Using Genetic Algorithms for 3D Object Recognition

George Bebis†, Sushil Louis† and Sami Fadali‡

†Department of Computer Science, University of Nevada, Reno, NV 89557
‡Department of Electrical Engineering, University of Nevada, Reno, NV 89557

Abstract

We investigate the application of genetic algorithms for recognizing 3D objects from two-dimensional intensity images, assuming orthographic projection. The recognition strategy is based on the recently proposed theory of algebraic functions of views. According to this theory, the variety of 2D views depicting a 3D object under the case of orthographic projection can be expressed as a linear combination of a small number of 2D views of the object. This suggests a powerful strategy for performing object recognition: novel 2D views of a 3D object can be recognized by simply matching them to linear combinations of known (reference) 2D views of the object. The main advantage of this strategy is that 3D models are not required. Given an unknown view of an object, the goal of the recognition procedure is to find the coefficients of the linear combination scheme. In this paper, we propose using genetic algorithms for searching the space of coefficients efficiently. Interval arithmetic is used to restrict genetic search in the most feasible regions of the coefficients' space only.

Keywords: object recognition, algebraic functions of views, genetic algorithms, interval arithmetic

1. Introduction

Object recognition has been intensely studied during the past three decades, yet even the best systems today remain capable of recognizing only simple objects under carefully controlled conditions [1]. The main difficulty arises from the fact that the appearance of a 3D object's shape varies significantly as the 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.

Typical strategies for coping with the variable appearance of objects due to viewpoint changes include the use of invariants, explicit models and multiple views. According to the first strategy, invariant properties (i.e., properties that vary little or remain invariant as viewing conditions change) are employed during recognition. The problem with this strategy is that there are no general case invariants for 3D objects. The second strategy employs explicit 3D models. During recognition, a model of the image formation process is applied to the 3D model objects in order to predict the objects' appearance and determine whether something of similar appearance can be found in the image. Approaches based on this idea are not very practical since 3D models are not always available. The last strategy models an object by a collection of views showing how the object appears from various viewpoints. Systems based on this strategy store all of these views, and recognize the object in an image when they are able to match one of the reference views to some part of the image. This strategy is not very efficient since many views must be stored for each model object.

The recently proposed theory of *algebraic functions of views* provides a powerful mathematical foundation for tackling variations in the appearance of a 3D object's shape due to viewpoint changes [2]-[4]. According to this theory, the variety of 2D views depicting a 3D object can be expressed as a combination of a small number of 2D views of the object. In the case of orthographic projection, the combination of views is linear [2] while in the case of perspective projection, a nonlinear combination must be employed [3][4]. Only the case of orthographic projection is considered in this paper. The above result suggests a powerful strategy for performing object recognition: novel 2D views of a 3D object, obtained under the assumption of orthographic projection, can be recognized by simply matching them to linear combinations of known (reference) 2D views of the object. A new approach for object recognition based on algebraic functions of views and indexing can be found in [5][6].

One difficulty with this idea arises from the fact that the coefficients of the linear combination scheme are in general unknown (see [5][6] for other difficulties). Searching the space of coefficients can be very computationally expensive. In this paper, we propose using Genetic Algorithms (GAs) [7] for searching this space efficiently. This work is an extension of our previous work on applying GAs for recognizing real, flat objects, assuming that the viewpoint is arbitrary [8]. GAs are search procedures which have been shown to perform quite well when the space to be searched is very large [7]. This makes genetic search suitable for searching the coefficients' space. Interval arithmetic [9] is used to restrict genetic search to the most feasible regions of the coefficients' space.

The paper is organized as follows: In Section 2, we review the theory of algebraic functions of views. In Section 3, we present the genetic object recognition approach. Specifically, we describe the encoding mechanism, the selection scheme, genetic operators, and fitness function used. Section 4 includes our experiments while our conclusions, limitations, and extensions of the current work are given in Section 5.

