SEGMENTATION AND CONCATENATION OF EDGEL LISTS BY DYNAMIC PROGRAMMING AND STOCHASTIC MODELS

Andrew C. Sleigh

Royal Signals and Radar Establishment

This paper describes a shape recognition method which uses a reference template to edit linked lists of edgels by concatenation and re-segmentation to give an optimal match. Dynamic programming is used to find the optimal re-segmentation, and is tolerant to distortion, outlines embedded in longer lists, gaps and missing portions. Unlike previous work, the combinatorial advantage of using edge linking prior to matching is retained, but poor decisions made by the linking process can be revoked during the dynamic programming stage without significantly degrading optimality. The paper discusses the extension to scale and rotation invariance, and the use of stochastic models in place of reference templates for generic feature extraction and distortion modelling.

Suppose we aim to recognise objects in images by finding arrangements of characteristic shape features. These features might be primitive generalised shape features (eg arcs, lumps, geometric shapes) or shapes specific to an object (eg a roof profile). Ideally, we might proceed by first extracting arbitrary line features from the image using only local constraints, divide the lines extracted into perceptually significant segments, and then recognise the segments as characteristic features by some form of matching process. This philosophy is behind Marr's thesis for vision, and has figured in many image understanding systems where object specific constraints are not employed at this level.

The pursuit of this philosophy has led to a wide range of published work on edge extraction techniques and on methods for segmenting outlines into component elements, typically by examining local orientation derivatives, eg *Brady & Asada*[1].

Despite this extensive literature, success in using line segments to match against object level models has been limited to domains where sparse edges are clearly illuminated, surfaces are free from texture, and the objects are in a benign background. Even in these favourable conditions, most workers have found it necessary to use motion or stereo to aid the object matching process. In open world situations, such as those encountered by autonomous vehicles, monitoring or security systems, and military applications, the situation is far worse. The amount of data requires some measure of local feature analysis, but the effectiveness of edge extraction, linking and segmentation is so variable, and fragmentation so severe, that a data driven (bottom-up) approach to generalised shape extraction is not credible.

It is important to note that many edge extraction "defects" are caused by fundamental image ambiguities which can never be resolved by purely local means, eg lines may meet in ambiguous junctions, an object may have been viewed through a mesh, or may have fortuitous alignments with background features.

Various methods have been used to impose higher level more global constraints, for example the Hough transform[2], or the model based methods of Sullivan[3] or Bolles[4]. In open world scenes all these methods imply large amounts of computation unless some form of preliminary cueing or attention focussing is used.

What is needed is a technique which can allow high level constraints to be applied to the feature extraction processes, but which avoids the combinatorial explosion implied by such constraints. Several mathematical tools exist which have the right behaviour, the most promising candidates being simulated annealing and dynamic programming[5] (DP). Simulated annealing has generated extensive interest recently in the context of neural computing and associative memories, but

AVC 1988 doi:10.5244/C.2.44

special hardware will be needed before this approach can be used in real time applications. Dynamic programming is only applicable to sequential formulations, and uses local computation to find global optima provided certain constraints are satisfied, but where it can be used, it gives a dramatic reduction in problem dimensionality. Dynamic programming has been used with great success in speech recognition[6], but has received little attention in machine vision, perhaps because of the apparently inappropriate restriction to sequential domians.

Examples of the use of DP in image analysis are given in *Ohta & Kanade*[7] where DP is used to match intensity profiles between raster lines in stereo TV images; *Furst & Caines*[8] where edge locations are fitted to raw image data; *Fischler & Elschlager*[9] who consider several applications of DP to matching pixel data and high level relationships; whilst *Maitre & Wu*[10] use a pixel space distance metric to to match coast-line contours against map data. Much of the published work use DP to match individual pixels, rather than pre-segmented data. This has an attraction, and in principle DP will work without any explicit ordering of the data. However, with degraded data the sequential constraint of DP tends to cause desired matches to be rejected part way through the analysis in favour of spurious alignments, and unnecessarily heavy computation cost is incurred. Additionally, the concentration on matching in pixel coordinate space impedes shape generalisation. The use of DP to edit linked edge data, or match in other metric spaces, seems to have been overlooked. The vision literature also lacks reference to the extension of DP by the use of stochastic reference models or re-estimation techniques, and there is no discussion of explicitly modelling the allowed distortions in a particular shape.

