Model-Based Object Recognition from a Complex Binary Imagery using Genetic Algorithm

Samarjit Chakraborty[1], Sudipta De[2] and Kalyanmoy Deb[*3]

[1] Institut TIK, Eidgenössische Technische Hochschule, Zürich, Switzerland.
[2] KanGAL, Indian Institute of Technology Kanpur, India.
[3] Department of Computer Science, University of Dortmund, Germany.

Abstract. This paper describes a technique for model-based object recognition in a noisy and cluttered environment, by extending the work presented in an earlier study by the authors. In order to accurately model small irregularly shaped objects, the model and the image are represented by their binary edge maps, rather then approximating them with straight line segments. The problem is then formulated as that of finding the best describing match between a hypothesized object and the image. A special form of template matching is used to deal with the noisy environment, where the templates are generated on-line by a Genetic Algorithm. For experiments, two complex test images have been considered and the results when compared with standard techniques indicate the scope for further research in this direction.

1 Introduction

Finding the best transformation that maps an object model into the image of a scene is a central issue in object recognition. There are several approaches to this problem which explicitly rely on results from computational geometry. Among them are geometric hashing [17], alignment [14] and voting [2]. The Hough transform [22], which is recognized as a powerful tool for curve as well as object detection falls into the third category. A different line of approach involves the development of cost functions for measuring the difference between two sets of points or line segments under various transformations. Such cost functions based on the Hausdorff distance have been extensively investigated in both computational geometry [1, 3, 12] and computer vision [13, 24] literatures. Although these methods give good results in the presence of small amounts of noise and occlusion, they do not scale well when applied to complex cluttered scenes, and in the presence of a lot of noise. For example, in a study on the noise sensitivity of the generalized Hough transform by Grimson and Huttenlocher [9], it was concluded that even for moderate amounts of noise and occlusion, these methods can hypothesize many false solutions, and their effectiveness is dramatically reduced. Similar conclusions were made by Sarachik [25] for the geometric hashing

* Author to whom all correspondence should be directed.
e-mail: deb@ls11.informatik.uni-dortmund.de

paradigm applied to 2-D object recognition. So these techniques are reliable only for relatively simple tasks in the absence of excessive noise and clutter, where the image data corresponding to correct solutions is a large fraction of the total data. In an effort to address this problem, in [6] we proposed a scheme for detecting analytic curves using a Genetic Algorithm (GA) [7] in combination with the Randomized Hough transform [27]. The present paper extends that work to consider the detection of any binary object model in a binary edge map of a scene image. There exists a large volume of literature on detecting curves and objects in noisy as well as cluttered images. But most of them assume a predefined error model, either uniform bounded for feature displacement or a 2D Gaussian. Additionally, several approaches also assume the presence of the model in the image and the worst case search time in the presence of noise is exponential in the problem size [10]. The proposed method in this paper is flexible, and does not assume any error model. It is particularly effective in the case of complex images where the number of pixels belonging to the object being searched for is a very small fraction of the total number of edge pixels. In image processing literature there is a mathematical distinction between *clutter* and *noise*. The former might refer to all features or points that come from something different than the model, where as *noise* usually refers to the phenomenon in which the identified locations of the image points are slightly displaced from where they should be. Coupled with these, there might be several spurious data points in the image arising out of various sources, for example, edge points arising out of brightness discontinuities and imperfect edge detection. For the purpose of this paper it is not required to distinguish between these different errors and we will refer to all such points jointly as *false attractors*.

The concept of using GA for curve extraction has been explored in the past [11, 23]. But the problems of noise or clutter were not considered. Object recognition in a complex image using GA has also been attempted [26]. The method that we present is more flexible in terms of the allowable similarity between the model and the object in the scene image. This has important advantages for successful recognition of real life images, since it results in a flexibility in evaluating an hypothesis about the occurrence of the object in the scene. We illustrate this through examples presented in Section 4.

There has been an enormous amount of research in automatic object recognition. But despite this fact the problem remains largely unsolved. A comprehensive overview of this subject from a variety of perspectives can be found in [21]. We believe that the use of GA can help in dealing with the uncertainties that arise in any practical object recognition system. Further, since such a task involves a very large search space, a suitably designed GA approach can reduce the search time by several orders of magnitude with respect to an exhaustive search.