2. Algebraic Functions of Views

Algebraic functions of views were first introduced, in the case of orthographic projection by Ullman and Basri [2]. They showed that if we let an object undergo 3D rigid transformations, namely, rotations and translations in space, and we assume that the images of the 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 object. Specifically, let us consider three reference views of the same object

V_1, V_2, and V_3, which have been obtained by applying different rigid transformations, and 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 which 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 follows:

$$x = a_1 x' + a_2 x'' + a_3 x''' + a_4 \tag{1}$$

$$y = b_1 y' + b_2 y'' + b_3 y''' + b_4 \tag{2}$$

where the parameters a_j, b_j, $j = 1, \ldots, 4$, are the same for all the points which are in correspondence across the four views. The above result can be simplified if we generalize the orthographic projection by removing the orthonormality constraint associated with the rotation matrix. In this case, the object undergoes a 3D linear transformation in space and only two reference views are required. The corresponding algebraic functions are shown below:

$$x = a_1 x' + a_2 y' + a_3 x'' + a_4 \tag{3}$$

$$y = b_1 x' + b_2 y' + b_3 x'' + b_4 \tag{4}$$

where the parameters a_j, b_j, $j = 1, \ldots, 4$, are the same for all the points which are in correspondence across the three views. These results have been generalized to the case of perspective projection where the algebraic functions of views are nonlinear [3][4].

The main idea in this paper is to apply genetic search in the coefficients' space in order to determine linear combinations of reference views which might predict the appearance of an object in an unknown view. Merely letting the GA search every region of this space inefficient since the space is extremely large. Recently, we have proposed a methodology for estimating the ranges of values of the coefficients of the linear combination [5][6]. The idea is to rewrite (3) and (4) as follows:

$$\begin{bmatrix} x_1' & y_1' & x_1'' & 1 \\ x_2' & y_2' & x_2'' & 1 \\ \cdots & \cdots & \cdots & \cdots \\ x_N' & y_N' & x_N'' & 1 \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \\ a_4 & b_4 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \cdots & \cdots \\ x_N & y_N \end{bmatrix}$$

where (x_1', y_1'), (x_2', y_2'), $\cdots (x_N', y_N')$ and (x_1'', y_1''), (x_2'', y_2''), $\cdots (x_N'', y_N'')$ are the coordinates of the points of the reference views V_1 and V_2 respectively, and (x_1, y_1), (x_2, y_2), $\cdots (x_N, y_N)$ are the coordinates of the points of the novel view V. The above system of equations can be expressed in terms of two subsystems: $Pc_1 = p_x$ and $Pc_2 = p_y$. Both of these subsystems are over-determined and a least squares approach, such as Singular Value Decomposition (SVD) [10], is needed for solving them.

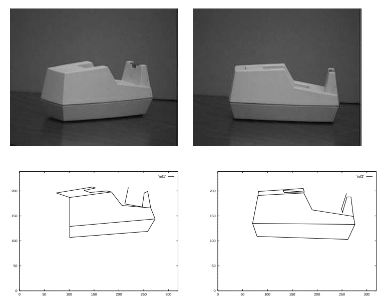

Figure 1. Two reference views of a model object.

Using SVD, we can factorize the matrix P as $P = U_P W_P V_P^T$ where both U_P and V_P are orthonormal matrices, while W_P is a rectangular matrix whose diagonal elements w_{ii}^P are always non-negative and are called the singular values of P. The solution of the above two systems is $c_1 = P^+ p_x$ 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 greater than zero (or a very small threshold in practice) and zero otherwise. Hence, the solutions of (6) and (7) are given by the following equations [10]:

$$c_1 = \sum_{i=1}^{k} \left(\frac{u_i^P p_x}{w_{ii}^P}\right) v_i^P \tag{8}$$

$$c_2 = \sum_{i=1}^{k} \left(\frac{u_i^P p_y}{w_{ii}^P}\right) v_i^P \tag{9}$$

where u_i^P denotes the i-th column of matrix U_P, v_i^P denotes the i-th column of matrix V_P and $k = 4$.

