Comparison of Approaches to Feature Detection

R.W. Series, C.J. Radford, M.J. Varga, P. Fretwell, A.C. Sleigh.

RSRE, St Andrews Rd, Malvern, Worcs. WR14 3PS. UK.

This paper contrasts three different object identification approaches applied to feature detection. The first approach uses the conventional Hough transform, while the other two, namely the Edge List Search (ELS) and Full Image Search (FIS) algorithms are dynamic programming based techniques.

The Hough transform approach accumulates evidence from the whole image in support a particular feature or shape, which in this implementation is restricted to circles. In contrast the FIS and ELS algorithms are based on local search techniques and are able to search for more complex shapes. The principal difference between the FIS and ELS method is that the ELS is based on a strongly edge based philosophy, while the FIS algorithm retains the concept of an edge only as a line which defines the shape of the sought feature.

Feature detection is a key component of most image processing schemes and yet, despite its importance, it is still the Achillies heel of many systems.

Selection of an optimum feature detector for some target application requires a compromise to be sought between several incompatible factors. The detector should be sensitive and capable of detecting the feature in the presence of noise, and yet it should minimise the number of false alarms. Equally a tolerance to the distortions brought about by changes in illumination, view angle and camera artefacts is desirable to minimise the number of models which must be retained in the feature library.

While there is a considerable literature on the design of optimal feature detectors of one form or another, a system may only be considered optimal within the constraints of the formulation of the problem. One serious problem in many areas of image processing is that the major sources of noise are far from random and coherent noise in the form of camera artefacts, geometrical distortions, obscuration and illumination. These many and varied effects may only be tested by critical examination of the performance of the feature detectors on actual images.

A further problem in determining the overall performance of a system concerns the manner in which errors propagate. It is often found that gross failure of a system may be attributed to poor performance at a low level and that the severity of an error as its consequences propagate upwards increases.

In this paper we compare the performance of three feature detection schemes. The domain of interest is detection of wheels (e.g. car wheels) in urban and semi-urban settings. The three methods selected adopt very different styles of tackling the problem.

EXPERIMENTAL

Hough Transform

The first approach is based on the use of a Hough transform and is described more fully in [1].

A line-edge description of the scene is first extracted using an operator based upon the Sobel edge convolution. The output of the horizontal and vertical Sobel convolutions yields both magnitude and direction of edge gradient. This information enables a non- maximum gradient suppression operation to thin the edge image to a single pixel wide edge map, and also allows a simple heuristic which limits the maximum turn between neighbouring pixels to be performed to suppress noise. Thinning is then followed by a hysteresis threshold similar to that employed by Canny [2] to produce a skeletal edge map.

With many semi-urban scenes it is found that the edge map contains an excessive line detail in regions of high texture, such as produced by foliage, and other vegetation. A discrimination between man-made and natural edge features is made using a simple measure of the fractal dimension of line segments. The algorithm determines the total change in line orientation within each eight-connected line segment. This is normalised to give a line roughness factor which is used as a measure of the fractal dimension of the line segment and is thresholded to remove all 'rough' lines.

AVC 1989 doi:10.5244/C.3.25

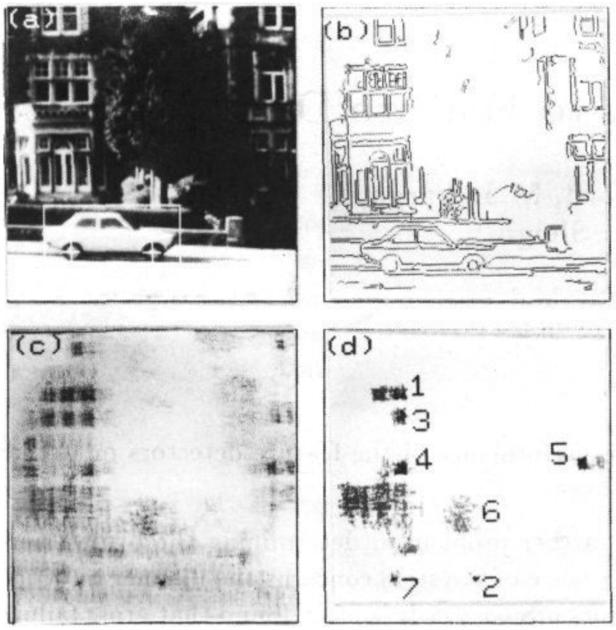

Figure 1: *Performance of Hough transform method on a typical image. (a) Input image. (b) Edge map after fractal discriminant. (c) Hough space (d) extracted peaks labelled in order of intensity. Only peaks 6, 2 and 7 comply with the peak shape requirement.*