In this paper, the object recognition task is performed by representing the model and the image in the form of their binary edge pixels. This representation has a number of benefits. Edge pixels are robust to changes in sensing conditions and edge-based techniques can be used with many imaging modalities. Several previous approaches have considered modeling objects as a set of straight line segments, and matching these to the straight line segments extracted from an image [5, 19]. Our use of the complete edge map to model objects, rather than approximating them as straight line segments, allows irregularly shaped objects to be modeled accurately. We specifically address images with a very large fraction of points constituting the false attractors by using a special form of template matching and compare our results with standard methods. Our templates are generated on-line, guided by the GA.

In the next section we briefly identify cases where standard methods fail due to the presence of a large number of false attractors. Towards this we use an example of straight line detection, following which we describe our method. In Section 4 we describe test results with two images and compare the performance with standard methods. Section 5 concludes the paper.

2 Motivation

The various approaches towards searching for the occurrence of an object in a scene can be roughly classified depending on whether the search is performed in the *correspondence space*, *transformation space*, or both. Correspondence space is the space of *matches*, which are sets of pairings between model and image features or points. Transformation space is the space of possible object poses.

The interpretation tree approach [8] exemplifies those methods that search entirely in the correspondence space. Its name refers to a search tree of choices concerning the interpretation of each image feature. Proceeding from the root of the tree, the match search examines an additional image feature at each level of the tree. Branches at each level represent different choices among model features that can be matched to that image feature, plus the choice of matching nothing at all to it. A complete interpretation of the image, assigning some subset of image features to corresponding model features, is associated with each of the tree's leaves. This method is computationally very costly and is generally exponential to the number of image and model pixels. Hence in the presence of excessive number of false attractors, such a method is rendered infeasible.

The generalized Hough transform is an example of a method that searches the transformation space. An accumulator array indexed by parameters of object pose, is first initialized as empty. Then, for each possible match between one image feature and one model feature, poses consistent with that match are determined and votes are cast in the bins of the accumulator array corresponding to those poses. The second stage is an exhaustive search for parameters in the accumulator array which are local maxima. Each such local maximum represents a candidate match between the model and the image. In this approach, points on the same object occurring in the image result in points in the parameter space which are close together, whereas the false attractors result in randomly distributed points in the parameter space. Thus a large cluster of points in the parameter space represent a match between the model and an object in the image. The validity of this assumption, however, depends on there being a low likelihood that clusters due to false attractors will be comparable or larger in size than clusters due to points on genuine objects. We believe that in many real life images, this assumption does not hold. Fig. 1(a) shows two straight lines L_1 and L_2, where each line is composed of a small number of disconnected points. In Fig. 1(b), random noise is superimposed on the line L_1 (Fig. 4 in Section 4 shows one example where such a situation really arises in practice). Let us call the lines in Fig. 1(a) as *true lines* and the line in Fig. 1(b) that corresponds to line L_1 of Fig. 1(a), as a *pseudo line*. Line L_2 in this figure still remains a true line. If our model is a simple straight line, then ideally the recognition algorithm should detect both L_1 and L_2 from Fig. 1(a) but only L_2 from Fig. 1(b). Note that there are a large number of pseudo lines in the noise region in Fig. 1(b). Since the number of points on each of these pseudo lines is comparable or more than than the number of points on the line L_2, it gets masked in the parameter space by these pseudo lines.

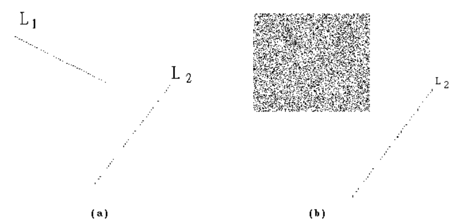

Fig. 1. A binary edge image (a) Two straight lines (b) Noise superimposed on one of the lines

3 A Genetic Algorithm for Object Recognition

To overcome the effects of noise in curve detection, the Window RHT and Random Window RHT due to Kälviäinen *et al.* [16], randomly place a window on an edge point and try to locate a curve within the window. Similarly template matching [4, 28] has been widely used in computer vision for object recognition. An object in an image is defined to be recognized if it correlates highly with a template image of the hypothesized object. The template image is usually a transformed version of the model of the hypothesized object. Our technique is conceptually similar to this. We place a weighted template on an edge point and measure the weighted difference between pixels on a real object and the spurious points surrounding it. The templates are constructed online, guided by the GA.