In [5][6], a methodology based on (8), (9) and Interval Arithmetic (IA) [9] was employed to determine interval solutions for c_1 and c_2, assuming p_x and p_y are restricted within some interval (for example, the input image can be scaled such that p_x and p_y belong to [0,1]). As an example, consider the 3D object shown in Figure 1. Two different reference views have been used to represent it (second row corresponds to the points extracted to represent the views). Table 1 shows the range of values computed for c_1 after the reference views have been preconditioned (i.e., an appropriate transformation has been applied on the original views in order to optimize the computed interval solutions, see [5][6]). Values for c_2 are exactly the same to those computed for c_1.

Table 1. The interval solutions.

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

3. Methodology

For testing, we used three scenes, Scene1, Scene2, and Scene3, with the object to be recognized being completely visible in Scene1 and Scene2 and partially occluded in Scene3 (see Figure 2). Our selection strategy was cross generational. Assuming a population of size N, the offspring doubles the size of the population and we select the best N individuals from the combined parent-offspring population for further processing [11]. We also linearly scale fitnesses to try to maintain a constant selection pressure.

Figure 2. The test scenes used in our experiments.

3.1. Encoding

A simple binary encoding scheme was also used to represent solutions in the space of coefficients. Each chromosome contains eight fields with each field corresponding to one of the eight coefficients of the algebraic functions of views. Since we have a methodology to estimate the minimum and maximum value of each parameter, only the range (difference between the maximum and minimum values) needs to be represented. For example, a_{11} assumes values in the interval [-0.4193 0.4193]. Thus, its range is $r = 0.4193 - (-0.4193) = 0.8386$. Assuming that up to two decimal points are important in the estimation of the affine transformation, 84 possible values ($\lceil 0.8386 x 100 \rceil + 1$) must be encoded. This means that 7 bits are enough to encode a_1's range. However, 7 bits can be used to represent values from 0 to 127 while we only need to represent values from 0 to 83. As a result, it is possible for the genetic algorithm to find solutions which are not within the desired range (i.e., [0, 83]). To deal with this problem, a simple transformation is used to map values in [0, 127] to values in [0, 81]. Two-point crossover and point mutation were used.

3.2. Fitness evaluation

We evaluate fitness of individuals by computing the back-projection error (BE) between the model and scene. After the coefficients have been obtained by decoding the chromosome corresponding to the best solution in the population, we apply the linear combination scheme (Eqs. (3) and (4)) in order to predict the appearance of the model into the scene (i.e., the model is back-projected onto the scene). Then, we compute the error, BE, between the back-projected model and the scene. This is performed by finding for every model point the closest scene point and by computing the distance dj between these two points. The sum of errors is the back-projection error:

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

Since we need to maximize fitness but minimize the error, our fitness function is

$$Fitness = 10000 - BE$$

and changes the minimization problem to a maximization problem for the GA.

4. Simulations and results

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 200 for Scene1 and to 400 for Scene2, and Scene3. Approximately 100 generations were required for each one of the scenes. For each scene, we ran each approach 10 times with different random seeds. Performance plots indicate that the GA very quickly gets close to the correct mapping and then spends most of its time making little progress. Scene1 was chosen to be exactly the same as our model, (i.e., the goal of the genetic algorithm was to find the identity mapping). Figure 3 (top) shows the best (left) and worst (right) solutions found. Results for Scene2 and Scene3 are shown in the second and third line of Figure 3 respectively.

Although there are many problems for which genetic algorithms can find a good solution in reasonable time, there are also problems for which they are inappropriate. These are mainly problems for which it is important to find the exact global optimum. It is well known that genetic algorithms do not perform well in these cases. In our problem, the best solutions found by the GA are very good. Observing the worst solutions found, we see that the GA has at least been able to perform a rough alignment of the model with the scene. Although near-exact matches might not solve the recognition problem completely, they are useful in the sense that they 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.

