# Computationally efficient Hough transform for 2-D object location 

E. R. Davies<br>Machine Vision Group, Department of Physics<br>Royal Holloway, University of London<br>Egham Hill, Egham, Surrey, TW20 0EX, UK


#### Abstract

This paper discusses the use of Hough transforms for feature-based object location. Though the GHT is highly robust and operates in polynomial $\left(\mathrm{O}\left(n^{2}\right)\right)$ time, in real-time applications further reductions in computation are often required. A feature-orientated HT has been designed which offers further speedup by factors of $\sim 3$, though robustness is also reduced. However, if objects are not immediately recognised, the new scheme can be made to revert gracefully to the normal GHT. Loss in robustness is most severe when objects exhibit a high degree of internal symmetry, but for objects with seven or more features arranged nonsymmetrically the new scheme should be especially valuable.


## 1 Introduction

Objects are commonly located in digital images by one of two main approaches. The first involves the use of edge detectors which produce boundary representations of the objects, which can then be located by boundary tracking followed by use of $(r, \theta)$ or ( $s, \kappa$ ) plots, or by techniques such as the generalised Hough transform (GHT) [1]. The second involves the use of salient features - typically corners or small holes - from which the presence of the objects is inferred by graph-matching procedures such as the maximal clique technique [1-3]. In either case, in many applications it is vital that the method adopted should be robust against noise, partial occlusions, and object defects or other artefacts, and Hough transforms (HTs) and maximal cliques are powerful enough to achieve this within the respective approaches [1].

In this paper we shall be concerned with the second approach. Though (as indicated above) this conventionally uses graph-matching techniques, these often suffer from a combinatorial explosion in computational requirements, and computation may even be in the same class of 'NP-complete' algorithms as the travelling salesman problem. For this reason there is much interest in the application of HTs for this purpose, and it has been demonstrated that computation is polynomial rather than exponential in the number of salient features on an object, both in the time required to build parameter space and in the time required to analyse it for the peaks representing objects [1,4,5].

This paper relates to real-time applications such as robot assembly and automated visual inspection, and is much concerned with the time required to locate objects in images. Accordingly, the paper describes a new variant on the HT approach which offers the possibility of reducing computation still further. First we consider how the GHT is applied to the location of objects from their salient features. Then we examine the new variant on the HT, and study its advantages and limitations.

## 2 The generalised Hough transform approach

In the GHT a localisation point $L$ is defined on the idealised object shape, and each pair of image features found in the image is examined to determine whether it corresponds to a pair of idealised object features (typically, this requires the pair of
features to be the same distance apart in the image and object model). Every such match is used to estimate the position of L in parameter space, and votes are accumulated at all such positions. As in the normal GHT (i.e. that used for locating objects from their boundaries), peaks in parameter space are interpreted as likely positions for the localisation points of the objects being sought. The computation is $\mathrm{O}\left(M^{2} n^{2}\right)$ for building parameter space, and to $\mathrm{O}\left(M^{2} n^{2}\right)$ or better for finding peaks in parameter space, where $M$ features have been found in the image and objects possess $n$ features. Thus speedup factors of at least an order of magnitude are to be expected when using the GHT instead of the maximal clique technique in typical applications [6].

## 3 A feature-orientated Hough transform

The basic idea for improving the speed of the HT technique was to cut out the computationally heavy calculations of the position of the object localisation point L that are normally made once it has been confirmed by reference to the object model that a pair of image features could be part of an object. Thus instead of voting for objects directly we vote for the specific features of an object, accumulating evidence that they belong to objects. Clearly, the maximum number of votes obtained by any one feature will then be $n-1$, if objects possess $n$ features, and $n_{\mathrm{v}}-1$ if only $n_{\mathrm{v}}$ of these features are visible (unoccluded) in the image. Object distortions and noise will also affect the number of votes obtained by any feature. In addition, certain noise features may receive some votes and these will have to be distinguished from real object features.