A crucial problem with ordinary template matching is the size of the search space [20, 18]. An attempt to overcome this is through the randomized versions like Window RHT and Random Window RHT. We feel that a search guided by a GA is more superior than a simple random search and can reduce the search time by orders of magnitude.

3.1 Generating Templates from Model Images

Given a binary edge map and a model, or possibly a library of models, our objective is to identify the occurrence of these models in the image. If a model is represented by the set of its edge pixels $\mathcal{M}$, then a template $\mathcal{T}$ is generated from $\mathcal{M}$ by choosing three parameters that describe a transformation of $\mathcal{M}$ into $\mathcal{T}$, along with some additional parameters which determine the quality of allowable matches. The parameters used for transformation are translation, rotation and scaling, and possibly also mirror image about any arbitrary line. We say that the model $\mathcal{M}$ occurs in the given image at the location indicated by the template $\mathcal{T}$ if $\sum_{x \in \mathcal{T}} Z_x \geq N_{min}$, where Z_x is the gray level of the pixel x (0 or 1 in a binary edge map) in the binary edge map of the image. The template $\mathcal{T}$ is the set of points $\{x : d(x, \mathcal{I}(x')) \leq \delta$ and $x' \in \mathcal{M}\}$, where $\mathcal{I}$ is some composition of translation, rotation and scaling, and $d(x, \mathcal{I}(x'))$ is the Euclidean distance between the points x and $\mathcal{I}(x')$. δ is a parameter which describes the width of a strip or band around the transformed model, which allows for certain tolerance. N_{min} is the minimum number of pixels of the edge detected image that must occur within the template so that the presence of the hypothesized object corresponding to the model $\mathcal{M}$ can be ascertained. A relatively large value of δ allows objects to be detected which are fuzzy or have a weak similarity with the model $\mathcal{M}$.

For images with relatively less or no spurious points such as Fig. 1(a), this formulation is sufficient and is in fact similar to the Window RHT used for curve detection, except for the fact that we do not use any transformation mapping from the image to the parameter space as is common in Hough transform. Rather, we simply count the number of points lying within the template $\mathcal{T}$. But in the case of images with a large proportion of false attractors such as Fig. 1(b), whenever the template is placed on a region consisting of such points, a false alarm in the form of a pseudo object will be raised. To extend this method to include such images, we formulate a weighted template rather than the simple one described above. The response of the template $\mathcal{T}$ under this formulation is given by $R = \sum_{x \in \mathcal{T}} W_x Z_x$, where W_x is the weight or coefficient of the pixel x. We shall say that the model $\mathcal{M}$ occurs in the image at the location indicated by template $\mathcal{T}$ if the response R of the template is greater than a constant R_{min}, fixed, depending on the dimensions of the model, template width δ, and the coefficients W_x. The coefficients of pixels that lie away from the transformed model i.e. $\mathcal{I}(\mathcal{M})$, are assigned negative values. So when a lot of spurious points are present in the neighborhood of $\mathcal{I}(\mathcal{M})$, as in the case of the pseudo lines in Fig. 1(b), the positive response due to the points on and near $\mathcal{I}(\mathcal{M})$ is offset by the negative response due to the spurious points surrounding it. As a result, false alarms are avoided.

3.2 Parameter Search Space and the use of GA

Even if a particular object is known to be present in the image a priori, the space of transformations from the model to the image is extremely large. Hence an exhaustive search of this space would take too long to find a good match between templates and images. A random search in the presence of excessive noise and clutter is also not beneficial. So instead of randomly choosing the transformation parameters to generate a template, we use a genetic algorithm to search the parameter space for all instances of objects for which the template response is greater than R_{min}. For this, each of the parameters - x and y-coordinates of the translation vector, the rotation angle, and the scaling factor, are coded as fixed length binary strings. The resulting string, obtained by concatenating all these strings, gives the chromosomal representation of a solution to the problem. Note that the domains of each of the parameters may be different and the length of the string coding a given parameter depends on the required parameter resolution. The fitness of a solution is taken to be the response of the weighted template, as described in the previous section.