Wheel like features are found from the retained line edge image using a version of the circle Hough transform. The normal circle Hough transform maps to a three dimensional parameter space, but with knowledge of edgel direction a two dimensional transform formed by projecting normals to the edgels in the image will yield peaks at the centres of all arcs and circles in the image. An advantage of this technique is that all concentric arcs and circles will contribute to the same peak, allowing in this instance wheel arches and wheel arc segments to contribute to a peak at the nominal wheel centre. Partial evidences for the existence of a feature are therefore accumulated into global evidence for that feature.

Since many shapes such as rectangles may also create peaks in the transform space it was found necessary to make measurements of the peak profiles, accepting only 'sharp' peaks as attributable to centres of circles and arcs.

Figure 1 demonstrates the application of the method to a typical image. Fig 1(a) shows the original image and (b) the edge extracted image after application of the fractal discriminant. (c) shows the Hough transform output and in (d) the peaks in Hough space have been labelled in order of intensity. Peaks 1,3,4 and 5 are rejected as being too diffuse. Only peaks 2, 6, and 7 satisfy the peak shape criterion, and in a rather complex image only one false centre is located. The method is relatively fast and is suited to parallel systems.

Edge List Search

Edge List Search (ELS) is an object identification approach to feature extraction that uses dynamic programming. It is more general than the Hough transform approach because it can extract shapes that are complex. It is used here to identify an object, e.g. wheels embedded in the large number of edges which generally result from edge extraction applied to a complex natural scene.

Dynamic programming can be used as an optimal technique of non-linear matching and is readily applicable to one-dimensional data. The principle of its use in pattern processing is that it is used to obtain a match between observed data and reference templates. Recognition or identification is made on the basis of finding the template that best matches the observed data on a global basis. This approach has been used with great success in one-dimensional applications such as speech recognition and digital communications, however, it is not easily extensible to two-dimensional data. The difference between the dynamic programming approach used here, [3], and that normally used, is that here, the process is reference or template data driven, rather than the more normal situation where it is observed data driven. Additional improvements to the efficiency and storage requirements of the algorithm have been made through the use of segment linkage records, rather than a full back linkage matrix; and the "pre-generation" of allowed connectivities between observed segments, in order to reduce the search space in a way analogous to the use of a grammar. Both these techniques have been used in speech recognition but have not been applied in image processing.

In this system a reference model is created consisting of a sequence of segments; outlines of circles of different sizes are used as the reference segments corresponding to large and small wheels. The observation data is generated as follows. As in the Hough transform approach the first step is to produce an "edgel" image, this is a raster organised edge map containing edge direction and intensity information. The raster organised edge map is then decomposed into a list format. This comprises two stages. First the node points (points with more than two edge point neighbours) are deleted in order to break the edge map into isolated lines or closed loops. Second a list format data file is then created based on an 8-adjacent neighbour connectivity. Closed shapes are arbitrarily broken. The resulting segments are therefore edges, each segment being made up of a list of connected pixel data, each element of which has associated with it information about its horizontal and vertical positions and its orientation. Each segment is also recorded in reverse order so as to allow matching to be performed in either direction. The segments are unique with no cross junction so as to eliminate ambiguities.

Long unbroken segments are used for the reference templates, whereas the large number of segments extracted from the natural images are short and discontinuous. In general there will be a large number of edges extracted from the whole image, only a few of which will be associated with the object to be identified, e.g. a car wheel. The system is designed to find any set or sequence of observed segments that could, when associated together, correspond to a complete reference segment. The technique is reference data driven in that the search is carried out by looking for a sub-set of the observed segments to match the whole of the reference rather than the other way round.