We have found that a good strategy for proceeding with the analysis is to consider first those features with the largest numbers of votes, and to progress systematically to other features compatible with these which also have a large number of votes. In this way the features on individual objects are grouped together, starting with the most easily distinguishable objects, and then they and their features can be eliminated from further consideration: this makes it easier to discern further objects, which will in some cases be less visible, e.g. having greater degrees of occlusion or distortion. In the end relatively few features remain; if these are widely separated and have little support, they will be interpreted as 'artefacts' or noise, though of course they may be objects which are so occluded that their presence is hardly worth conjecturing (the GHT approach described in the previous section will, however, give a full formal hypothesis of their existence).

The above discussion ignores the exact placement of the features on the objects, but in fact we have found that this has to be analysed more or less concurrently with the grouping of features. (In fact, we first find a tentative grouping, then classify the features within it, then firm up on the grouping to identify an object completely, and repeat the process until all the objects that can be identified have been identified.)

The process of feature classification is described in detail in the next section: it involves analysing the set of distances to other features within the same object and comparing them with an ideal set deduced from the object model. Although in principle this permits features to be identified uniquely, noise and other occurrences (such as internal object symmetries) affect the interpretation, as we shall see below. However, the general principle is that there will normally be sufficient redundancy in the experimental data for features to be identified uniquely without error.

An interesting logistic of the method is that the same total number of pairs of features are considered as for the GHT approach of section 2. Thus the total number of votes cast in parameter space is still $2\binom{n}{2}=n(n-1) \approx n^{2}$ for each object, though the maximum number appearing at any one peak is reduced from half this total number to $n-1$. As we shall see, this accounts for a certain reduction in robustness of the method. However, as stated earlier, computation is cut down significantly because the calculation associated with placing each vote is much less heavy, votes being placed at the raw feature point positions rather than at some distant point $L$ whose position has
to be computed using trigonometric formulae [6].

## 4 Details of the new approach

The basis of the new approach is to vote for each feature individually, accumulating evidence not only that it is on an object of the chosen type but also that it is at a particular position on that object. We start by assuming that the objects we are locating are completely general, so that the distances between pairs of features are all different. We retain a lookup table of all these distances. Then, when a new feature is located in an image, its distance from all the other features in the image will be compared with those in the table, and votes will be placed against the possible interpretations of the new feature. When this process has been completed for all the features in the image, the votes for each feature will be examined to determine the optimum classification for each feature.

Because each inter-feature distance will occur twice, once for each of the features at its ends, there will be some votes for erroneous classifications. We illustrate this for a general 5-feature object in Table 1. Evidently, the number of votes for each feature in a general $n$-feature object is $n-1$ or unity, under the $n$ different classifications. Clearly, the erroneous votes give no real ultimate disadvantage, since we can regard the set of votes for a given image feature as forming a codeword with high redundancy and there will be considerable distance between codewords (the minimum distance between codewords is $2(n-2)$ ). Thus this is a forward error correcting code which is highly robust against errors such as those arising from occlusion, noise, object distortion and other artefacts. However, it is relevant that the minimum distance between codewords increases linearly with number of object features, so the objects that are found most reliably by this method will be those which are large and complex.

The main problem with this concept is the assumption that objects are totally general. If any inter-feature distance happens to be repeated in a different part of the object template, then it will give rise to ambiguities in that further erroneous votes will arise and the protection provided by the code will be eroded. Table 2 shows the result of this, when $\mathrm{AB}=\mathrm{DE}$.

It is worth exploring the possibility of a different voting scheme which might offset this disadvantage. The simplest such scheme would be to decide that the interfeature distance $\mathrm{AB}(=\mathrm{DE})$ gives very little discriminatory information, and to ignore it. In that case we get the revised voting scores shown in Table 3. A second approach would be to acknowledge that less discriminatory information arises from this interfeature distance, and to allot it a vote of low but non-zero value, e.g. 0.5. In fact, we treat such schemes with caution, since the information provided by the nondiscriminatory votes is useful in showing that a given feature is well-supported as being on an object. The reason for this is to distinguish more readily those noise features that happen to be valid distances from certain object features. Thus it would be necessary to include the total number of votes of Table 2 as an extra column in Table 3.