Since in practical situations an exact match between the model and a hypothesized object is not expected, we construct the template $\mathcal{T}$ such that points near the transformed model $\mathcal{I}(\mathcal{M})$ are associated with positive coefficients and points lying further away have negative coefficients. This, along with a suitably chosen value of the minimum response R_{min}, offers considerable flexibility regarding the quality of the resemblance between the model and the detected objects.

Creation of initial population. In most GA applications, the initial population consists of entirely random structures to avoid convergence to a local optima. However, in this problem, the question is not of finding the global optima, but of finding all solutions with fitness greater than R_{min}. To identify prospective regions of the search space, the *hypothesize and test* paradigm commonly used in visual object recognition might be effectively used. In [6] we used a Randomized Hough transform for this purpose. For object recognition, a variation of this method similar to the generalized Hough transform might be used to generate an initial set of hypotheses. Towards this, pairs of points are randomly chosen and possible transformations which map these two points onto points in the image are computed, as in the alignment method [14]. However instead of explicitly testing such transformations, the *count*, in the accumulators representing the parameter space, corresponding to such transformations are incremented by one. After repeating this process for a predefined number of times, points in the parameter space with counts exceeding a predefined threshold represent candidate hypotheses. The GA searches the entire parameter space with a bias towards these hypotheses. Corresponding to each candidate hypothesis, a suitable number of solutions are introduced into the initial population. Further, a fixed number of random samples from the solution space are also introduced. The total number of solutions is kept fixed over all the generations.

It should be noted that the above mentioned method of generating candidate hypotheses is rendered ineffective in the presence of excessive clutter and extreme scaling, where this scheme is no better than randomly generating the initial population. However, for images with even moderate amounts of noise and clutter, it can lead to a considerable speedup.

Selection. The selection used here falls into the category of dynamic, generational, preservative, elitist selection [27]. Let there be M distinct solutions in a given generation, denoted by $S_1, S_2, \ldots, S_M$. The probability of selecting a solution S_i into the mating pool is given by :

$$P(S_i) \;=\; \frac{\mathcal{F}(S_i)}{\sum_{j=1}^{M} \mathcal{F}(S_i)}$$

Where $\mathcal{F}(S_i)$ is the fitness of the solution S_i. A fixed number of solutions are copied into the mating pool according to this rule and a small number of remaining solutions are randomly generated. In each generation, a fixed number of best solutions of the previous generation are copied in place of the present worst solutions, if they happen to be less fit compared to the former. This is a slight modification of the Elitist model where only the best solution is preserved.

Crossover and Mutation. Because of the number of parameters involved, it is intuitive that the single point crossover operation may not be useful. So crossover is applied to each substring corresponding to each of the parameters - x and y-coordinates of the translation vector, the rotation angle, and the scaling factor, the operation being the usual swapping of all bits from a randomly chosen crossover site of the two parents, chosen randomly from the mating pool [29]. Hence this crossover is similar to the standard single-point crossover operator, but operated on substrings of each parameter. Therefore, there are four single-point crossovers taking place between two parent strings.

We have used a classical mutation operator in which each bit position of the solution strings is complemented with a small mutation probability.

The overall algorithm. The initial population consisting of a fixed number of solutions is created as already described. In each generation, the entire population is subjected to selection, crossover and mutation. At the end of each generation, edge pixels corresponding to solutions having fitness greater than R_{min} are removed from the edge map. After fixed number of generations, the accumulators corresponding to the parameter space used for generating the candidate hypotheses are reset and the voting process is repeated to generate a fresh set of hypotheses. Candidate solutions corresponding to these are then introduced into the population and whole process is once again repeated. This iteration is continued until no new curve segments are extracted for a given number of generations, which in our experiments was set to 200.

4 Test Results and Comparisons

We have experimented with two different images. For the ease of comparison with standard methods, in our first experiment the model is a simple straight line. Although this is the simplest possible case, as evident from the previous sections, our algorithm is blind to this fact. For comparing the performance of our method with Hough transform which is the most popular method for straight line detection, we used a public domain software package XHoughtool

[15], where a number of non-probabilistic and probabilistic Hough transform algorithms have been implemented.

