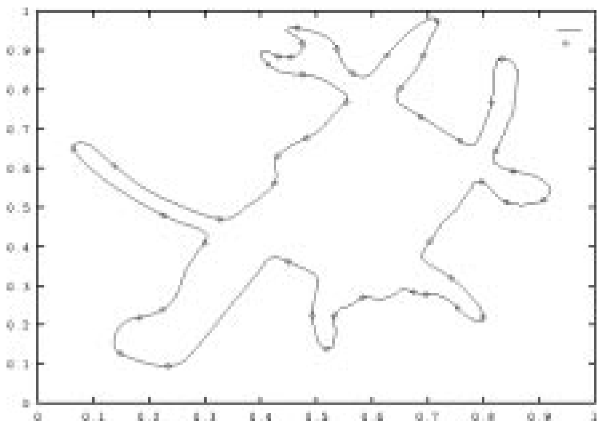
(a)



(b)



(c)



d)

Fig. 7. Some of the test scenes used in our 2-D experiments. (a) First test scene. (b) Second test scene. (c) Interest points extracted from the first test scene. (d) Interest points extracted from the second test scene. Note that we have intentionally left out the internal contour of the scene to make the problem more challenging.

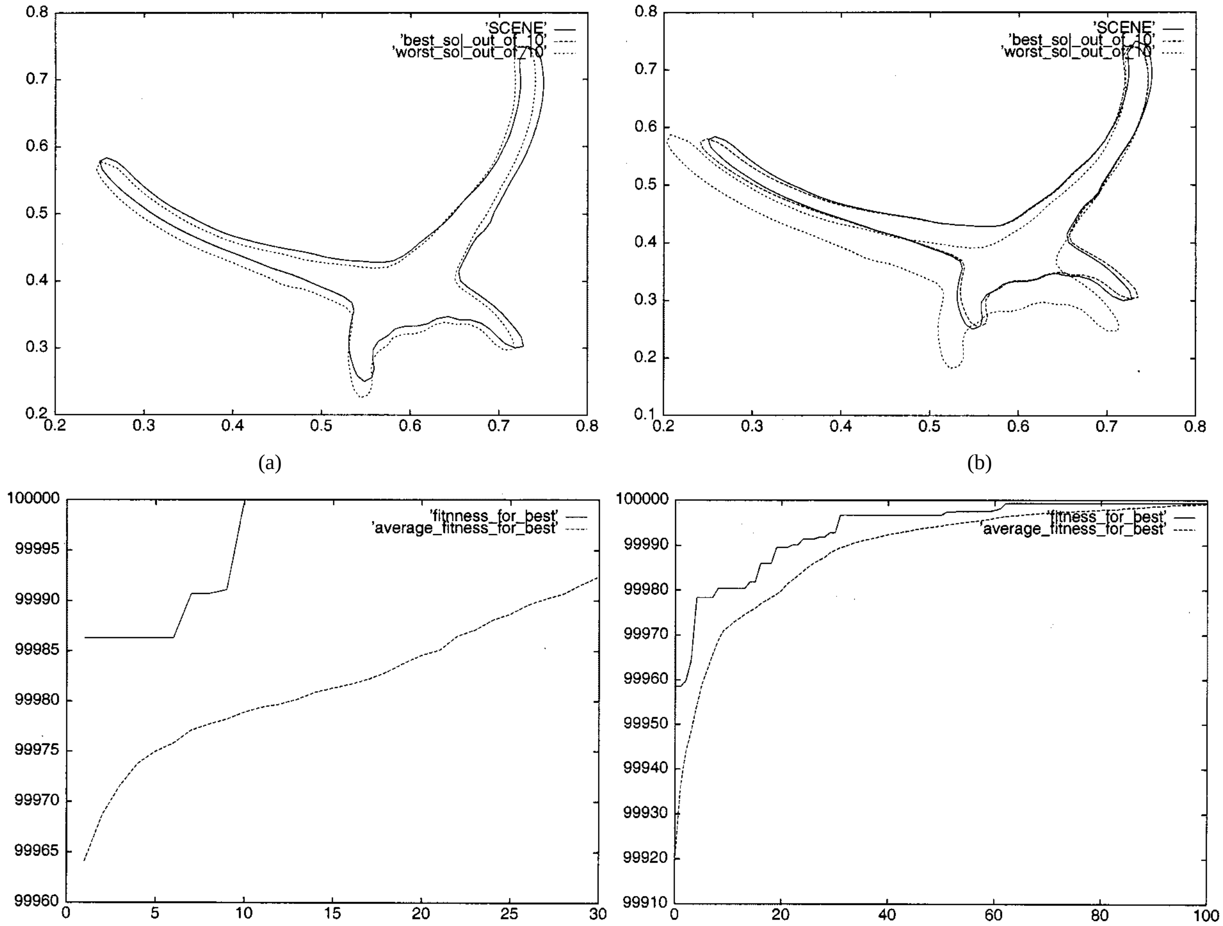
III. Solution Spaces

In general, object recognition approaches operate in one of the following two solution spaces: the IS [39] or the TS [37]. IS techniques begin by first extracting a collection of scene features (e.g., points corresponding to curvature maxima or zero-crossing). Then, a correspondence between these features and a set of model features is hypothesized. The transformation that aligns the model with the scene is then determined by this hypothesis. In the case of similarity transformations, e.g., at least two point correspondences between the model and the scene are required. Fig. 3 illustrates a simple case. The open circles correspond to model point features and the black circles correspond to scene point features. *Match1* and *Match2* are two different hypothetical matches. Based on these hypothetical matches, we compute a similarity transformation to align the model with the scene. As can be seen, *Match1* is a good match since it aligns more model points with the scene (i.e., not just the hypothesized two-point correspondences). On the other hand, *Match2* is a bad match since no more model points, besides the ones hypothesized, have been aligned with the scene.

Since there is usually no *a priori* knowledge of which model features correspond to which scene features, recognition can be computationally too expensive, even for relatively simple scenes. Assuming M model points and S image points, the maximum number of possible alignments is of the order of $O(M^k S^k)$, where k is the minimum number of point matches required. Usually, due to errors in the location of the scene points, more than k point correspondences are required to compute the corresponding transformation more accurately.

The same arguments hold true using AFoVs for object recognition. Specifically, given an unknown scene containing an instance of the model to be recognized, we need to match sets of scene points to sets of points from known views of the model. From each match, the parameters of the views combination that predict the appearance of the model in the scene can be computed. The number of points that need to be matched depends on the the number of the parameters of the combination [i.e., three-point matches are required if (5) and (6) are used and four-point matches are required if (3) and (4) are used]. Our first approach involves using GA-IS to find the point matches efficiently.

Alternatively, TS techniques deal with the space of possible geometric transformations between the model and the scene. The objective is to search the space of all possible transformations in order to find a transformation which would bring a large number of model points into alignment with the scene. In the case of similarity transformations, e.g., the TS is four-dimensional. Using AFoVs, the dimensionality of the TS is determined by the parameters of the view combination scheme



(a) (b) (c) (d)

Fig. 8. Best and worst solutions found by (a) and (d) GA-IS and (b) and (d) GA-TS for *Scene1*. The x axis corresponds to the x coordinates of the contour and the y axis to the y coordinates. (c) and (d) Performance plots for the best solutions (the horizontal axis corresponds to number of generations, while the vertical one to the fitness).

[i.e., six parameters if (5) and (6) are used and eight parameters if (3) and (4)are used]. Searching the TS is very expensive computationally due to the large number of possible parameter values. Our second approach involves using GA-TS to find the best transformation efficiently.

In general, the TS being much larger than the IS would imply that the GA-TS approach has more work to do. Without somehow bounding the parameters of the combination scheme, it is difficult and time consuming for the GA to find good solutions. To facilitate GA-TS, it is important that we restrict genetic search to a smaller subspace of the TS. In the next Section, we describe a method based on SVD [19] and IA [20] for estimating the intervals of values that the parameters of the combination scheme can assume.