Table 1
observed feature
A
B
C
D
E
$\begin{array}{llllll}\text { B } & 1 & 4 & 1 & 1 & 1 \\ \text { C } & 1 & 1 & 4 & 1 & 1 \\ \text { D } & 1 & 1 & 1 & 4 & 1 \\ \text { E } & 1 & 1 & 1 & 1 & 4\end{array}$

Table 2


Our experimental work has not given particular support for such alternative schemes, and we have maintained ${ }^{*}$ the scheme of Table 2, but have enhanced it in a different way, by a scheme of penalty points. For each observed feature, we generate a list ${ }^{\dagger}$ of the inter-feature distances it gives rise to. Then we examine to what extent the list is a sublist of the various ideal lists of inter-feature distances (sublists could arise because of the effects of partial occlusions). A penalty point is awarded for each additional inter-feature distance (which might be due to extra noise or clutter features) which prevents the list of inter-feature distances from being a sublist of the ideal list. Thus we get a table of penalty points which is much like the table of votes, but in each row the minimum value must be sought. This is seen from the penalty point table (Table 4) which corresponds to the first example (Table 1) above.

To summarise, our scheme gives two tables of interpretation values for each observed feature. One is a set of votes, in which the column of maximum voting value must be selected; the other is a set of penalty points, from which the column giving the minimum value must be selected. In the ideal case of completely general objects in ideal conditions, the votes determine the class of each image feature, and no ambiguity arises; the same occurs if penalty points are used for classification. In a case where partial object occlusion occurs, the same unambiguous situation arises, and no problems of interpretation can occur, though redundancy is reduced, and the minimum distance between codewords falls from $n-1$ to $n_{v}-1$, where $n_{v}$ features are visible on a given object. The main problems arise when objects are not general, and when noise or clutter features appear in the image. Note that object distortions can be more serious, as they can be considered as eliminating some features and replacing them with noise features, i.e. we are effectively getting occlusions and noise features at the same time.

Let us now consider the effect of adding a noise feature. First, note that if the noise feature is not compatible with the remaining features (i.e. has no permitted interfeature distances in common with them) then there is no problem. Second, a noise feature will in general have fewer total votes than a normal object feature, and can therefore be recognised as noise and eliminated; straightforward elimination may also be possible if the feature's pattern of votes does not correspond to any class. If this is not possible immediately, then we must proceed to analyse all the features as best we can, and then show that the noise feature does not fit into the scheme of things arising with the remaining features. But what is the effect of the noise feature on the remaining features? If there is no occlusion of the object, then adding a noise feature will add extra votes to the existing features, and may well disrupt interpretation. However, it will also add points to the columns of the penalty point table, and subtraction of these penalty points from the corresponding votes will tend to restore proper interpretation. This procedure will not necessarily lead to an accurate result if there is some occlusion, as the feature point may in some ways simulate a valid feature, and penalty point subtraction will then fail to provide correction, but in general it will be valuable. In our experiments we have found that considering the votes alone, or the penalty points alone, does not give the best results: the best results are obtained by using the table of votes minus penalty points for feature classification. Here 'best' means obtaining the least ambiguity or error in the final classification.

We now proceed to consider cases where objects possess a high degree of

[^0]Table 3 observed feature A B

| B | 1 | 1 | 1 | 1 | 1 |
| :--- | :--- | :--- | :--- | :--- | :--- |
| C | 1 | 1 | 1 | 3 | 1 |
| D | 1 | 1 | 1 | 0 | 3 |

Table 4 observed feature
A
B
C
D
E

Table 5 observed feature A
B
C
D
E

Table 6 observed feature A
B
C
D
$\begin{array}{llllll}\mathrm{E} & 2 & 2 & 2 & 2 & 4\end{array}$