As indicated in the previous section, there are various parameters that our algorithm uses. The parameters related to the template are its width, the coefficients or weights associated with each pixel, and the threshold response R_{min}. The allowable quality or degree of correspondence between the model and the objects extracted from the image is determined by the template coefficients and its width. A wide template with more than one row of positive coefficients will detect objects whose pixels are spread out along its width compared to the model in question. Thus, a suitably designed template, along with a proper threshold value R_{min}, will be able to distinguish between an object having a relatively weak similarity with the model, and a false attractor. In our first experiment where the model is a straight line, we have used a template width of 3, to detect only perfect straight lines. The coefficients of all pixels lying on the straight line were set to 2 and the others to -1 as shown in Fig. 2. Too low a value, R_{min}, of the threshold might detect a pseudo line where as a too high value might miss a faint, disconnected, but visually detectable line. The results shown in this section were obtained with R_{min} set to the length of the transformed line, i.e. $\mathcal{I}(\mathcal{M})$.

-1	**-1**	**-1**	**.......**	**-1**	**-1**	**-1**
2	**2**	**2**	**.......**	**2**	**2**	**2**
-1	**-1**	**-1**	**.......**	**-1**	**-1**	**-1**

Fig. 2. A mask of width 3

For the GA parameters, we used a mutation probability of 0.1 and any population size around 100 was found to work well. In each generation, 25% of the solutions were randomly created and the rest copied from the mating pool in accordance with the fitness proportionate selection. Further, the best 10% solutions of the previous generation were copied in place of the worst solutions of the current generation.

Fig. 3(b) shows a 512 by 512 binary image obtained after edge detection of the corresponding gray scale image shown in Fig. 3(a). Our model in this case consists of a simple straight line. Note the three disconnected, but visible real lines in the image, two at the center and one the the extreme left end. The straight lines detected by our algorithm are shown in Fig. 3(c). Altogether seven different Hough transform algorithms are implemented in the XHoughtool package. In spite of a serious attempt being made to select the test parameters for each method as optimally as possible, none of the algorithms gave useful results because a large number of pseudo lines were detected. A typical result is shown in Fig. 3(d). Since the number of edge points lying on the real lines are much less compared to those lying on many of the pseudo lines, no suitable accumulator threshold value exists which can detect only the real lines. Generally these algorithms work well even in the case of noisy images, where the lines are connected and the number of edge points lying on these lines are at least comparable to the number of noise points. It is to be noted in this example that there are edge detectors which if used for Fig. 3(a) along with proper thresholding, would eliminate most of the false attractors now appearing in Fig. 3(b). In such a situation, a simple HT algorithm would suffice. However, we have used the edge detection algorithm incorporated in *xv*, the interactive image viewer available on any X-window system. This, in some way artificial route, was adopted only to illustrate a situation where the proposed method might be useful. Secondly, the 'lines' detected in Fig. 3(c) are actually edges, with no width. But for the purpose of this algorithm we do not distinguish between an edge and a line.

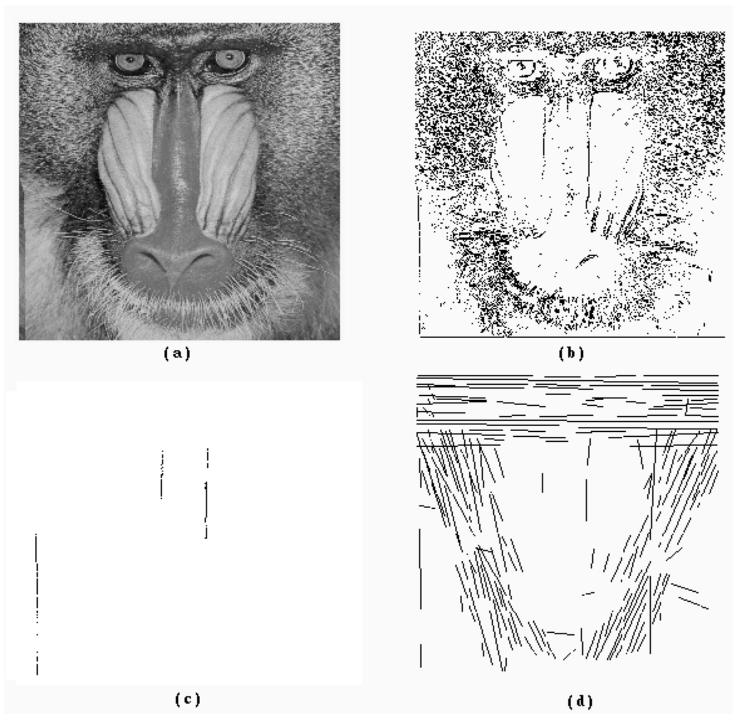

Fig. 3. Test results with the model being a simple straight line (a) A 512 by 512 gray scale image (b) The corresponding binary edge map obtained after edge detection (c) Straight lines detected by the proposed method (d) A typical result obtained using a Hough transform algorithm