The aim of the current research is to re-visit the application of dynamic programming to machine vision, with particular interest in the concatenation and re-segmentation of connected lists of edgels. Given that connected segments of useful length can be successfully extracted, it seems unreasonable to ignore this connectivity in formulating the DP optimisation, but instead use DP to amend already formed edgel lists. As this paper shows, this approach opens a new avenue for model based matching in image data.

BASIC MATHEMATICS

Assume we have a low-level image operator which converts an image into a thinned edgel map, and that the edgel map is then linked into a set of line segments constrained by a local property (eg orientation, strength, fuzziness). These *edgel lists* are, in the main, a useful representation of the image data, but will suffer from fragmentation, unhelpful linkages between object and background, with several identifiable features (eg corners and arcs) embedded in any one list.

We wish to ascertain the location and probability of a set of *reference features* which have been defined, perhaps by manual annotation, as an intermediate level feature space suitable for the task at hand. The reference features might be important shapes, outlines or internal elements specific to particular objects, they might be generic features such as arcs or corners, or they might be complete object outlines.

ie we have a mapping from pixels to thinned edgels with property vector P:

$$I(x,y) \rightarrow E(x,y,P)$$

Edgels are formed into a set of segments {S}, each segment being itself a linked set of edgels:

$$E(x,y,p) \rightarrow \{S_1, S_2, \ldots S_n\}$$

Reference shapes $\{R_m\}$ are annotated example sets of edgels linked by a reliable method, eg by hand or under controlled illumination.

Our aim is to find the sub-set of {S} which provides best support for the existence of an R_j, subdividing particular S_i's where necessary.

Dynamic Programming

Dynamic programming in the context of pattern recognition is concerned with matching two patterns represented as a sequence of observations $\{X^1\}$ and $\{X^2\}$. Each element of X^1 is matched against

each element of X^2 and a local matching cost calculated. This local cost is added to the cost of the allowed previous states, to find the lowest accumulated cost so far, $C_{i,j}$. The allowed previous states are determined by a set of *productions*, as indicated in Figure 1, each with an associated cost. When all the $C_{i,j}$ has been computed, the optimum match is given by the lowest terminating cost, and if a record has been kept of the productions used at each state, the optimum match can be directly traced back through the $C_{i,j}$ matrix, where

$$C(i,j) =$$

$$c(i,j) + \min_{k}[P_k^{cost} + C(i+P_k^x, j + P_k^y)]$$

where $c(i,j)$ is the local distance metric between X_i^1 and X_j^2 and **P** is a set of production vectors and costs which define the permitted transitions between states.

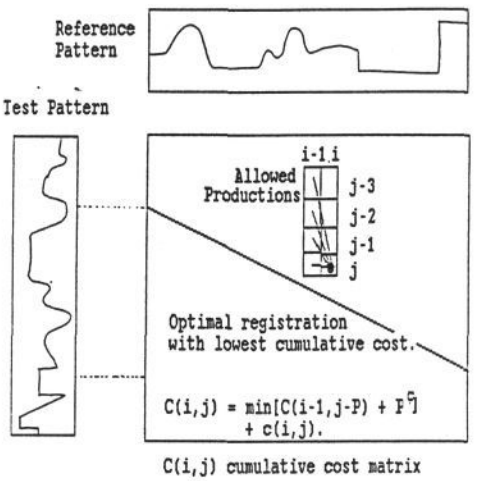


Figure 1. Dynamic programming algorithm, showing an embedded match.

If the reference and test patterns have m and n states respectively, the total computation scales as mxn, rather than n^m in the case of an exhaustive match. No loss of optimality occurs as long as later decisions in the cost calculations have no impact on decisions already made. The success of dynamic programming in shape analysis hinges upon whether this is an acceptable assumption.