Table 7
observed inter-feature distance
feature

| $\alpha$ |  | $2 a$ | $d$ |
| :--- | :--- | :--- | :--- |
|  | $a$ |  |  |
| $\beta$ | $d$ | $2 b$ | $b$ |
| $\gamma$ | $a$ | $b$ | - |,

Table 8

| observed | votes |
| :---: | :---: |
| feature | $\alpha \quad \beta$ |
| $\alpha$ | 42 |
| $\beta$ | 24 |
| $\gamma$ | 2 |

Table 9

| observed | penalty point |  |  |
| :---: | :---: | :---: | :---: |
| feature | $\alpha$ | $\beta$ |  |
| $\alpha$ | 0 | 2 | 3 |
| $\beta$ | 2 | 0 | 3 |
|  |  | 3 | 0 |

Table 10 observed difference values

| feature |  | $\alpha$ |  | $\beta$ |
| :--- | :--- | ---: | ---: | ---: |
| $\alpha$ |  | $\gamma$ | 0 | -2 |
| $\beta$ |  | 0 | 4 | -2 |
| $\gamma$ |  | -1 | -1 | 4 |

Table 11 observed

| feature | Q ${ }^{2}$ | $\gamma$ |
| :---: | :---: | :---: |
| A | 3 | 1 |
| B | 2 | 1 |
| C | 3 | 1 |
| E | 21 | 3 |

b


Figure 1 Object with features arranged in a diamond shape
This figure shows five point-features arranged in the shape of a diamond. This arrangement is characteristic of some real objects, e.g. custard cream biscuits, which have the outline shown in grey. (a) shows the normal labelling, and (b) shows the symmetryreduced labelling for the object. The respective sets of inter-feature distances are given in Tables 5 and 7.
symmetry, so that they have a good many repeated inter-feature distances. Consider the case of a diamond-shaped object with an additional feature at its centre (Figure 1). Its inter-feature distances are given in Table 5, and the resulting votes for the ideal unoccluded object are given in Table 6. The symmetry actually makes this table unnecessarily complex, the reason being that it is meaningless to try to distinguish between A and C or B and D. Hence we adopt the procedure (also used with the maximal clique approach) of relabelling the object template using the sets $\alpha=\{\mathrm{A}, \mathrm{C}\}$, $\beta=\{B, D\}$, and $\gamma=\{E\}$. Then we get the simplified inter-feature distance table (Table 7) and table of votes (Table 8). Compared with the original table for a general 5 -feature object, we see that there is reduced redundancy in the codewords and reduced protection against occlusion or noise. We now proceed to examine the penalty points. These are shown in Table 9, and lead to a significantly more reliable table of votes minus penalty points (Table 10).

There are two points to be noted about such tables: first, they are not in general symmetric (if noise features are ignored, the penalty point table is only guaranteed to be symmetric in the absence of occlusion - see below); and it is not necessarily the case that the vote table entries and the penalty point table entries sum to $n-1$. This situation is perhaps more obvious when real image features are to be classified, as the features will then be labelled individually, while the classes will have the set labels. To demonstrate these ideas, we next examine this last example in the case that feature D is occluded. The problem now reduces to the sets of votes, penalty points and differences shown in Tables 11-13. Notice how the differences give the most protection against noise. Note also that the sum of votes and penalty points is not in all instances equal to $n_{v}-1$ : where this is not so, the difference table shows improved rather than worse discrimination between classes - i.e. if anything we genuinely do better if we use the difference table.

We have applied this technique to a significantly more complex problem, in which biscuits with 9 symmetrically placed holes have to be located for inspection. Here the holes are arranged in the form of a regular hexagon which has been slightly compressed, and three additional holes have been added symmetrically at its centre (Figure 2). The inter-feature distances are given in Table 14, together with a symmetry-reduced labelling, as in the above example. Note in addition that $j=a$, a fact which could not be deduced from the symmetry of the situation. In this case the votes, penalty points and differences are given in Tables 15-17.