IV. ESTIMATING THE PARAMETERS' VALUE INTERVALS

In this section, we consider the case of orthographic projection under the assumption of linear transformations. However, the same methodology can be applied on all other cases. As discussed in Section II, two reference views V_1 and V_2 must be combined in order to obtain a new view V [see (3) and (4)]. Given the point correspondences across the reference and new views, the following system of equations should be satisfied:

$$\begin{bmatrix} x'_1 & y'_1 & x''_1 & 1 \\ x'_2 & y'_2 & x''_2 & 1 \\ \cdots & \cdots & \cdots & \cdots \\ x'_N & y'_N & x''_N & 1 \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \\ a_4 & b_4 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \cdots & \cdots \\ x_N & y_N \end{bmatrix} \tag{7}$$

where $(x'_1, y'_1), (x'_2, y'_2), \ldots, (x'_N, y'_N)$, and $(x''_1, y''_1), (x''_2, y''_2), \ldots (x''_N, y''_N)$ are the coordinates of the points in the reference views V_1 and V_2, respectively, and (x_1, y_1) , $(x_2, y_2) \ldots (x_N, y_N)$ are the coordinates of the points in the new view V. Splitting the above system into two subsystems, we have

$$Pc_1 = p_x \tag{8}$$

$$Pc_2 = p_y \tag{9}$$

where the columns of the P matrix are shown in (7), c_1 and c_2 are vectors corresponding to a_j's and b_j's (the parameters of the combination scheme), and p_x, p_y are vectors corresponding to the x and y coordinates of the new view. To solve (8) and (9), a least-squares approach, such as SVD [19], can be used. Using SVD, P can be factorized as $P = U_P W_P V_P^T$, where both U_P and V_P are orthonormal matrices, while W_P is a diagonal

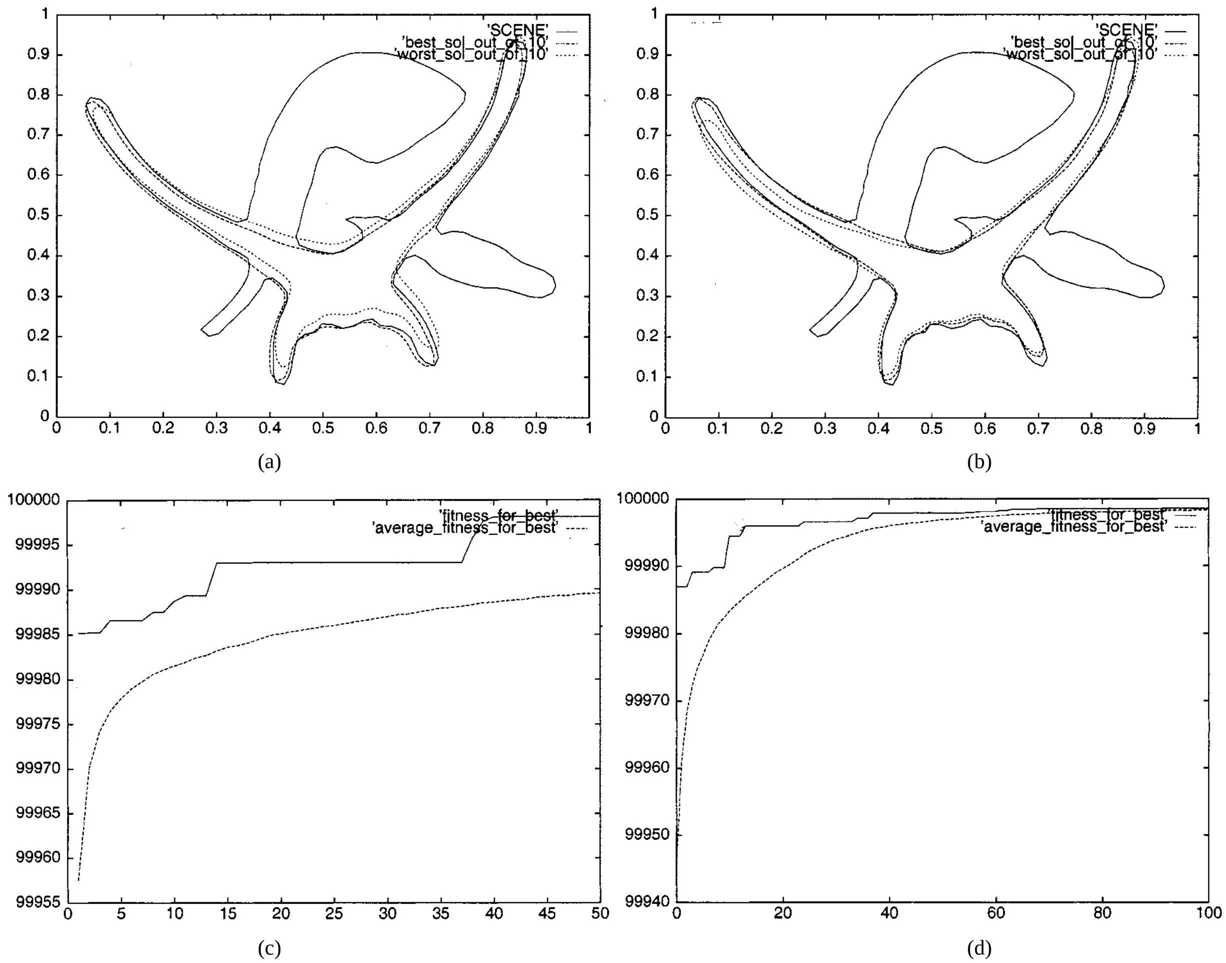


Fig. 9. Best and worst solutions found by (a) and (c) GA-IS and (b) and (d) GA-TS for *Scene2*. The x axis corresponds to the x coordinates of the contour and the y axis to the y coordinates. (c) and (d) Performance plots for the best solutions (the horizontal axis corresponds to number of generations, while thE vertical one to the fitness).

matrix whose elements w_{ii}^P are always nonnegative (the singular values of P).The solutions of the above two systems are $c_1 = P^+ p_x$ and $c_2 = P^+ p_y$, where P^+ is the pseudoinverse of P. Assuming that P has been factorized, its pseudoinverse is $P^+ = V_P W_P^+ U_P^T$, where W_P^+ is also a diagonal matrix with elements $1/w_{ii}^P$ if w_{ii}^P is greater than zero (or a very small threshold in practice) and zero otherwise. The solutions of (8) and (9) are given by the following [19]:

$$c_1 = \sum_{i=1}^{k} \left(\frac{u_i^P p_x}{w_{ii}^P} \right) v_i^P \tag{10}$$

$$c_2 = \sum_{i=1}^{k} \left(\frac{u_i^P p_y}{w_{ii}^P} \right) v_i^P \tag{11}$$

where u_i^P is the ith column of matrix U_P, v_i^P is the ith column of matrix V_P, and $k = 4$. To determine the range of values for c_1 and c_2, we assume first that the novel view has been scaled such that the x and y coordinates belong within a specific interval. This can be done, e.g., by mapping the novel view to the unit square. In this way, its x and y image coordinates will be mapped to the interval [0,1]. To determine the range of values for c_1 and c_2, we need to consider all possible solutions of (8) and (9), assuming that p_x and p_y belong to [0,1]. We are using IA [20] to solve this problem. In IA, each variable is represented as an interval of possible values. Given two interval variables $t = [t_1, t_2]$ and $r = [r_1, r_2]$, the sum and the product of these two interval variables are defined as the following [20]:

$$t + r = [t_1 + r_1, t_2 + r_2] \tag{12}$$

$$t^* r = [min(t_1 r_1, t_1 r_2, t_2 r_1, t_2 r_2), max(t_1 r_1, t_1 r_2, t_2 r_1, t_2 r_2)]. \tag{13}$$

Applying IA to (10) and (11), instead of the standard arithmetic, we can compute interval solutions for c_1 and c_2 by setting $p_x = [0, 1]$ and $p_y = [0, 1]$. In interval notation, we solve the systems

$$Pc_1 = p_x^I \tag{14}$$

$$Pc_2 = p_y^I \tag{15}$$

where the superscript I denotes an interval vector. The solutions c_1^I and c_2^I should be understood to mean $c_1^I = [c_1 : Pc_1 = p_x, p_x \epsilon p_x^I]$ and $c_2^I = [c_2 : Pc_2 = p_y, p_y \epsilon p_y^I]$. It should be mentioned that since the matrix P and the intervals for p_x, p_y are all the same, the interval solutions for c_1^I and c_2^I will be the same.We have applied this methodology on the models used in our experiments to estimate the parameter intervals. The first row of Table I shows the intervals computed for the object shown in Fig. 2(a) [planar object assuming (5) and (6)]. It can be shown that the width of the ranges depends on the condition of the matrix P and that the reference view(s) can be "preconditioned"