Representation Used

Setting up the matching space and cost function is crucial to success in using DP, and it is here that other workers seem to have chosen inappropriate representations. After considering various options, an orientation vs distance mapping was adopted. This mapping, which has been used elsewhere in shape analysis and segmentation schemes, gives translation invariance and is amenable to rotation by the simple addition of an angle offset. It leads to a simple matching cost function, and local orientation can be extracted with reasonable accuracy. It is also plausible that a sequence of orientations along an outline should generalise well compared with, say, distance based metrics.

Linked segments passed to the algorithm are first converted into a re-sampled list of points of constant linear spacing (allowing for the cartesian sampling in the edgel location), with the local orientation and other properties (contrast and edge type) being stored as a list of records. Reference templates are treated in identical manner. Each segment is also represented in reversed form to allow tracing in either direction, and a matrix of Euclidean endpoint distances is formed between all segments.

The DP algorithm is applied between the reference and each segment of linked edgels, with productions which force a steady progression along the reference states. The local cost function used is

$$c^k_{i,j} = [\ A(\theta_i - \theta_j)\ ]^2/\text{angle\_weight}$$

where A delivers the angle difference and angle_weight is currently set to 10.

A cost accumulation matrix $C^k_{i,j}$ is set up for each segment, and at the initiation of the DP computation, all elements of each C^k are set to infinity except the columns $C^k_{1,-}$ which are set to zero to allow entry at any state in the segment at the start of the template. This enables embedded shapes to be segmented from a longer outline, providing only leading and trailing segments are connected to unwanted data (this can be achieved by the line extraction algorithms breaking all segments near other segment ends).

A cost matrix is set up for the productions P = {(-1,0) (-1,-1) (-1,-2) (-1,-3)} for every reference state, which enables the production costs to be dependent upon the position in the reference, eg make some parts

"springier" than others. Normally, the (-1,-1) production has zero cost, with a rising cost for the other productions to penalise distance distortions between the segment and the reference.

For the the first state of each segment a connection cost to all other segments is computed. This refers to the endpoint distance matrix and selects the lowest cost transition from the end states of all the other segments for the previous reference state.

ie on entry to the kth segment,

$$C^{k}_{1,j} = \min_{1} [\, d_{k,1} + C^{1}_{j-1,end} \,]$$

where d is the Euclidean distance between the start of segment k and the end of segment 1.

To allow for gaps between segments, a low cost threshold is imposed for the (-1,0) production at the start and end of each segment. For gaps or rogue points within a segment, another higher cost threshold is used to prevent the killing of otherwise good paths.

Closed outlines can be easily accommodated. Normally, a segment is not allowed to join to itself, and never to its reversed copy. However, if the reference template is a closed shape, ie its ends are close relative to its length, segments are permitted to join to themselves to allow an arbitrary start/end position around the shape.

During the computation, the best production used is recorded in a decision matrix. At the completion, all the cumulative costs are scanned for the lowest value, being the termination for the optimum route. Using the production decision matrix, it is then possible to trace back selecting the particular states on the optimum path, forming the recognised shape by selecting edgels from the input data. If more than one instance of the shape exists, several termination points will be found with comparable costs (assuming they do not include common segments). During the traceback phase, sub-costs can be accumulated for angle errors, segment gaps, and distortion, which can be used by higher level processes to select between a number of optimal and sub-optimal solutions.

Setting the Weight Parameters

A number of controlling weight parameters are needed to encourage the required behaviour. These are:

Angle Difference Weight. This scales the local cost imposed for a given angular difference between a segment and reference state (10).

Connection Weight. This scales the connection cost between segment ends (10).

Production Weights. Determines how springy the match will be (50).

Skip Penalty Limit for Ends. Sets a limit on the angle cost at the start and ends of segments, to allow a low cost for missing portions (100).

Skip Penalty Limit. Sets a limit within a segment for the cost from a rogue or missing edgel (200).

Kill Threshold. Sets an upper limit on accumulated costs which terminates a path part way.