As in the case of Table 13, there is a genuine gain in discriminability to be obtained from use of the difference table: in this case it has also been found practically that the difference table gives the best protection from errors. Figure 3 shows an image that has been interpreted in this way. While little trouble occurred with the top two objects, in spite in the second instance of two missing features and additional noise points, the third highly occluded object was more problematic. Successful interpretation required application of the following simple algorithmic procedures: (a) elimination of noise points having less than half the number of links with other features than the maximum for the group, (b) application of the difference table technique, (c) resolution of ambiguities to make individual features (such as the second feature down in the third object) fit into the scheme of things laid down by existing firm feature interpretations. However, it was not necessary to use iterative (relaxationtype) implementation to resolve conflicts.

It is important to note that the method was pressed quite hard by this example. The reason for this is the lack of redundancy in the feature codewords, caused by the ambiguities arising from the high degree of object symmetry. Effectively, it is straightforward to write an algorithm that "knows there is an object present", but less straightforward to obtain a complete, correct interpretation, i.e. placement of the template on the image. Though the method was here near the limits of its capabilities, it demonstrated that in relevant cases it can produce a considerable saving in

Table 12
observed penalty points
feature
A
B
C
$\begin{array}{llll}\mathrm{E} & 2 & 2 & 2 \\ & 2\end{array}$

Table 13
observed difference values
feature
A
B
$\begin{array}{llll}\mathrm{C} & 3 & -1 & -1 \\ \mathrm{E} & 0 & -1 & 3\end{array}$

Table 14 Inter-feature separation distances for the object model
In this table, (a) shows the inter-feature separation distances for the object model with the original labelling (Figure 2(a)), and (b) shows these distances with the symmetry-reduced labelling (Figure 2(b)): $\alpha=\{A, C, D, F\}, \beta=\{B, E\}, \gamma=\{G, I\}, \delta=\{H\}$.

Note also that $c=2 a, h=2 e, j=2 i$, and $j=a$ (see text).
a observed inter-feature distance

| feature | A | B | C | D | E | F | G | H | I |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
|  | - | $a$ | $b$ | $c$ | $d$ | $e$ | $f$ | $a$ | $g$ |
| B | $a$ | - | $a$ | $d$ | $h$ | $d$ | $k$ | $e$ | $k$ |
| C | $b$ | $a$ | - | $e$ | $d$ | $c$ | $g$ | $a$ | $f$ |
| D | $c$ | $d$ | $e$ | - | $a$ | $b$ | $g$ | $a$ | $f$ |
| E | $d$ | $h$ | $d$ | $a$ | - | $a$ | $k$ | $e$ | $k$ |
| F | $e$ | $d$ | $c$ | $b$ | $a$ | - | $f$ | $a$ | $g$ |
| G | $f$ | $k$ | $g$ | $g$ | $k$ | $f$ | $\bar{c}$ | $i$ | $j$ |
| H | $a$ | $e$ | $a$ | $a$ | $e$ | $a$ | $i$ | - | $i$ |
| I | $g$ | $k$ | $f$ | $f$ | $k$ | $g$ | $j$ | $i$ | - |


| observed | inter-feature distance |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| feature |  | $\beta$ |  | $\delta$ |
| $\alpha$ | $b c e$ | ad | fg | $a$ |
| $\beta$ | ad | $h$ | $k$ | $e$ |
| $\gamma$ | $f g$ | $k$ | $j$ | $i$ |
| $\delta$ | $a$ | $e$ | $i$ | - |

Table 15

| observed | votes |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| feature |  |  | $\gamma$ | $\delta$ |
| $\alpha$ | 8 | 4 | 4 | 3 |
| $\beta$ | 5 | 8 | 4 | 3 |
| $\gamma$ | 5 | 3 | 8 | 2 |
| $\delta$ | 6 | 6 | 6 | 8 |

Table 16

| observed | penalty points |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| feature | $\alpha$ | $\beta$ |  | $\delta$ |
| $\alpha$ | 0 | 4 | 5 | 5 |
| $\beta$ | 4 | 0 | 7 | 5 |
| $\gamma$ | 5 | 7 | 0 | 6 |
| $\delta$ | 5 | 5 | 6 | 0 |