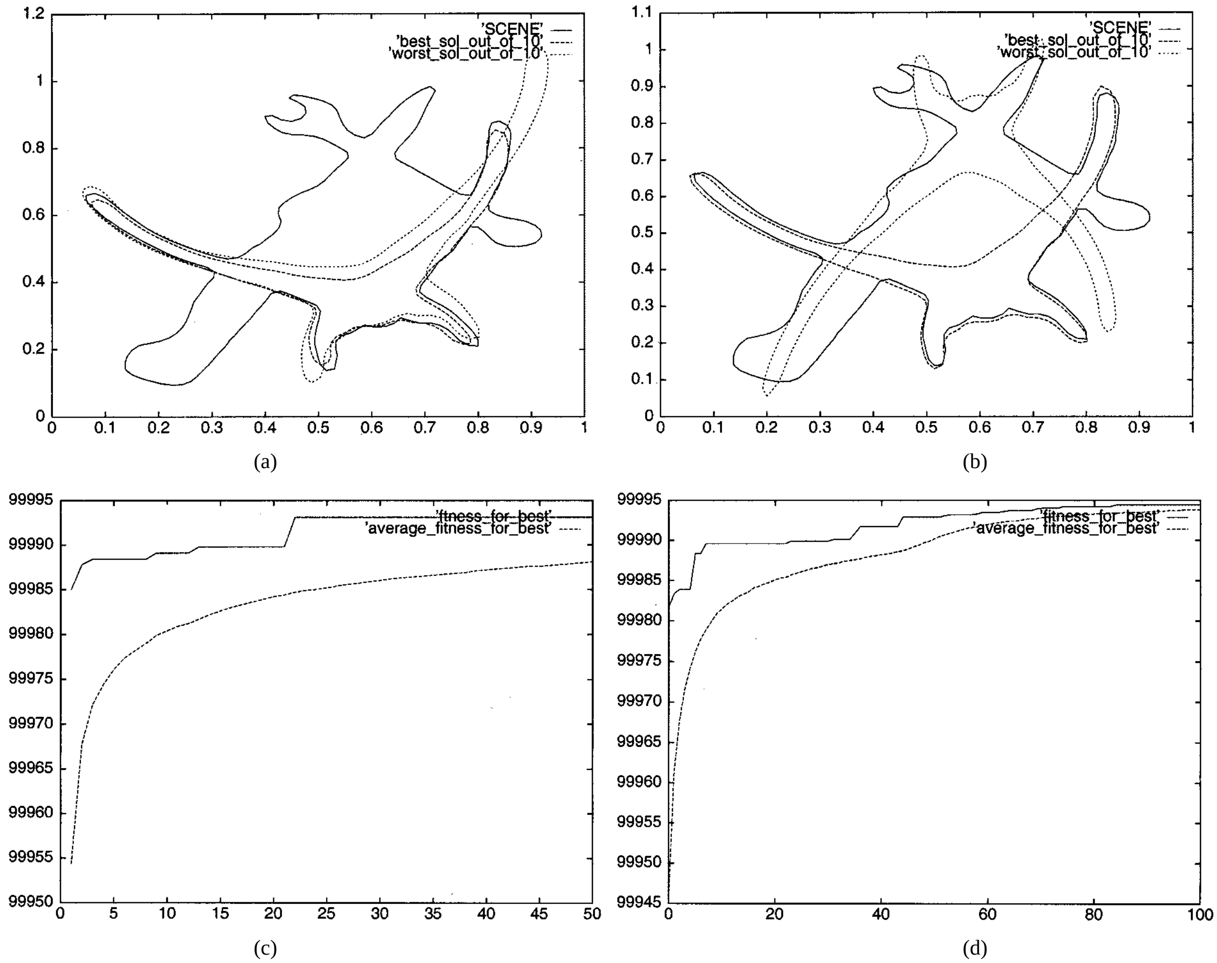


Fig. 10. Best and worst solutions found by (a) and (c) GA-IS and (b) and (d) GA-TS for *Scene3*. The x axis corresponds to the x coordinates of the contour and the y axis to the y coordinates. (c) and (d) Performance plots for the best solutions (the horizontal axis corresponds to number of generations, while the vertical one to the fitness).

TABLE III
SUMMARY OF RESULTS (GA-IS APPROACH)

Results			
Scene	Scene Points	Number of Matches	$GA-IS_{matches}$
Scene1	19	5,633,766	1800(0.0003)
Scene2	40	57,442,320	47,800(0.0008)
Scene3	45	82,500,660	133,250(0.0016)

TABLE IV
SUMMARY OF RESULTS (GA-TS APPROACH)

Results	
Scene	$GA-TS_{matches}$
Scene1	8010 (0.000000018)
Scene2	8760(0.00000002)
Scene3	8620(0.00000002)

to narrow the parameter intervals (see [22]–[24]). By preconditioning, we mean an appropriate transformation that maps the reference views to new reference views, yielding a matrix P with good condition number. The second row of Table I shows the tighter intervals obtained using this procedure. The model and reference views used in our 3-D experiments are shown in Fig. 4. Table II shows the interval values obtained in this case (after preconditioning).

V. METHODOLOGY

In this section, we present the genetic search approaches. Specifically, we present the encoding mechanism, the selection scheme, genetic operators, and the fitness function used.

A. Image Space Encoding

A simple encoding scheme where the identity of the points to be matched made up the alleles in the genotype gave good results on all the scenes we tried in our experiments. The parameters are provided in the next section. Using (5) and (6), three pairs of points are needed to compute the parameters of the combination scheme. Thus, the chromosome contains the binary encoded identities of the three pairs of points. The model used in our experiments [see Fig. 2(a)] was represented by 19 points, corresponding to curvature extrema and zero crossings [38]. This required $\lceil \log 19 \rceil = 5$ bits per point. The scenes used in our experiments had between 19 and 45 points (extracted in the same way) and required either five or six bits per point. We did not check for repeated points and used simple two-point crossover and point mutation. A simple linear transformation was used to map values from the actual range to the desired one (e.g., from [31] using five bits to [18] assuming 19 interest model points). Fig. 5 illustrates the encoding scheme.

B. Transformation Space Encoding

A simple binary encoding scheme was also used to represent solutions in the TS. In the case of 2-D objects, each chromosome contains six fields, with each field corresponding to one of the six parameters of (5) and (6). Fig. 6 illustrates the encoding scheme. In the case of 3-D objects, each chromosome contains eight fields with each field corresponding to one of the eight parameters of (3) and (4). Only the range (difference between the maximum and minimum values) needs to be represented. In the 2-D case, for example, a_1 assumes values in the interval $[-0.408, 0.408]$ (see second row of Table I). Thus, its range is $r = 0.408 - (-0.408) = 0.816$. Assuming that up to two decimal points are important in the estimation of the parameters, 82 possible values ($\lceil 0.816 \times 100 \rceil + 1$) should be encoded. This means that seven bits are enough to encode a_1's range. It only needs to represent values from zero to 81. As a result, it is possible for the GA to find solutions that are not within the desired range (i.e., [0, 81]). To deal with this problem, a simple transformation is used to map values from [0, 127] to [0, 81]. Assuming that W is a binary encoded solution corresponding to a_1, a_1's actual value is obtained as the following:

$$a_1 = \mathrm{MIN}(a_1) + (\frac{82}{2^7})^* \mathrm{Decimal}(W)$$

where $\mathrm{MIN}(a_1) = -0.408$ and $\mathrm{Decimal}(W)$ is the decimal representation of W. The decoding of the other parameters is performed in a similar way. The constant $(82/2^7)$ is used to map values from [0, 127] to [0, 81].