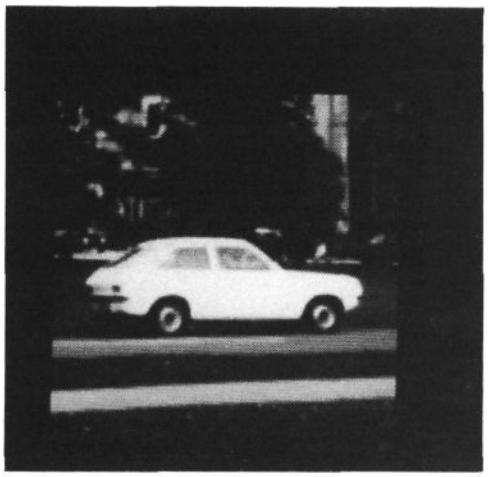
The figures currently used are shown in brackets. These have been derived on the basis that orientation noise is likely to be about 10 degrees, giving a matching cost of about 10 per pixel. A similar penalty is imposed per pixel gap between segments, and an off-diagonal production is penalised to be equivalent to about twice the angular noise. The skip penalty is set to be about twice this value, on the basis that gaps between segments will be small, and an arbitrary cost of double this was chosen for the within-segment threshold. So far there has been no need to alter these values, although adjustment will control how the algorithm generalises reference templates.

RESULTS

Figure 2 shows the segments extracted from a car image taken from the Alvey MMI/007 image database using the *Sleigh*[11] line extraction algorithm. In order to exercise the shape algorithm, some additional breaks and connections have been introduced by hand. Four reference shapes have been annotated from this data using a line segment editor, one corresponding to the

combined roof and bonnet profile, one to left hand window, a mirror image of this window, and finally the lower sill and wheel arches. These are displayed in Figure 3.

Note that the line segment which runs round the front of the car will need breaking to match to the 1st and 4th references, there are several breaks in the other segments, and the a gap has been introduced over the rear wheel arch with an extraneous segment terminating in the gap. The rear window is a closed shape with a start/end point different from the reference, and the right hand window, as well as being formed from two segments, has a slightly different shape to the reference.



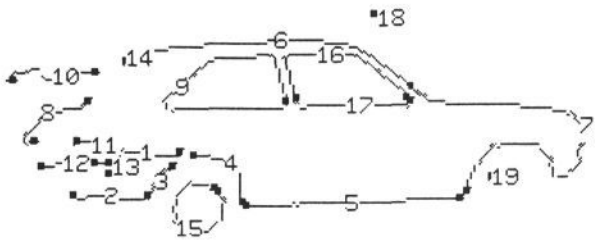


Figure 2. Image of car and extracted line segments

Figure 4 shows the identified shapes found by the DP algorithm, and it can be seen that the lines have been correctly re-segmented to give the best matches to the reference templates. Figure 5 shows the cost matrices for the first 12 segments when matched against reference 1, showing the path traced through segments 2, 3, 4, 5 and 7.

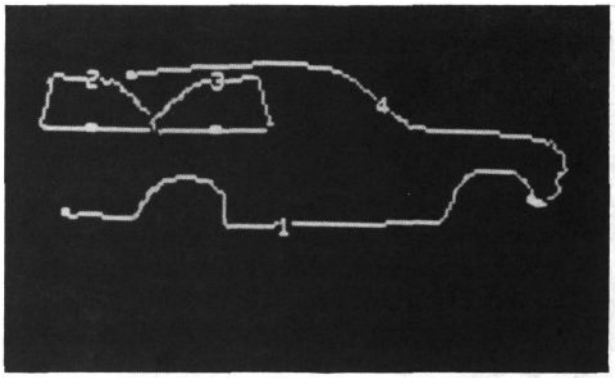


Figure 3. Reference templates annotated by hand

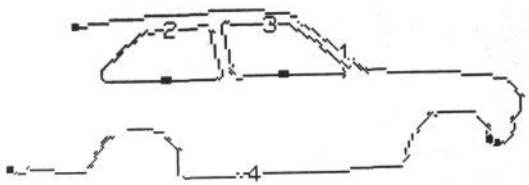


Figure 4. Re-segmented shapes extracted from Figure 2

The raw image was then reduced by a factor of two, and a new set of segments extracted, which were again matched to the original references. Once again the correct concatenations and segmentations were achieved (Figure 6) but, of course, the cost is greater because of the large number of off-diagonal productions. As discussed in the next section, it is possible to distinguish between angle and production costs, and to confirm that this match is a good one but at a different scale.