Our second example is a synthetic image shown in Fig. 4 consisting of instances of the letters **A** and **Z** under various orientations and scaling, along with random noise. The model is a letter **A**. Note that there are seven instances of the letter **A** in the scene image. In most of the test runs the algorithm could detect all the seven **A**s and avoid any false alarms. However, it was crucial to approximately choose the value of the threshold R_{min}. In this case also we used a template width of three as in the previous example.

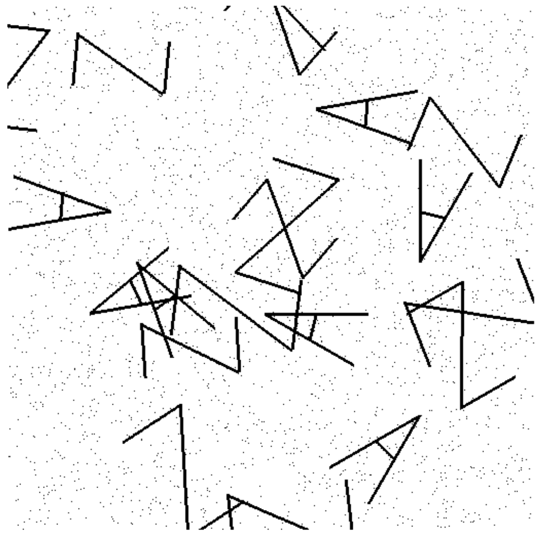

Fig. 4. A binary scene image where the model **A** is to be detected

We should emphasize here that the procedure for generating the initial solutions described in the last section, is much more effective in the case of our second example where the proportion of false attractors is much less compared to the first.

5 Summary

The Hough transform and its variants are the most popular methods for detecting analytic curves from binary edge data. However, they do not scale well when applied to complex environments in the presence of excessive noise and clutter. In [6] we presented a GA in combination with the Randomized Hough transform but using a different scoring function, to deal with such environments. This paper extended that technique to incorporate model based object recognition. Towards this we used a special form of template matching which offers a considerable flexibility regarding the quality of the allowable matches. Although there has been attempts to use simple random search for several computer vision problems, a search guided by a GA is probably superior in this case.

For future work, further experimentation could be performed using a variety of different image and model pairs to illustrate the general applicability of this method. One possible application domain might be automatic target recognition where, because of its military applications, the goal is to avoid the object being detected. It should also be possible to utilize other methods than the one described here for generating the set of hypotheses used for initializing the population. Further, it would be interesting to extend the set of transformations considered here with shearing for example, to test weak similarities between the model and the image.

Acknowledgements

The authors are grateful to the anonymous referees for their constructive criticism and suggestions.