A one pass dynamic programming technique is used. It gives the optimal path in terms of cumulative cost, where the cost is as a function of the connectivity, shape and orientation. In order to speed up the program a prunning cost threshold is set on the accumulated cost so as to "kill" bad paths at the earliest possible instance. At the end of the forward pass the traceback identifies the segment sequence with the lowest cost. The algorithm can provide a complete set of graded sub-optimal paths which can be used as alternative choices, i.e. multiple object hypothesis can be extracted. In this application it means that more than one circular object can be identified if it exits.

Since the observed segments are not connected the dynamic programming match has to be built up on the basis of the possible connectivity between the short, disconnected, observed segments as well as their shape and orientation. Here connectivity is based on the Euclidean distance between observed segments. A fairly low cost penalty is imposed on connecting ends of segments together (for instance, a higher penalty is imposed on the missing pixels within a segment). However, a list of permitted connectivities between segments is pre-generated, based on a thresholded Euclidean distance; this list is used to constrain the dynamic programming search in a way analogous to a syntax. Other preprocessing carried out includes the elimination of all one or two pixel segments from the search.

Figure 2, 3 and 4 demonstrate some results obtained using the ELS algorithm. In 3 the second and third match traverse the same segments in the image, but in different directions. When the scale between reference and image feature is well matched (as in figure 4) the algorithm performs well and finds the three wheels present in the starting edge map as it's first three paths. However, as with all methods based on a chain code representation in orientation space, the method can make matches to features which, when compared in Euclidean space

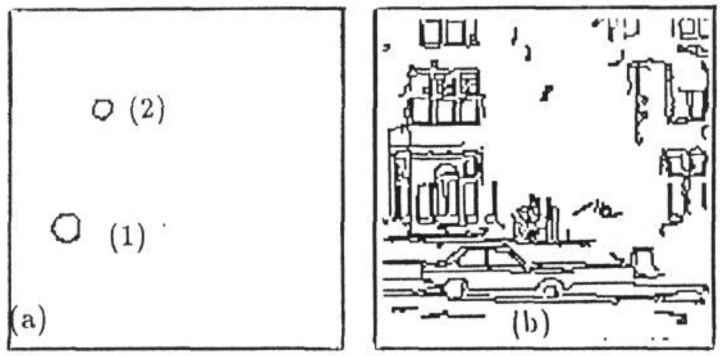

Figure 2: *Reference shapes and test image for ELS algorithm*

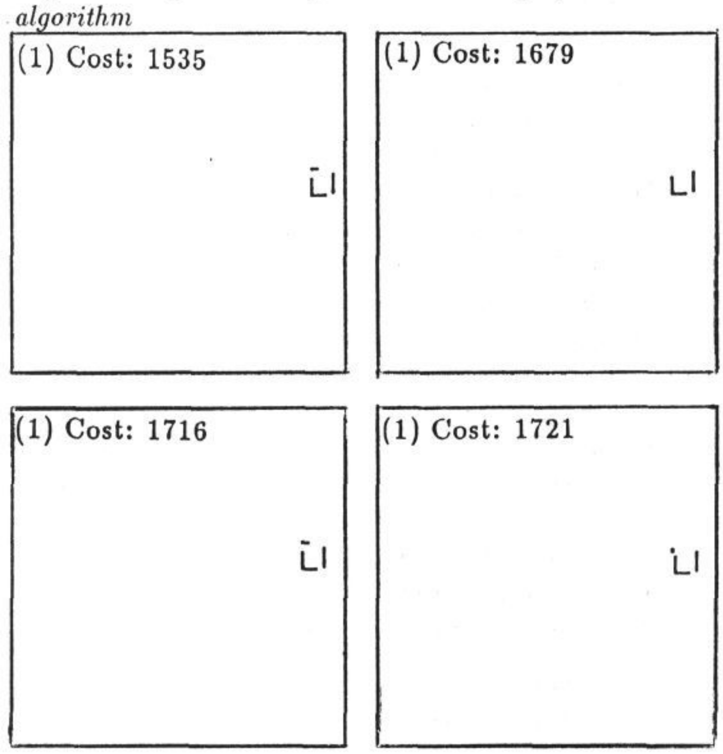

Figure 3: *Best 4 matches against large wheel reference using ELS algorithm. (Note that low cost denotes better match.)*