As a final illustration, Figure 7 shows the 1st and 4th references were applied to lines extracted from an image of a different car viewed from the same angle. A good match was obtained for the sill and wheel arch shape, but the limiting cost threshold was reached before a valid shape was found for the roof line. This is not unreasonable, since line 26 (the windscreen) is closed and has no endpoints near the desired lines 15 and 27. As already noted, although leading and trailing segments can be entered at any point without any cost, this cannot apply to interior segments. If the pre-segmentation stage had broken line 26 by the proximity of the ends of lines 27, a suitable match would have been found.

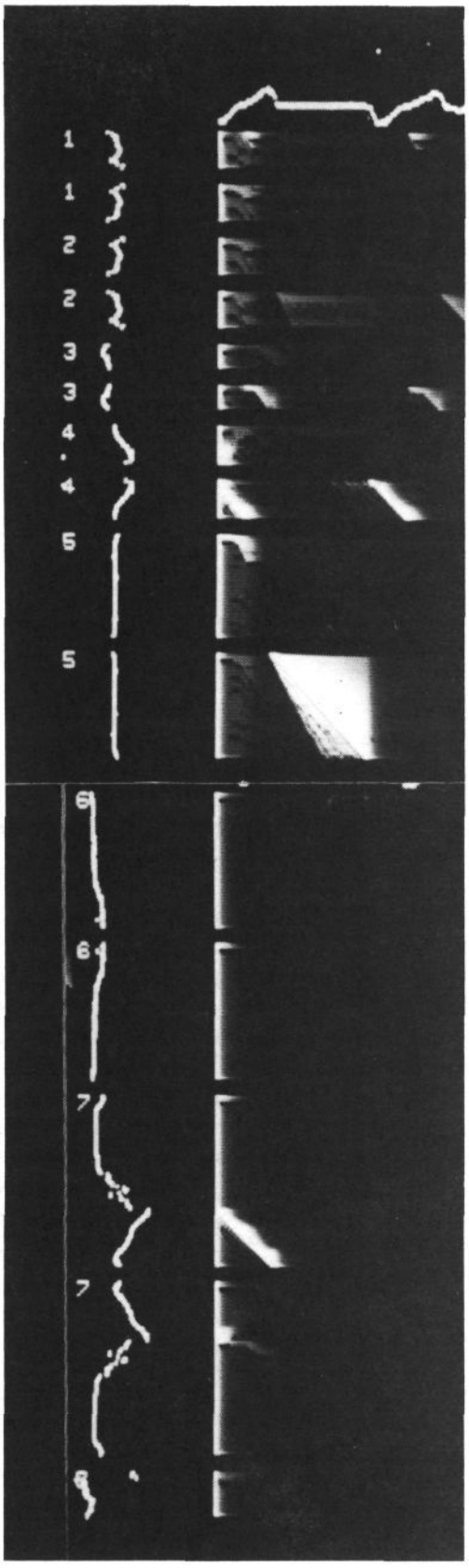


Figure 5. Cost matrices for reference 1 in Figure 3

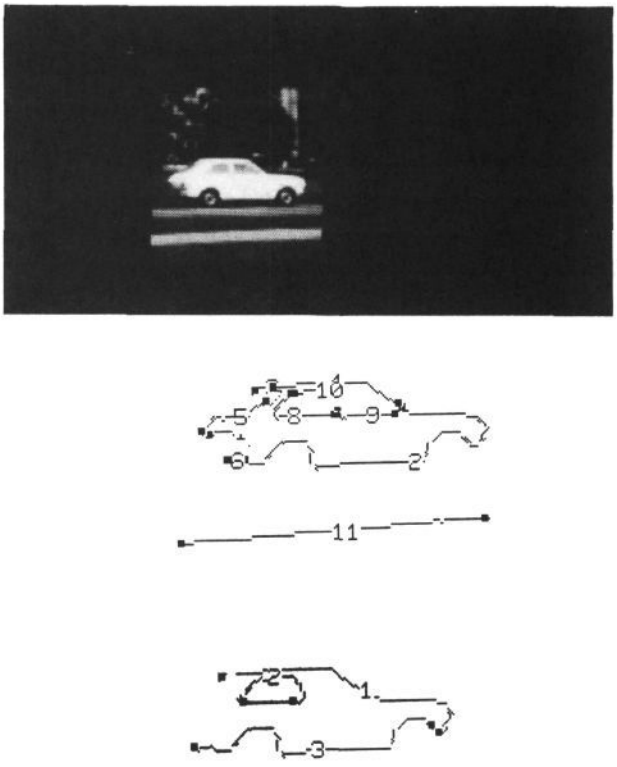


Figure 6. Image, lines and re-segmentation for half size image.

INVARIANCES AND AMBIGUITY

Any probabilistic evidence accumulation technique will invariably find some measure of support for any shape, but it is important that a valid match is distinguishable from random similarities. It is also important that in finding a particular poor match, a more appropriate match has not indavertently been rejected.