C. Fitness Evaluation

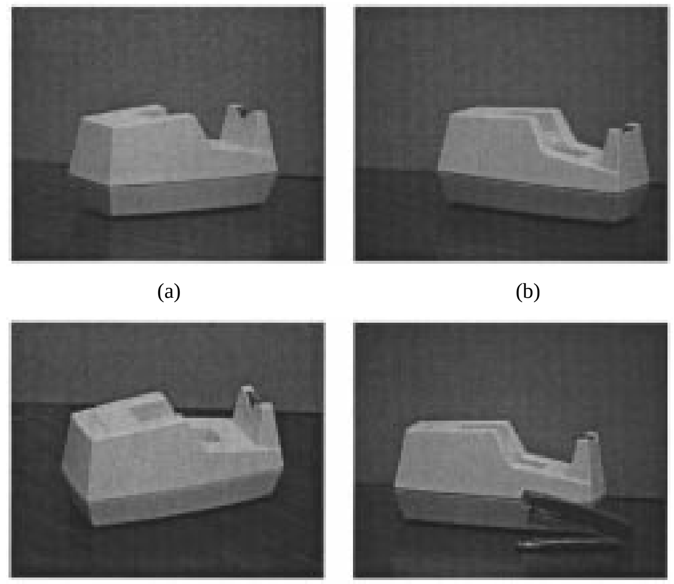
We evaluate fitness of individuals by computing the back-projection error (BE) between the model and scene. Specifically, to evaluate the goodness of the match specified by an individual in the case of the GA-IS approach, we first compute the parameters of the combination scheme which maps the model points to their corresponding scene points. This requires solving a system of six equations with six unknowns in the case of (5) and (6) and a system of eight equations with eight unknowns in the case of (3) and (4). After the parameters have been computed, we predict the appearance of the model in the scene and we compare it with its actual appearance (i.e., back-project the model onto the scene). This is the standard verification step used in object recognition [4], [5], [23], [39]. Finally, we compute the error BE between the back-projected model and the scene. To compute BE, for every model point, we find the closest scene point and then compute the distance d_j between these two points. The overall back-projection error is given by

$$BE = \sum_{i=1}^{M} {d_j}^2$$

where M is the number of model points. Since we need to maximize fitness but minimize the error, our fitness function is

$$\mathrm{Fitness} = \mathrm{Max} - BE$$

where the constant Max is used to change the minimization problem to a maximization one. Max was set to 10 000 in all of our experiments.



(a) (b) (c) (d)

Fig. 11. Some of the test scenes used in our 3-D experiments. (a) First test scene is the same as the first reference view. (b) Second test scene was obtained by transaltion and 3-D rotation around the y axis. (c) Third test view was obtained by moving the camera higher and closer to the object (i.e., to introduce more perspective distortions). (d) Fourth test view demonstrates some degree of occlusion.

To evaluate fitness of individuals in the case of the GA-TS approach, we follow exactly the same procedure except that we do not have to solve for the parameters of the combination scheme. This is because the parameters are directly encoded in the chromosome and all that is required is to decode it by following the procedure outlined in the previous section.

VI. Experiments and Results

We have performed a number of experiments using both 2- and 3-D objects. In the case of 2-D object recognition, we used the set of equations given by (5) and (6), while in the case of 3-D object recognition, we used the set of equations given by (3) and (4). A number of increasingly complex recognition tasks were used to demonstrate the proposed approaches. Our selection strategy was cross generational. Assuming a population of size N, the offspring double the size of the population and we select the best N individuals from the combined parent–offspring population for further processing [39]. This kind of selection does well with small populations and leads to quick convergence (sometimes prematurely). We also linearly scale fitnesses to impose a constant selection pressure [17].

A. 2-D Object Recognition Experiments

The scenes we used in our experiments are shown in Figs. 7 and 8. All GA parameters were identical except for the population size and number of generations. We used a crossover probability of 0.95, a mutation probability of 0.05, and a scaling factor of 1.2. The population sizes were set to 100, 200, and 500 for scenes *Scene1*, *Scene2*, and *Scene3*, respectively. The number of generations needed in the case of the GA-TS approach were

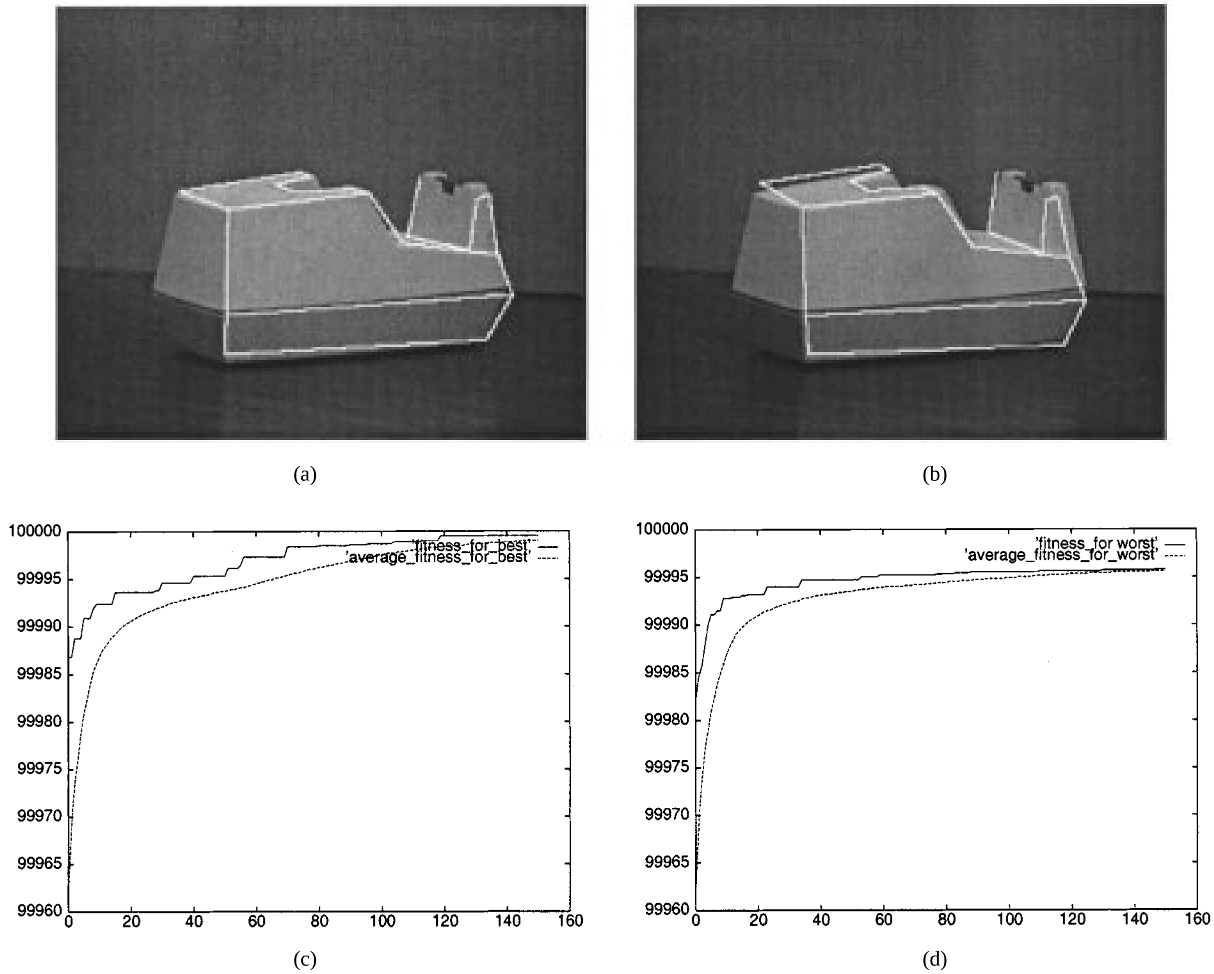


Fig. 12. (a) and (d) Best and (b) and (d) worst solutions for *Scene5*. (c) and (d) Performance plots (the horizontal axis corresponds to number of generations, while the vertical one to the fitness).

twice as many than in the case of the GA-IS approach. In particular, less than 30 generations were required by the GA-IS approach for *Scene1* and less than 50 for *Scene2* and *Scene3*.