References

1. P.K. Agrawal, M. Sharir, and S.Toledo. Applications of parametric searching in geometric optimization. In *Proc. of 3rd. ACM SIAM Symp. on Discrete Algorithms*, pages 72–82, 1992.
2. T. Akutsu, H. Tamaki, and T. Tokuyama. Distribution of distances and triangles in a point set and algorithms for computing the largest common point sets. In *Proc. 13th. Annual ACM Symp. on Computational Geometry*, pages 314–323, Centre Universitaire Méditerranéen, Nice, France, 1997.
3. H. Alt, B. Behrends, and J. Blömer. Measuring the resemblance of polygonal shapes. In *Proc. of 7th. Annual ACM Symposium on Computational Geometry*, pages 186–193, 1991.
4. Dana Ballard and Christopher M. Brown. *Computer Vision*. Prenctice Hall, 1982.
5. Ross J. Beveridge. *Local Search Algorithms for Geometric Object Recognition: Optimal Correspondence and Pose*. PhD thesis, University of Massachusetts, Amherst, May, 1993.
6. S. Chakraborty and K. Deb. Analytic curve detection from a noisy binary egde map using genetic algorithm. In *Proc. 5th. International Conference on Parallel Problem Solving from Nature (PPSN V)*, pages 129–138, 1998. Lecture Notes in Computer Science 1498.
7. D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, M.A., 1989.
8. W. E. L. Grimson. *Object Recognition by Computer : The Role of Geometric Constraints*. MIT Press, 1990.
9. W. E. L. Grimson and D. P. Huttenlocher. On the sensitivity of the Hough transform for object recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-12:255–274, 1990.
10. W. Eric L. Grimson. The effect of indexing on the complexity of object recognition. Technical Report A.I. Memo No. 1226, Artificial Intelligence Laboratory, MIT, 1990.
11. A. Hill and C. J. Taylor. Model-based image interpretation using genetic algorithms. *Image and Vision Computing*, 10:295–300, 1992.
12. D.P. Huttenlocher, K. Kedem, and M. Sharir. The upper envelope of voronoi surfaces and its applications. *Discrete and Computational Geometry*, 9:267–291, 1993.

13. D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. Pat. Anal. and Mach. Intel.*, 15:850–863, 1993.
14. D.P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *Inter. Journal of Computer Vision*, 5(2):195–212, 1990.
15. H. Kälviäinen, P. Hirvonen, L. Xu, and E. Oja. Houghtool–a software package for Hough transform calculation. In *Proceedings of the 9th Scandinavian Conference on Image Analysis*, pages 841–844, June 1995. (http://www.lut.fi/dep/tite/XHoughtool/xhoughtool.html).
16. H. Kälviäinen, L. Xu, and E. Oja. Recent versions of the Hough transform and the Randomized Hough transform : Overview and comparisons. Technical Report 37, Department of Information Technology, Lappeenranta University of Technology, Finland, 1993.
17. Y. Lamdan and H.J. Wolfson. Geometric Hashing: A general and efficient model-based recognition scheme. In *International Conference on Computer Vision*, pages 238–249, 1988.
18. Z. Q. Liu and Terry M. Caelli. Multiobjective pattern recognition and detection in noisy backgrounds using a hierarchical approach. *Computer Vision, Graphics, and Image Processing*, 44:296–306, 1988.
19. Chris Loader. Local search algorithms for 2d geometric object recognition. Master's thesis, Department of Computer Science, The University of Western Australia, 1995.
20. Avrahm Mergalit and Azriel Rosenfeld. Using probabilistic domain knowledge to reduce the expected computational cost of template matching. *Computer Vision, Graphics, and Image Processing*, 51:219–234, 1990.
21. Arthur R. Pope. Model-based object recognition : A survey of recent research. Technical Report TR-94-04, Department of Computer Science, University of British Columbia, January, 1994.
22. J. Princen, J. Illingworth, and J. Kittler. A formal definition of the Hough transform : properties and relationships. *J. Math. Imaging Vision*, 1:153–168, 1992.
23. G. Roth and M. D. Levine. Geometric primitive extraction using a genetic algorithm. *IEEE Trans. Pattern Anal. Machine Intell*, PAMI-16(9):901–905, 1994.
24. W.J. Rucklidge. Locating objects using the Hausdorff distance. In *Proc. of 5th. International Conference on Computer Vision*, pages 457–464, 1995.
25. Karen B. Sarachik. Limitations of geometric Hashing in the presence of gaussian noise. Technical Report A.I. Memo No. 1395, Artificial Intelligence Laboratory, MIT, 1992.
26. D. L. Swets, B. Punch, and W. John. Genetic algorithms for object recognition in a complex scene. In *Proceedings of the International Conference on Image Processing*, pages 595–598, October, 1995. Washington, D.C.
27. L. Xu and E. Oja. Randomized Hough transform (RHT) : Basic mechanisms, algorithms, and computational complexities. *CVGIP : Image Understanding*, 57(2):131–154, 1993.
28. Leonid P. Yaroslavsky. *Digital Picture Processing*. Springer Verlag Berlin, 1985.
29. K. S. Y. Yuen, L. T. S. Lam, and D. N. K. Leung. Connective Hough transform. *Image and Vision Computing*, 11(5), 1993.

This article was processed using the LaTeX macro package with LLNCS style