Table 17

| observed | difference values |
| :---: | :---: |
| $\underline{\text { feature }}$ | $\alpha \quad \beta \quad \gamma \quad \delta$ |
| $\alpha$ | $8 \quad 0$ |
| $\beta$ | $1 \begin{array}{llll}1 & 8 & -3 & -2\end{array}$ |
| $\gamma$ | $\begin{array}{llll}0 & -4 & 8 & -4\end{array}$ |
| $\delta$ | 110 |



Figure 2 Object model of a more complex, symmetrical shape
This figure shows nine point-features arranged symmetrically in the form of a regular hexagon (plus three additional features) which has been slightly compressed. (a) shows the normal labelling, and (b) shows the symmetry-reduced labelling for the object. The respective sets of inter-feature distances are given in Table 14.


Figure 3 Image with point-features located
This figure shows a noisy, $256 \times 256$ pixel, grey-scale image of three biscuits in which a number of point-features have been located by a hole detector. Note the noise-induced features, the features that have not been located, and the partial occlusion of the lowest object. A computationally efficient algorithm is required which will identify, locate and place an idealised template on all the objects, ready for inspection or robot placement. The object model for each biscuit is shown in Figure 2.
computation, relative to the usual HT approach - which itself gives a very considerable saving over the maximal clique technique. Times for solving this problem using the maximal clique, standard HT, and the present feature classification approach were respectively $67.33,3.30$ and 1.05 seconds, the computation being carried out on a DEC VaxStation 2000 running VMS with algorithms coded in Pascal. Thus the new approach showed a more than 3-fold speed advantage over the standard HT.

## 5 Discussion

The fact that the votes are made at the raw feature point positions is rather interesting, and distinguishes the method from the original Hough approach. In particular, there is now no need to vote in an image-like parameter space, but instead we can vote against a list of feature points in a completely abstract parameter space. This saves computation, for there is no need to pre-clear a whole image space - or to search a whole image space for HT peaks. The search is only through a list of $M$ datapoints. Overall, the main problem with the method is the fact that the maximum number of votes in each peak has been markedly reduced, thereby giving less protection against occlusion, noise and object distortions. However, it is easy to institute a failsafe approach to object identification and placement: if an object has been identified but not unambiguously placed (which is the way the method actually fails in practice), then its set of feature points can be processed by a normal GHT (cf. the case of the minimal match graph technique [7]). If this procedure is invoked sufficiently rarely, then little additional computation will be necessary, and overall image analysis will be performed accurately and with considerably improved efficiency. Thus the main problem with the method occurs with highly symmetrical objects which must be analysed fully in all cases in order to be sure which is which of the feature points at many highly similar locations within an object. Essentially, short-cut methods of object identification tend to exhibit equivocal behaviour when objects have a high degree of internal symmetry in that case close scrutiny is always needed for rigorous interpretation.

The problem of object distortion has been referred to several times in this paper, with the implication of a threshold tolerance on the inter-feature distance measure. If the threshold is made too small, fewer valid feature-pairs are found and robustness suffers. Similarly, if the threshold is made too large, many false-alarm pairs are located and robustness again suffers. In general there is a non-critical optimal threshold: a more detailed analysis of this effect will be made in a future publication.

The method described here should perhaps be compared with other methods for shape representation and analysis which examine pairwise distance constraints between features. The Grimson and Lozano-Pérez approach [8] is similar to ours in taking basic image features which have been located satisfactorily, but then it uses a tree search procedure for image analysis: this is in close parallel with the GHT approach, and indeed both methods reduce the size of the search space by appropriate pruning of the input data. This does not prevent the approach in [8] from being described as a wholeobject detection scheme.