Ambiguity and Scale Invariance

Dynamic programming will guarantee to find the optimal solution, but only within the sequential decision constraint on which it is based. This leads to potential ambiguities in scaled data, since a shape with noisy orientation or non-linear distortion will accrue a similar global cost to a scale difference with the reference.

As described, the DP algorithm will match shapes which have a particular *sequence* of orientations, and is not so concerned with the distances between differing orientations. Increasing the off-diagonal production cost reduces tolerance to length distortions, but still does not encourage the costs to be distributed uniformly along the shape,

as might be desired. This does not cause confusion, since it is always possible to examine the distribution of costs in a solution after tracing the optimum path, but could result in a better solution, with more regular costs, never being found.

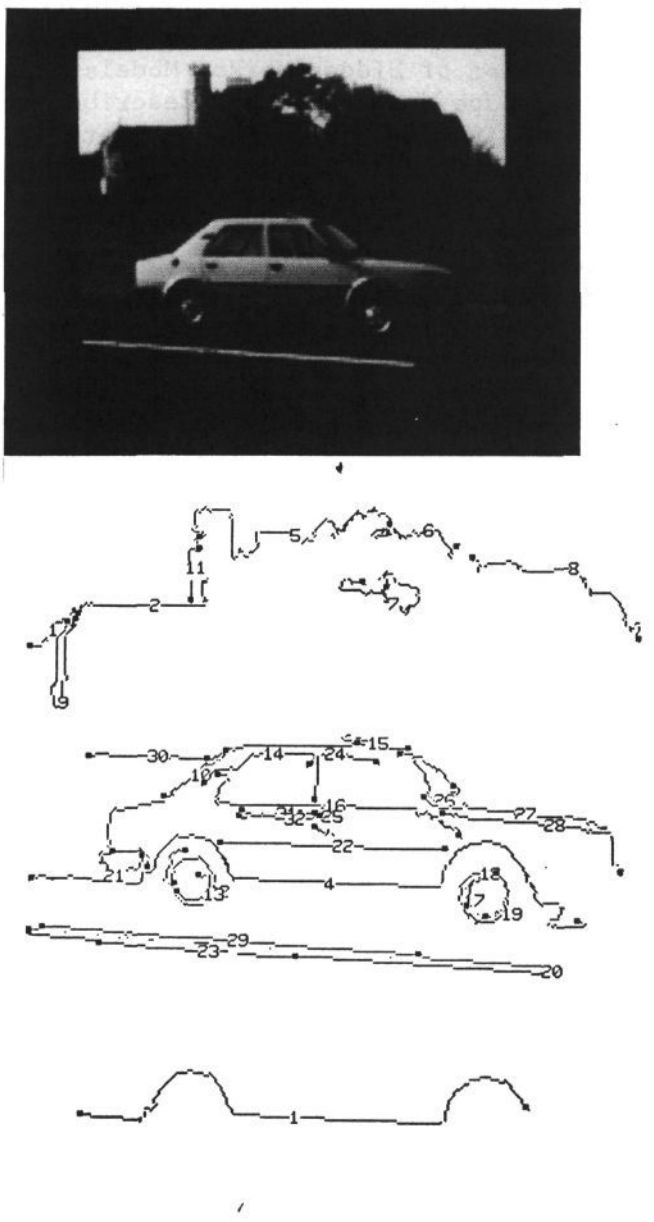


Figure 7. Lines extracted from different car, with only reference 1 *identified.*