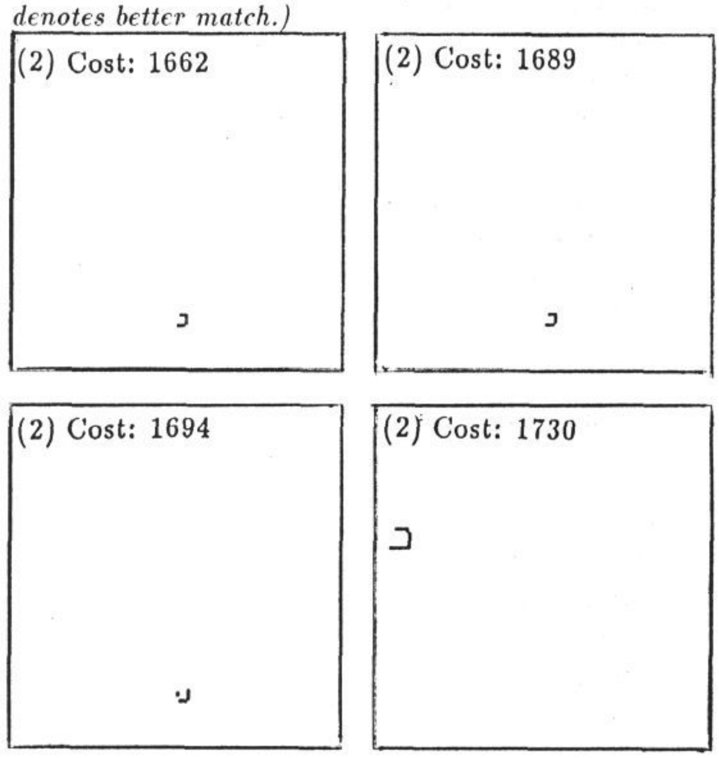

Figure 4: *Best 4 matches against small wheel reference using ELS algorithm.*

have a very poor fit. As with the HT method this may be treated by an additional stage of post-processing, or possibly more elegantly by a better choice of cost function during the dynamic programming stage.

Full Image Search

The full image search (FIS) algorithm is an attempt to avoid some of the precommitment inherent in the use of edge based approaches. The method is best described by comparison to the ELS algorithm.

In the ELS algorithm the possible paths which the search may take are constrained to fall only on the pixel points which have been designated as 'edge' by the the edge finding stage, and may only depart from following a sequence of 'edge' points which have nearest neighbour adjacency at the discretion of the list forming algorithm, an algorithm which has no domain specific knowledge. In contrast the FIS algorithm lifts this restriction and the path may move over any pixel locations in the image, limited only by the choice of productions which embodies domain specific knowledge derived from the reference shape and choice of production costs. In effect all pixels are considered to have some degree of 'edginess', and the algorithm searches for the path which maximises the total 'edginess' consistent with the shape constraints imposed by the reference.

The quality of the match between reference model and image path is composed of the sum of two sets of terms. The first relate to the distortions suffered by the path in the image and are refered to as the production penalties. The second set of terms measure the local cost of the association between a point in the reference and that in the image. We compute the local cost as the dot product of the intensity gradient in image and reference.

Figure 5 demonstrates the scheme of productions chosen in more detail. As an example the tessallation path taken by a four stage reference shape is shown. At stage 3 the reference path moves in a North East direction. The corresponding production in the paths being tested in the image is unpenalised as it corresponds to no shape change between reference and image paths. To permit some elasticity in the match, and to enable the algorithm to deal with noise (and to a certain extent obscuration) four other productions are allowed as shown. These productions correspond to Euclidean shape distortions between reference and image paths and are penalised. Depending on the path taken by the reference at each stage the choice of productions and penalties are changed. The five chosen productions give approximately the same degree of flexibility as is present with the three productions employed in the ELS algorithm.