In the case of the GA-TS approach, approximately 100 generations were required for each scene. This can be justified by the fact that the TS is larger than the IS. It should be mentioned, however, that although the GA-TS approach required more generations to converge, this does not mean that it is necessarily slower than the GA-IS approach. The reason is that the GA-TS approach does not have to solve for the parameters of the combination (i.e., they are directly encoded in the chromosome), a time-consuming step. For each scene, we ran each approach ten times with different random seeds. Performance plots indicate that the GA very quickly predicts roughly the correct appearance of the model in the scene and then spends most of its time making little progress.

Scene1 was chosen to be the same as the reference view. Exact mappings (i.e., zero back-projection error) were found using the GA-IS approach in nine out of ten trials. Fig. 8(a) shows the best and worst solutions found (the solid lines represents the scene and the dashed lines correspond to the best and worst solutions). In the case of the best solution, the GA has predicted the model appearance exactly, thus, the model and the scene overly each other and present a single outline. Fig. 8(c), shows thefitness corresponding to the best solution (solid line) and average fitness (dashed line) corresponding to that solution. As it can be observed, the GA-IS approach found this solution within the first ten generations or so. In the case of GA-TS approach, almost exact mappings (i.e., very small back-projection error, but not zero) were found in all cases. Fig. 8(b) shows the best and worst solutions. Fig. 8(d) shows the best and average fitnesses for the best solution found by the GA-TS approach. Comparing the performance curves in both cases, the convergence of the GA-TS approach is slower.

Results for *Scene2* and *Scene3* are shown in Figs. 9 and 10. The best and worst solutions found by the two approaches in the case of *Scene2* are comparable. In the case of *Scene3*, the GA-IS approach found the correct solution in all ten trials, while the GA-TS approach missed the correct solution once. This case is actually shown in Fig. 10(b). It should be mentioned, however, that *Scene3* is considerably more difficult than *Scene2* since a large part of the boundary of the object to be recognized is missing (the boundary was removed intentionally to make the problem harder). More experiments have shown that finding the wrong solution can be alleviated by increasing the population size and the number of generations. Performance plots in both cases illustrate similar results (GA-IS converges faster than GA-TS).

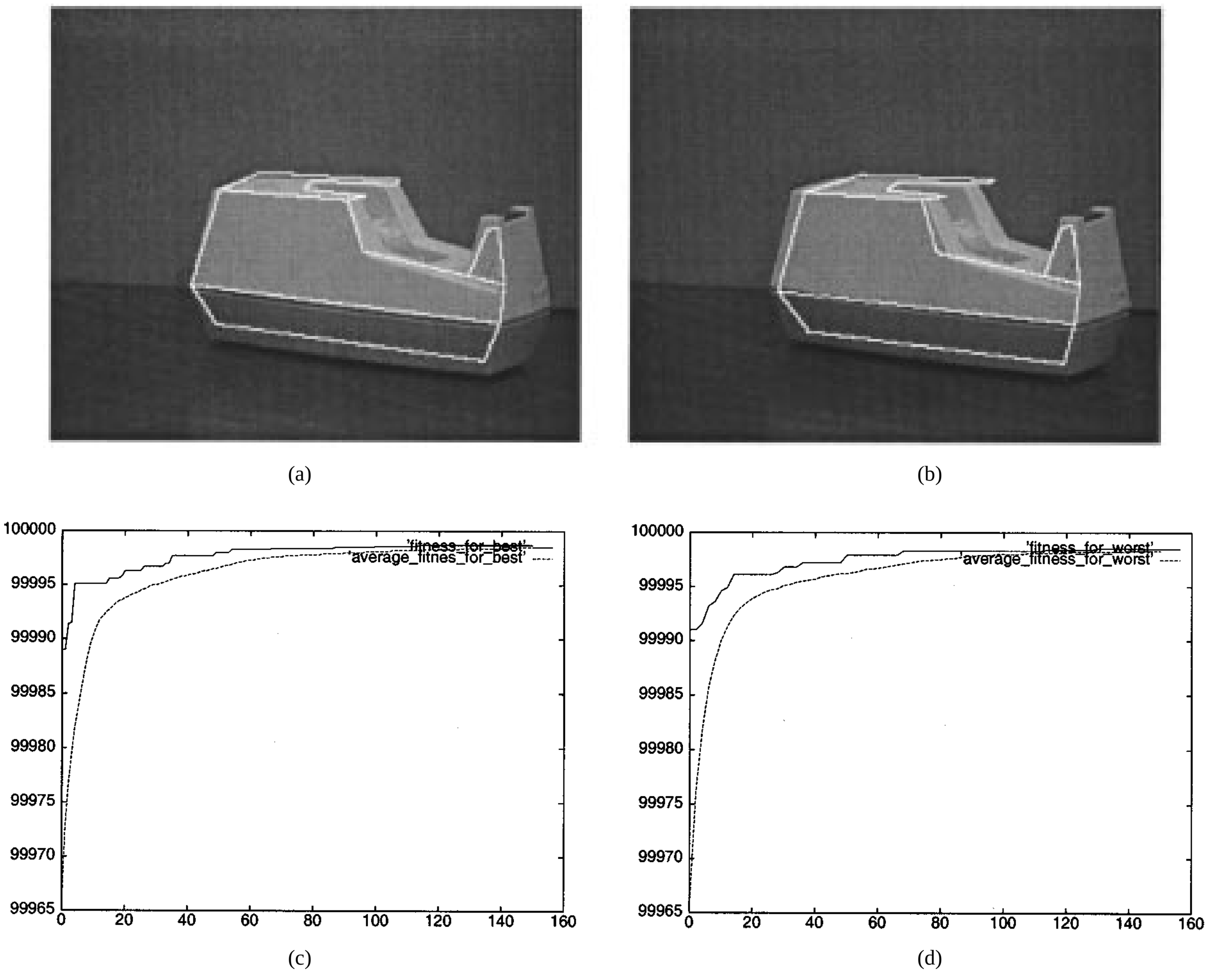


Fig. 13. (a) and (c) Best and (b) and (d) worst solutions for *Scene6*. (c) and (d) Performance plots (the horizontal axis corresponds to number of generations, while the vertical one to the fitness.

Table III provides a summary of our results in the case of the GA-IS approach. The first column specifies the scene. The second and third columns describe the size of the test problems.

The columns list (in order) the number of scene points and the size of search space. The last column indicates GA-IS's effort in terms of the number of matches explored and the corresponding fraction in the searched space. In our experiments, the number of model points M is 19; thus, the number of possible triplets M_3 is

$$M_3 = \binom{19}{3} = 969.$$

The order of points matters in computing the total number of possible matches between model and scene points and is, thus, given by $3!M_3S_3$, where S_3 is the number of possible scene triplets. Table IV presents similar results for the case of the GA-TS approach. The number of values we need to consider in order to represent a_1's range is 82 (see our discussion in Section VI-B). In the case of a_2, we need to consider 79 values, while in the case of a_3, we need to consider 101 values. The values for b_1, b_2 and b_3 are the same (same interval solutions — see our discussion in Section V). The total number of possible transformations is, thus, $82^2 \times 79^2 \times 101^2 = 428\ 079\ 701\ 284$. The second column of Table IV indicates the number of matches the GA-TS approach searched through. The numbers in the parentheses have the same meaning as before.

B. 3-D Object Recognition Experiments

In this section, we report experiments using 3-D objects. Since the GA-TS approach performed very satisfactorily in the previous experiments and does not require any feature extraction, we have not considered the GA-IS approach here. The 3-D model used in the experiments is shown in Fig. 4. We have used two views to model the aspect of the object shown in Fig. 4 since we employ (3) and (4). Some of the test scenes used in our experiments are shown in Fig. 11 (*Scene5–Scene8*).

The crossover probability, mutation probability, and scaling factor were the same as before. The population size was set to 200 in all the following experiments. Approximately 150 generations were required for each scene. Each experiment was ran ten times with different random seeds. Performance plots indicate again that the GA very quickly predicts the appearance of the model in the scene and then spends most of its time making little progress.