The DP literature is rich with various refinements to overcome this effect, mainly based on adjustment of differing costs and use of longer range productions. Some gain can be achieved by these methods, but usually at considerable computational increase. A later section in this paper suggests a solution to this based on duration modelling in Hidden Markov Models, but a number of simpler extensions to DP algorithm are worth considering. The following strategy has been adopted to minimise the distortion ambiguities.

First, an edgel extraction algorithm is used which gives an accurate measure of local orientation, even in noise, by adaptively adjusting the scale of analysis in response to the local image properties. This is important to avoid swamping other costs by noisy orientation values. The non-linear angle cost function helps here. Serious ambiguities can arise from the horizontal production (ie staying at same point in the segment along the reference), but since this production is frequently used to compensate for scale mismatch, it cannot be given a high cost. An effective solution is to carry a scratch vector during the forward phase, this vector recording the number of times this production has been used consecutively for each segment position. The cost can therefore be made to rise to encourage a more uniform distortion, and can cut off after a certain number of stationary counts (corresponding to a gap or missing segment). This additional term does not guarantee optimality, with a better overall path being excluded by an earlier decision, but with the single valued data used in this application, this should not be a serious limitation.

To reject unreasonable distortions, a check can be made during traceback to ensure that a uniform production pattern emerges. Partial costs are economically formed for orientation, production, end connection and gap costs. It is then possible to reject a particular solution and look for others. In some cases, for example where an alternative solution does not share any segments, this can be done simply by tracing the next lowest cost, but if the rejected shape shares segments with a better solution, more drastic and sub-optimal methods must be used, such as marking the undesired route with a special cost, and re-applying the dynamic programming computation.

Rotation Invariance

An important advantage of the (S,θ) mapping is the easy way orientation shifts can be executed by adding a constant. If it is desired to match shapes in arbitrary orientations (this is often not a requirement), two approaches can be employed. If segments

are already well segmented, a simple approach is to normalise the orientation of the reference and segments. A more powerful approach is to use a second cost matrix to store the local orientation difference between reference and each edgel. The local cost then ignores angle difference, but the production cost then includes a term which encourages a constant orientation between adjacent states. This may lead to matches to "droopy" shapes, but this can be tested during traceback.

Computational Cost

If the reference shape has length L, and there are N line segments of total length P, the total number of d(i,j)'s to be calculated is

$$2(3P + N^2)L$$

where the first term represents the costs for applying the productions within each segment, and the second term the costs associated with segment transitions. A typical problem might have reference shapes 100 pixels long, and have 100 extracted segments with average length of 50 pixels. This involves 5 Million cost calculations for each reference. Each cost calculation involves about 5 operations, hence a 25 MOP processor would process one reference per second. By killing very expensive paths during the DP calculation, only a fraction of the cost matrix usually needs evaluation, bringing the figure down to below 10 MOPs per reference. Such a problem could be easily mapped onto a Transputer array, with each processor computing one reference match.

As a comparison, a generalised Hough implementation would require a much larger number of operations, depending upon the total image size and the number of degrees of freedom. Additionally, the entire accumulation space needs to be searched for a peak, instead of the simple traceback phase of the DP algorithm, to find the optimum shape.