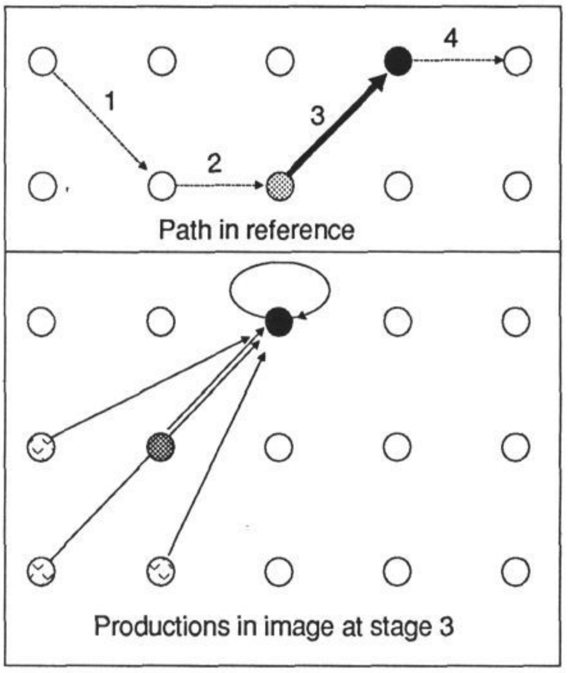

Figure 5: *Relationship between productions used in FIS and the tessellation pattern of the reference. At stage 3, productions from the pixel marked ● lead to no shape change and are unpenalised, while productions from the pixel positions marked ◯ and ● lead to a distortion of the path and are penalised.*

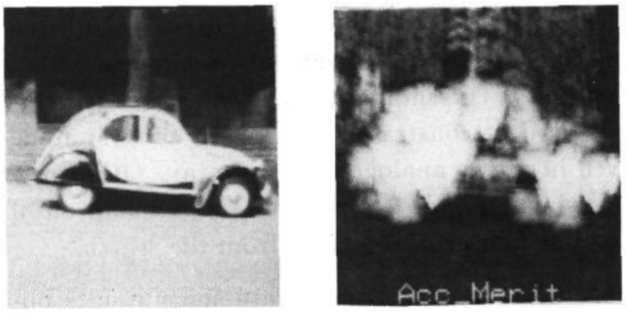

Figure 6: *Original image and the corresponding accumulated merit image (black denotes low merit) for the FIS method when searching for a wheel shaped feature.*

Figure 7: *Accumulated merit image with a selection of paths superimposed.*

Figure 6 demonstrates the method as applied to a search for a wheel in a car. As may be seen the accumulated merit is low in regions of uniform texture in the image. In figure 7 a selection of the paths have been traced back. It may be seen that in regions of low intensity gradient in the image (e.g. the road) the paths are mainly determined by the tessellation pattern of the reference. Under these circumstances the accumulated merit then represents the integral of the local cost function along the line. This approximates to zero. In contrast, in regions of high image detail the paths become distorted to flow along the high gradient regions in the image. This may be seen in the paths near the bonnett and, though not visible in the figure, in most of the paths over the car. In these paths the accumulated merit is now a combination of the integral of the local cost function and the path distortion penalties.

In comparing the underlying mathematics of the ELS and FIS algorithms, it should be noted that in the former we are maximising line integrals whose path is constrained by the edge finder output, while in the FIS algorithm we compute the line integral along lines which are less heavily constrained. We would therefore expect that, other things being equal, the FIS algorithm should yield better results.

DISCUSSION

Each method has it's own strengths and weaknesses.

The Hough transform approach requires the image to be reasonably free of extraneous detail for successful operation, hence the need for the fractal discriminant. The discriminant brings with it a number of problems since it groups edge segments with no use of high level knowledge. Hence a contorted line may abut to a smooth line and the discriminant will then threshold both smooth and contorted edge together. (To a certain extent it is possible to minimise this problem by selection of the intensity hysteresis thresholds but this makes the algorithm more sensitive to image illumination and exposure.) A particular strength of the method is that wheel features rarely give rise to a single isolated circle, but rather a set of concentric arcs. The Hough transform combines the evidence from all these features into one point and is insensitive to fragmentation of the image. In this respect the Hough transform may be considered as a form of integral over an area. The most serious disadvantage of the method is that it is very specific to circular features and performance for more generalised shapes, though possible in theory, tends to fail in practice.