The chord distribution method of Taylor and Cooper [9] searches images for object boundaries that are disguised by noise. In principle, these boundaries are continuous, and a chord length probability distribution is used to iteratively enhance boundaries taking account of prior knowledge of the shape. Although the method gradually focuses on and enhances specific boundary points, it again appears best to describe the method as searching for the object as a whole, and in this respect it has greater similarities to the GHT and to probabilistic relaxation schemes than to our new approach.

Finally, the geometric histogram technique used by Evans et al. [10] has some similarities to our method. On the other hand, it uses angle and distance features for line segments rather than distances between point features, and analysis is based on
continuous boundaries rather than discrete features; thus it is in the end difficult to make a useful detailed comparison of the two methods.

## 6 Conclusion

This paper has discussed the use of Hough-based approaches to the location of objects from their salient features. It takes as its starting point the fact [1,4,5] that the GHT is better suited to the task than graph-matching methods such as the maximal clique technique, in that it normally requires considerably less computation - though both approaches are robust with regard to occlusions, noise and other artefacts. However, in real-time applications such as automated visual inspection, time is at a premium, and further reductions in computation and improvements in speed are often required. A feature-orientated HT has been designed which offers speedup factors $\sim 3$, though robustness is reduced by predictable amounts. If in a practical situation an object is not being recognised, a two-stage process can be invoked so that the new scheme gracefully reverts to the normal GHT.

Loss in robustness is most severe when objects exhibit a high degree of internal symmetry, and, as a rough guide, the new scheme should be especially valuable for objects with seven or more features arranged in a non-symmetrical pattern.

## Acknowledgements

The author is grateful to the ACME Dircctorate of the UK Science and Engineering Research Council for financial support during the course of this research. Figure 3 is reproduced from [6] by kind permission of Elsevier Science Publishers B.V. (Amsterdam).

## References

1. Davics, E.R. (1990) Machine Vision: Theory, Algorithms, Practicalities, Academic Press, London
2. Ambler, A.P., Barrow, H.G., Brown, C.M., Burstall, R.M. and Popplestone, R.J. (1975) "A versatile system for computer-controlled assembly", Artif. Intell., 6, pp. 129-156
3. Bolles, R.C. and Cain, R.A. (1982) "Recognizing and locating partially visible objects: the local-feature-focus method", Int. J. Robot. Res., 1, pp. 57-82
4. Davies, E.R. (1988) "An alternative to graph matching for locating objects from their salient features", Proc. 4th Alvey Vision Conf., Manchester (31 August - 2 Sept.), pp. 281-286
5. Davies, E.R. (1991) "Alternative to abstract graph matching for locating objects from their salient features", Image Vision Comput., 9, no. 4, pp. 252-261
6. Davies, E.R. (1992) "Locating objects from their point features using an optimised Hough-like accumulation technique", Pattern Recogn. Lett., 13, no. 2, pp. 113-121
7. Davies, E.R. (1991) "The minimal match graph and its use to speed identification of maximal cliques", Signal Process., 22, no. 3, pp. 329-343
8. Grimson, W.E.L. and Lozano-Pérez, T. (1987) "Localizing overlapping parts by searching the interpretation tree", IEEE Trans. Pattern Anal. Mach. Intell., 9, no. 4, pp. 469-482
9. Taylor, C.J. and Cooper, D.H. (1990) "Shape verification using belief updating", Proc. British Machine Vision Assoc. Conf., Oxford (24-27 September), pp. 61-66
10. Evans, A.C., Thacker, N.A. and Mayhew, J.E.W. (1992) "Pairwise representations of shape", Proc. 11th IAPR Int. Conf. on Pattern Recogn., The Hague (30 Aug. - 3 Sept.), Vol. I, pp. 133-136

[^0]:    * There is always the danger, when introducing an ad hoc improvement to a scheme, that it will have side-effects which make the resulting scheme worse than the original in some respects. Hence it was thought best in this initial work to clarify the particular advantages and limitations of the basic scheme: better understanding should increase the chance of obtaining substantial future improvements.
    ${ }^{\dagger}$ Here the term 'list' is used to denote a set of inter-feature distances augmented by the number of occurrences of each distance.