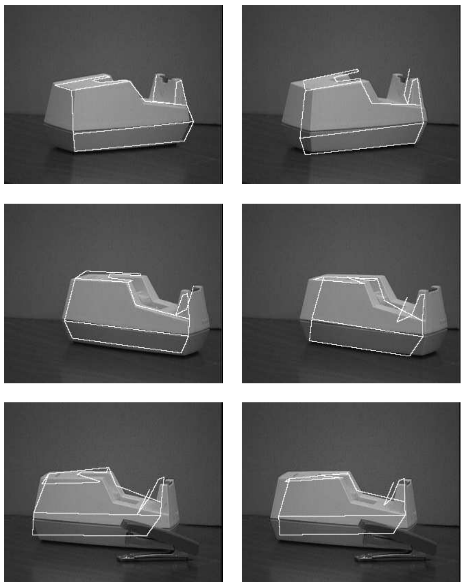

Figure 3. Best and worst solutions found.

Table 2 provides a summary of our results. The first column specifies the scene while the second column describes the size of the test problems. The number of values we use to represent a_1's range is 84 (see our discussion in Section 3.1). For a_2, we need 73 values, for a_3 we need 86 values, while in the case of a_4, we need 101 values. The values for b_1, b_2, b_3, and b_4 are the same (same interval solutions - see our discussion in Section 2). The total number of possible transformations is therefore $84^2 x 73^2 x 86^2 x 101^2 = 2,836,899,445,552,704$. The last column of Table 3, indicates the number of matches the GA approach searched through.

Table 2. Summary of results.

Results		
Scene	Number of Transforms	$GA_{matches}$
Scene1	2,836,899,445,552,704	19,600
Scene2	2,836,899,445,552,704	37,600
Scene3	2,836,899,445,552,704	37,600

5. Conclusions

In this paper, we used genetic algorithms to recognize real, 3D objects from intensity images, assuming orthographic projection. A recognition strategy based on the theory of algebraic functions of views was employed. Genetic search was applied in the space of parameters of the algebraic functions of views. Our experimental results demonstrate that genetic algorithms are a viable tool for searching this space efficiently. One limitation of our current work is that we consider only one model object in our experiments. For future research, we plan to consider more model objects as well as to extend the proposed approach to the case of perspective projection.

References

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

[2] S. Ullman and R. Basri, "Recognition by linear combination of models", *IEEE Pattern Analysis and Machine Intelligence*, vol. 13, no. 10, pp. 992-1006, October 1991.

[3] A. Shashua, "Algebraic Functions for Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 779-789, 1995.

[4] O. Faugeras and L. Robert, "What can two images tell us about a third one ?", *Third European Conference on Computer Vision (ECCV'94)*, pp. 485-492, 1994.

[5] G. Bebis, M. Georgiopoulos, M. Shah, and N. da Vitoria Lobo, "Algebraic functions of views for model-based object recognition", *6th International Conference on Computer Vision (ICCV-98)*, pp. 634-639, January 1998, India.

[6] G. Bebis and M. Georgiopoulos, M. Shah, and N. da Vitoria Lobo, "Indexing based on algebraic functions of views", *Computer Vision and Image Understanding (CVIU)*, in press (December 1998) - also available from *http://www.cs.unr.edu/~bebis*

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

[8] G. Bebis, S. Louis, and Y. Varol, "Using genetic algorithms for model-based object recognition", *Proceedings of the International Conference on Imaging Science, Systems, and Technology (CISST'98)*, pp. 1-8, 1998.

[9] R. Moore, *Interval analysis*, Prentice-Hall, 1966.

[10] W. Press et. al, *Numerical recipes in C: the art of scientific programming*, Cambridge University Press, 1990.

[11] L. Eshelman, The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination, *Proceedings of the Foundations of Genetic Algorithms Workshop - 1*, Morgan Kauffman, Gregory J. E. Rawlins (ed.), 1990, San Mateo, CA.