Scene5 was chosen to be exactly the same as one of our reference views. Fig. 12 shows the best and worst solutions found. The GA-TS approach had no difficulty finding good solutions in all runs. The object in *Scene6* has undergone rotation around

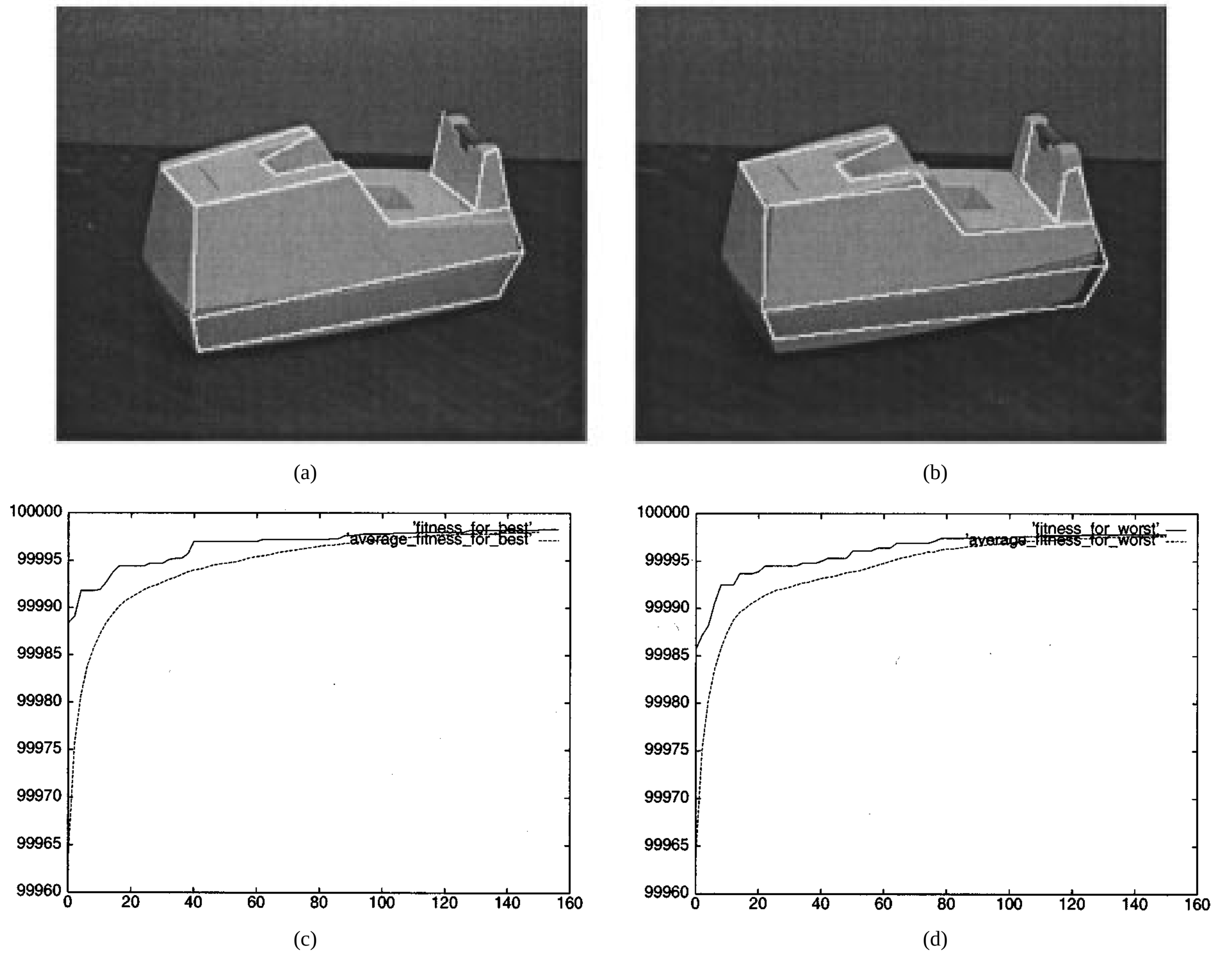


Fig. 14. (a) and (c) Best and (b) and (d) worst solutions for *Scene7*. (c) and (d) Performance plots (the horizontal axis corresponds to number of generations, while the vertical one to the fitness.

the y axis. However, all of the features present in the reference views are also present in *Scene6*. Fig. 13 shows the best and worst solutions found as well as the fitness and average fitness graphs associated with them. *Scene7* is more interesting since the camera has been moved higher and closer to the object. The reason we moved the camera closer was to test the tolerance of the (3) and (4) to perspective distortions (see Section II) and the ability of GAs finding good solutions in such cases. Fig. 14 shows good results were obtained in all cases. Finally, *Scene8* shows the model with part of it occluded by a stapler. Due to the occlusion, the matches found are not as good as in the other cases. However, they can serve as starting points to local optimization techniques (see next section).

To demonstrate the efficiency of the proposed approach, we compare the number of transformations tried by the GA-TS approach versus the total number of transformations. The number of values used to represent a_1's range is 84 (see our discussion in Section VI-B). For a_2, we used 73 values, 86 values for a_3, and 101 values for a_4. The values for b_1, b_2, b_3, and b_4 were the same (i.e., same interval solutions). Thus, the total number of possible transformations is $84^2 \times 73^2 \times 86^2 \times 101^2$, which is of the order of 2×10^{15}. The last column of Table V shows the number of transformations tried by the GA-TS approach. Obviously, the GA searches a very small portion of the TS. Comparing the GA approach with our previous object recognition approach [22], the solutions found by the two approaches are of comparable quality (i.e., similar back-projection errors). Our previous approach finds more accurate alignments since the parameters of the AFoVs are computed explicitly by considering feature matches and solving a system of linear equations. However, there is not a significant difference in the quality of the solutions found by the two approaches, at least for recognition purposes. It should be emphasized, however, that the two approaches are fundamentally different and have different capabilities. Our previous approach is an indexing-based approach, which trades space for time. Briefly, indexing is a mechanism that, when provided with a key value, allows rapid access to some associated data. Thus, instead of searching the space of all possible objects and their appearances and explicitly reject in-

TABLE V
SUMMARY OF RESULTS

Results	
Scene	$GA_{transformations}$
Scene1	19,600
Scene2	37,600
Scene3	25,600
Scene4	37,000