STOCHASTIC MODELLING OF TEMPLATES AND SHAPE RIGIDITY

The DP algorithm as described here suffers from three limitations: a) it uses only one sample template for each reference; b) length distortions cannot be controlled without compromising optimality; and c) the various weights controlling the relative costs must be set by hand. All three limitations can be overcome by the use of hidden Markov models (HMM), particularly when the HMM is extended to include longer range correlations between states, using the methods developed in hidden semi-Markov models by *Russell & Moore*[11]. The application of Hidden Markov Models and DP to shape recognition is described in outline in this section. A detailed account is to be published separately.

The basic idea is to represent the template as the output of a set of states in an HMM, using a training set of example references to determine the HMM probabilities, using the Baum-Welch or Viterbi re-estimation algorithms. One also models the duration probability for each state, where the duration corresponds to staying in a particular HMM state for a given number of image segment pixels. One then uses the DP approach as before, except that instead of simple productions to step through a reference template, one has a set of duration probabilities determined by the training set which are equivalent to the production costs used before. Hence as well as capturing an archetype, the HMM also learns the distortion penalties along the reference. For example if, at some points in the reference shape, greater linear distortions are found, the re-estimation technique will give lower costs for off-diagonal productions than in the less variable parts.

Assuming a representative training set (the size of which will depend upon the variability within each shape), this approach offers significant improvement in both specific and generic shape recognition, with only a modest computational increase.

CONCLUDING REMARKS

A continuing difficulty in robust machine vision is providing a suitable interface between high level techniques and pixel operations. "Intermediate level" vision is increasingly being seen as the main remaining technological hurdle to achieve improved machine vision competence, and can be described as techniques whose role is to take data which has been formed with minimal commitment to a particular situation,

and produce symbolic statements about perceptually important features which can be directly mapped into high level object descriptions. Examples of such features include arcs, 'D'shapes, polygons, etc, but also include objects' specific outlines or outline elements, such as wheel arches, roof shapes, window outlines.

This paper extends previous attempts to use dynamic programming in shape analysis, and successfully addresses the concatenation and re-segmentation of fragmented and embedded line segments using reference shapes as the constraint. The technique can be used to recognise shapes or image features in its own right, or it can be used to select edgel data which is then passed to other classification methods.

Acknowledgement

I would like to acknowledge the useful discussions with my colleagues J S Bridle and Dr M J Russell.

REFERENCES

1. **Brady, M. & Asada, H.** "The curvature primal sketch" *IEEE Trans. PAMI* Vol 8 (1986).

2. **Ballard, D.H.** "Generalising the Hough transform to detect arbitrary shapes" *Pattern Recognition* Vol 13, 2, (1981) pp111-122.

3. **Sullivan, G.D.** "Alvey MMI/007 vehicle exemplar: performance limitations" *Proc. 3rd Alvey Vision Conference* (1987) pp39-45.

4. **Bolles, R.C. & Cain, R.A.** "Recognising and locating partially visible objects, the local-feature-focus method" *Int. J. Robotics Research (1982)* Vol 1 pp 57-82.

5. **Bellman, R.** *Dynamic Programming* Princeton Univ. Press (1957).

6. **Sakoe, H. & Chiba, S.** "Dynamic programming algorithm optimisation for spoken word recognition" *IEEE Trans. Acoustics, Speech and Signal Processing* Vol 26 (1978) pp 43-49.

7. **Ohta, Y. & Kanade, T.** "Stereo by intra- and inter-scanline search using dynamic programming" *IEEE PAMI* Vol 7 (1985) pp 139-154.

8. **Furst, M.A. & Caines, P.E.** "Edge detection for digital grey level images via dynamic programming" *Proc. IEEE 7th Int. Conf. on Pattern Recognition* (1984) pp 55-58.

9. **Fischler, M.A. & Elschlager, R.A.** "The representation and matching of pictorial patterns" *IEEE Trans. Computers* Vol 22 (1978) pp 67-93.

10. **Maitre, H. & Wu, Y.** "Improving dynamic programming to solve image registration" *Pattern Recognition* Vol 20 (1987) pp 443-462.

11. **Sleigh, A.C.** "The extraction of boundaries using local measures driven by rules" *Pattern Recognition Letters* (1984) pp 247-258.

12. **Russell, M.J. & Moore, R.K.** "Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition" *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing* (1985).