In contrast, the ELS algorithm is capable of searching for a wide variety of shapes and may be tailored to accept a range of distortions. An important additional factor is that the reference shape and cost functions may be determined from appropriate training data. One of the major weaknesses of the ELS method is its sensitivity to erroneous edge linkage at the list forming stage. Whilst the algorithm may detect a reference shape embedded in an image edge list, or detect a reference shape from a concatenation of image edge lists, the algorithm often fails if there is erroneous edge linkage within the image lists. To minimise these problems it is generally desirable to operate the algorithm with an asymmetry between the degree of segmentation of image and reference shape. If the image is over segmented an under segmented reference may be used to find the best concatenation of edge lists in the image. In the extreme the image may be treated as a collection of isolated edge elements and the algorithm may then be used to find the correct concatenation. This is however achieved at the expense of greater computational burden. At the other extreme the reference may be under segmented and the algorithm may then be used to search for a larger number of sub-features suitable for processing by a later operation.

The third algorithm, FIS, takes to the extreme the concept of over segmentation of the image. No thresholds are set and any pixel in the image may play a role in locating the feature of interest. In this manner the pre-commitment which is inherent in the edge extraction is avoided and all the domain specific knowledge becomes incorporated into the feature locating algorithm.

In all three methods, location of a single point in the image to represent where the desired feature is most likely to be found poses little problem. However, it is often required to find more than one candidate position. In the HT method use of a clustering al-

Figure 8: *Accumulated merit plotted at the mid-path position for all paths in the image. Same data as for figure 6.*

gorithm may be used to find any desired number of peaks in transform space. With the two DP methods however the representation is more complex and standard clustering algorithms, if applied to the accumulated cost maps, would not make use of the information available in the back pointer matrices. If the full paths are traced back, then it is generally found that the second best path differs from the best only by one pixel. While this is formally the correct answer, it is often of more benefit to know where the next best cluster of good paths begins. In the ELS algorithm we approach this problem by looking for paths which differ in the sequence of segments which they include. This approach has been adopted in preparation of the results presented in figure 3 and 4. It may be seen that although one or two paths fall around each wheel, the best few paths embrace most of the wheel like objects in the picture.

In the FIS algorithm the concept of edge segments is abandoned and a different technique must be tried to find the clusters of interest. One approach which may be followed is to note that strong features in the image act as valleys into which the paths are drawn. Thus, while the end points of the paths are show a diffuse profile, the mid-points are strongly clustered. We demonstrate this in figure 8. Here all paths have been traced back to their mid-points and the merit of the best path is noted at the location of the mid-path. Unlike the accumulated cost matrix, not all locations are filled. Indeed, as figure 8 demonstrates the structure is very sparse with a few intense peaks which are often localised at a single pixel location. This provides a 'focussing' action and a very tightly clustered image is seen. The degree of clustering is dependent on the production penalties employed and the image content. If the production penalties are so large as to prevent any distortion of the paths, then the mid-path array will appear exactly as the accumulated cost matrix and no focusing will be seen. However under these conditions the algorithm behaves as a linear method and is unlikely to be of interest.

One problem which cuts across many pattern recognition algorithms is the generation of appropriate reference models and the setting of thresholds in the system. Here the dynamic programming algorithms at the core of the ELS and FIS algorithm have an important advantage over the Hough transform approach because, in addition to being able to deal with complex reference shapes it is possible to use the Viterbi or Baum-Welch algorithm to refine initial estimates of the cost function and so improve detection efficiency.

References

[1] **Radford, C.J.** Vehicle Detection in Open-World Scenes using a Hough transform Technique *IEE Proc 3rd Int. Conf. on Image Processing and Applications*, Warwick, U.K. July 1989.

[2] **Canny, J.** A Computational Approach to Edge Detection. *IEEE Trans. P.A.M.I.* Vol 8, (1986) pp 679-696.

[3] **Sleigh, A.C.** Segmentation and Concatenation of Edgel Lists by Dynamic Programming and Stochastic Models. *Alvey Vision Conference* 1988.

[4] **Montanari, U.** On the Optimal Detection of Curves in Noisy Pictures. *Comm ACM* Vol 14. May 1971.