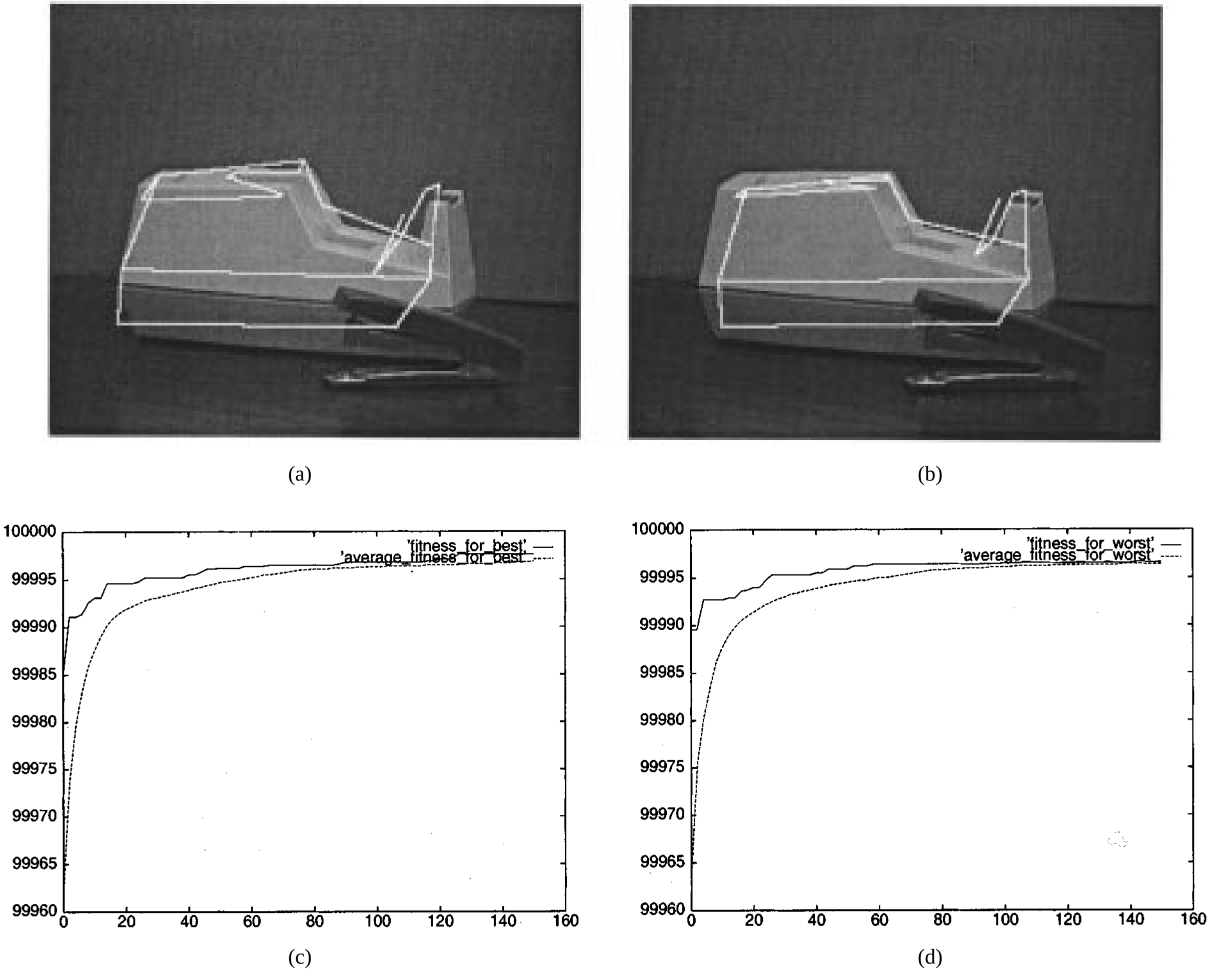


Fig. 15. (a) and (c) Best and (b) and (d) worst solutions for *Scene8*. (c) and (d) Performance plots (the horizontal axis corresponds to number of generations, while the vertical one to the fitness.

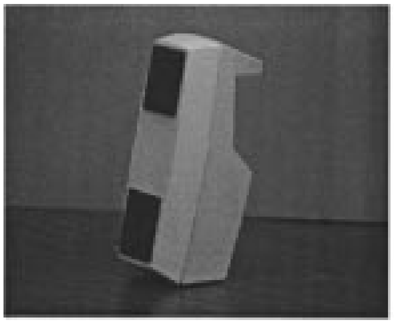
valid solutions through verification, indexing inverts the process so that only the most feasible predictions are considered. The idea is arranging the model appearances in an index space offline. During recognition, feasible solutions are found by indexing into this space.

Obviously, a major advantage of our previous approach is that it can search for multiple models in a scene simultaneously. The current implementation of our GA approach can search different appearances of the same model only. Suppose we are given a scene that contains more than one model, we would have to run the GA algorithm over the scene once for each model in order to recognize all the models. Extending the GA approach to searching for multiple models would require a more powerful encoding scheme (e.g., encoding the identity of the models in the chromosome). A disadvantage of our previous approach is that it is very memory consuming since it requires prestoring a lot of information in the index table. The GA approach, on the other hand, has very low memory requirements. In terms of time requirements, the GA approach seems to be able to find a rough alignment of the model with the scene very quickly (e.g., within 20 generations or so). We can, thus, claim that it is faster than our previous approach. However, this is not a fair comparison not only because we do account for the time required by the GA to improve the alignment, but also because our previous approach: 1) searches for multiple models in the scene , and 2) operates in the IS (i.e., more time is required for feature extraction and solving for the parameters of the AFoVs). The GA approach searches a smaller space since it looks for a single model in the scene. Moreover, it searches the TS explicitly, which implies that it does not have to extract any features, establish correspondences, or solve for the parameters of the AFoVs.

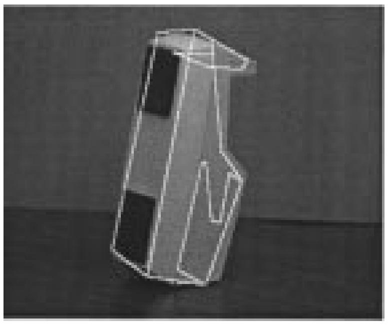
VII. Conclusion

In this paper, we considered using GAs to recognize real 2- or 3-D objects from 2-D intensity images. The recognition strategy used was based on the theory of AFoVs. Two different approaches were considered: GA-IS (i.e., feature matches among the views) and GA-TS (i.e., parameters of the algebraic functions). GA-TS was made more effective by searching small parameter intervals obtained using SVD and IA. Our experimental results demonstrate that GAs search these spaces efficiently and find good solutions.

It should be mentioned that although there are many problems for which GAs can find a good solution in reasonable time, there are also problems for which GAs are inappropriate. These are mainly problems for which it is important to find the exact global optimum. GAs do not perform well in these cases. In the context of our application, our expectation was that GAs will be



(a)



(b)

Fig. 16. Scene with self-occlusion. (a) Original scene. (b) Solution found overlaid on the original scene. The GA was able to perform a rough alignment of the model with the scene, which could be assumed to be used to initialize a local optimization technique.

able to perform at least a rough alignment of the model with the scene. The results obtained through our experiments show that GAs can find almost exact matches when there is little occlusion and near-exact matches when scenes contain occlusion (e.g., see Figs. 15 and 16). Near-exact matches are useful in the sense that can actually reduce the search space to a limited domain. Then, a local optimization technique can be used for finding an exact match. The preliminary stage of rough alignment may help preventing such methods from reaching a local minimum instead of the global one.

This work has two main limitations. First, we have only used two reference views in the case of 3-D object recognition. This is acceptable when different aspects of an object produce views containing many common features. When complex objects are considered (i.e., having very different aspects), more reference views are required to represent each aspect. Second, we do not search for multiple objects in the scene. If two or more objects need to be recognized, we need to run the GA approach several times, each time searching for a different object. For future research, we plan to extend the proposed approach so it can handle multiple aspects as well as multiple objects. This requires a more useful encoding scheme which encodes information about the identity of the object being searched as well as the views being considered for predicting its appearance in the scene.

REFERENCES

[1] R. Chin and C. Dyer, "Model-based recognition in robot vision," *Comput. Surveys*, vol. 18, no. 1, pp. 67–108, 1986.

[2] P. Suetens and P. Fua, "Computational strategies for object recognition," *ACM Comput. Surveys*, vol. 24, no. 1, pp. 5–61, 1992.

[3] I. Weiss, "Geometric invariants and object recognition," *Int. J. Comput. Vis.*, vol. 10, no. 3, pp. 207–231, 1993.

[4] D. Jacobs, "Matching 3-D models to 2-D images," *Int. J. Comput. Vis.*, vol. 21, no. 1/2, pp. 123–153, 1997.

[5] Y. Lamdan, J. Schwartz, and H. Wolfson, "On recognition of 3-D objects from 2-D images," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1988, pp. 1407–1413.

[6] D. Clemens and D. Jacobs, "Space and time bounds on indexing 3-D models from 2-D images," *IEEE Pattern Anal. Machine Intell.*, vol. 13, pp. 1007–1017, Oct. 1991.

[7] S. Ullman and R. Basri, "Recognition by linear combination of models," *IEEE Pattern Anal. Machine Intell.*, vol. 13, pp. 992–1006, October 1991.

[8] A. Shashua, "Algebraic functions for recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 779–789, Aug. 1995.

[9] R. Basri, "Recognition by combinations of model views: Alignment and invariance," in *Applications of Invariance in Computer Vision*, J. M. Mundy, A. Zisserman, and D. Forsyth, Eds., 1994, pp. 435–450.

[10] ——, "Paraperspective ≡ Affine," Technical Report Dept. of Applied Mathematics, The Weizmann Institute of Science.

[11] A. Shashua, "Trilinearity in visual recognition by alignment," in *Proc. 3rd Eur. Conf. Computer Vision*, 1994, pp. 479–484.

[12] O. Faugeras and L. Robert, "What can two images tell us about a third one?," in *Proc. 3rd Eur. Conf. Computer Vision*, 1994, pp. 485–492.

[13] R. Basri and S. Ullman, "The alignment of objects with smooth surfaces," *Comput. Vis. Graph. Image Process.*, vol. 57, no. 3, pp. 331–345, 1993.

[14] H. Bulthoff and S. Edelman, "Psychophysical support for a 2-D view interpolation theory for object recognition," in *Proc. Nat. Acad. Sci.*, vol. 89, Jan. 1992, pp. 60–64.

[15] M. Tarr, "Rotating objects to recognize them: A case study of the roleof viewpoint dependency in the recognition of three-dimensional obejcts," *Psychonomic Bull. Rev.*, vol. 2, no. 1, pp. 55–82, 1995.

[16] J. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press, 1975.

[17] D. Goldberg, "Genetic algorithm in search, optimization, and machine learning," in *Reading*. Reading, MA: Addison-Wesley, 1989.

[18] D. B. Fogel, *Evolutionary Computation*. Piscataway, NJ: IEEE Press, 1995.

[19] W. Press *et al.*, *Numerical recipes in C: The Art of Scientific Programming*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[20] R. Moore, *Interval Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1966.

[21] E. Hansen and R. Smith, "Interval arithmetic in matrix computations: Part II," *SIAM Journal of Numerical Analysis*, vol. 4, no. 1, 1967.

[22] G. Bebis, M. Georgiopoulos, M. Shah, and N. d. V. Lobo, "Indexing based on algebraic functions of views," *Comput. Vis. Image Understand.*, vol. 72, no. 3, pp. 360–378, 1998.

[23] ——, "Algebraic functions of views for model-based object recognition," in *Proc. 6th Int. Conf. Computer Vision*, Jan. 1998, pp. 634–639.

[24] G. Bebis, M. Georgiopoulos, N. d. V. Lobo, and M. Shah, "Learning affine transformations of the plane for model-based object recognition," in *Proc. 13th Int. Conf. Pattern Recognition*, vol. VI, 1996, pp. 60–64.

[25] G. Roth and M. Levine, "Geometric primitive extraction using a genetic algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 901–905, Sept. 1994.

[26] A. Katz and P. Thrift, "Generating image filters for target recognition by genetic learning," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 906–910, Sept. 1994.

[27] D. Swets and B. Punch, "Genetic algorithms for object localization in a complex scene," in *Proc. IEEE Int. Conf. Image Processing*, vol. II, 1995, pp. 595–598.

[28] G. Bebis, S. Uthiram, and M. Georgiopoulos, "Face detection and verification using genetic search," *Int. J. Artif. Intell. Tools*, vol. 9, no. 2, pp. 225–246, 2000.

[29] ——, "Genetic search for face detection and verification," in *Proc. IEEE Int. Conf. Intelligence, Information, and Systems*, 1999, pp. 360–368.

[30] M. Singh, A. Chatterjee, and S. Chaudhuri, "Matching structural shape descriptions using genetic algorithms," *Pattern Recognit.*, vol. 30, no. 9, pp. 1451–1462, 1997.

[31] N. Ansari, M. Chen, and E. Hou, "A genetic algorithm for point pattern matching," in *Dynamic, Genetic, and Chaotic Programming*, B. Soucek, Ed. New York: Wiley, 1992, pp. 353–371.

[32] J. Fitzpatrick, J. Grefenstette, and D. Gucht, "Image registration by genetic search," in *Proc. IEEE SoutheastCon Conf.*, 1984, pp. 460–464.

[33] G. Bebis, S. Louis, and S. Fadali, "Using Genetic Algorithms for 3-D Object Recognition," in *Proc. 11th Int. Conf. Computer Applications in Industry and Engineering*, Las Vegas, NV, Nov. 1998, pp. 13–16.

[34] G. Bebis, S. Louis, and Y. Varol, "Using genetic algorithms for model-based object recognition," in *Proc. 1998 Int. Conf. Imaging Science, Systems, and Technology*, Las Vegas, NV, July 1998, pp. 1–6.

[35] S. Louis, G. Bebis, S. Uthiram, and Y. Varol, "Genetic search for object recognition," in *Proceedings of the Seventh Annual Conference on Evolutionary Programming*, V. Porto, Ed. Berlin, Germany: Springer-Verlag, 1998, vol. 1447, Lecture Notes in Computer Science, pp. 199–208.

[36] D. Huttenlocher and S. Ullman, "Recognizing solid objects by alignment with an image," *Int. J. Comput. Vis.*, vol. 5, no. 2, pp. 195–212, 1990.

[37] D. Ballard, "Generalizing the hough transform to detect arbitrary patterns," *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, 1981.

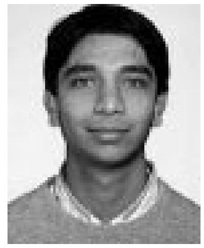
[38] F. Mokhtarian and A. Mackworth, "A theory of multi-scale, curvature-based shape representation for planar curves," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 789–805, Aug. 1992.

[39] CAL. Eshelman, "The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination," in *Proceedings of the Foundations of Genetic Algorithms Workshop -1*, G. Rawlins, Ed. San Mateo: Morgan Kaufmann, 1991, pp. 265–283.

George Bebis (S'89–M'98) received the B.S. degree in mathematics and the M.S. degree in computer science from the University of Crete, Greece, in 1987 and 1991, respectively, and the the Ph.D. degree in electrical and computer engineering from the University of Central Florida, Orlando, in 1996.

Currently, he is an Assistant Professor with the Department of Computer Science, University of Nevada, Reno (UNR), and Director of the UNR Computer Vision Laboratory. From 1996 until 1997, he was a Visiting Assistant Professor with the Department of Mathematics and Computer Science, University of Missouri, St. Louis, while from June 1998 to August 1998, he was a Summer Faculty Member with the Center for Applied Scientific Research, Lawrence Livermore National Laboratory. His research is currently funded by the National Science Foundation, the Office of Naval Research, and National Aeronautics and Space Administration. His current research interests include computer vision, image processing, pattern recognition, artificial neural networks, and genetic algorithms.

Dr. Bebis is on the Editorial Board of the *International Journal on Artificial Intelligence Tools*, has served on the program committees of various national and international conferences, and has organized and chaired several conference sessions. He is the Program and Vice Chair of the IEEE Northern Nevada Section.

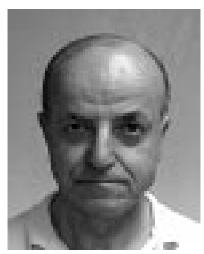


Sushil Louis received the B.Sc. degree in physics and the Master's degree in computer applications from the University of Delhi, Delhi, India, in 1983 and 1986, respectively, and the Ph.D. degree in computer science from Indiana University, Bloomington, IN, in 1993.

He is an Associate Professor of Computer Science with the University of Nevada, Reno, where he directs the genetic adaptive systems laboratory. His current research interests include evolutionary computation and machine learning.

Yaakov Varol received the B.S.E.E. degree from Robert College, Istanbul, Turkey, in 1967 and the Ph.D. degree from the University of Wyoming, Laramie, in 1971.

He is a Professor and Department Chair of Computer Science with the College of Engineering, University of Nevada, Reno. He has taught and conducted research at various universities, including Ben Gurion University, Israel, Witwatersrand University, South Africa, Southern Illinois University, Carbondale, and University of Nevada, Reno. He was also a Research Scientist for Israel Aircraft Industries, System Development Corporation, and Oak Ridge National Laboratories. His current research interests and areas of expertise include algorithm design and analysis, discrete simulation and modeling, scheduling, and network path problems.



Angelo Yfantis received the B.S. degree in mathematics from the University of Athens, Greece, in 1969, the M.S. degree in mathematics from Fairleigh Dickinson University, Rutherford, NJ, in 1972, the M.S. degree in statistics from Rutgers University, New Brunswick, in 1973, the M.S. degree in computer science from the New Jersey Institute of Technology, Newark, in 1975, and the Ph.D. degree in electrical engineering from the University of Wyoming, Laramie, in 1978.

He is a Full Professor of Computer Science with the University of Nevada, Las Vegas. He has authored or coauthored over 100 research papers in the areas of image processing, computer graphics, and computer science in general. He was a Consultant for Sandia Laboratories, Lawrence Livermore Laboratories, US Environmental Protection Agency, Department of Energy, National Aeronautics and Space Administration, US Army Corps of Engineers, Los Alamos Scientific Laboratory, Lockheed, Northrop, Statistical and Software Analysis, Inc., and Multimedia Communication Corporation.