Model Based 3D Grouping by Using 2D Cues

S. Zhang, L. Du, G. D. Sullivan and K. D. Baker

Department of Computer Science
University of Reading
RG6 2AX, U. K.
e_mail: zsj@uk.ac.reading

Methods are reported for deriving polygonal approximations of connected edges, and for identifying cues which indicate the presence of cars. Simple attributes of cue features are stored on the viewsphere, allowing numerical inversion to recover the viewpoint. Further features are then grouped with the cue, by projecting a 3D model into the image.

INTRODUCTION

This paper reports recent work on the use of model-based methods to identify vehicles in outdoor scenes. The paper builds on previous work, using an hypothesis-and-test paradigm consisting of the following main stages:

(i) Context-independent edge detection, followed by a local connection to find extended sequences of points (curves) in the image (after Canny, 1984).

(ii) Decomposion of curves into polyline approximations, using a split and merge algorithm based on curvature extrema (Angelikaki, 1988).

(iii) Identification of model-dependent "cues" in the polyline database associated with given features of the model (Sullivan *et al*, 1987).

(iv) 2-D reasoning to group a labelled cue with other fragments in the polyline database (Boddington *et al*, 1989).

(v) Use of viewpoint reasoning to identify the view-patch from which the extended cue set is visible (Rydz *et al*, 1987; Worrall *et al*, 1989)

(vi) Invertion of the view-point, using an iterative technique (Worrall *et al*, 1987).

(vii) Refinement and verification the view-point estimate, by iconic evaluation (Brisdon *et al*, 1988)

Stages (i-iii) are data-driven, and make no reference to the internal representation of the car, except for the choice of car-specific cues (quadrilateral, S- and U-shapes). Stages (iv-vii) involve a hypothesis-driven search process which is managed by means of a truth maintenance system (the ATMS) to avoid excessive re-evaluation of constraints (Boddington *et al*, 1990).

Currently, the weakest stages of the process are probably (ii-iv), and previous demonstrations have been forced to use manual interaction at these stages, in all but the simplest of images (see Boddington *et al*, 1990).

We report here improvements to the cue-finding stages (steps ii & iii) and new work using pre-compiled tables of the view-dependency of features to estimate the viewpoint, thus allowing image features to be grouped by means of 3-D reasoning.

CUE EXTRACTION

The edgelets produced by the Canny edge detector are first thresholded to reduce weak clutter. Beginning from an edgelet, we track along connected neighbours to get a list of edgelets. If one edgelet has more than two neighbours, we check all possible extensions of the curve, and segment the curves on a largest-first basis, breaking the others to form separate curves. This removes small spurs effectively.

1. Linear Approximation

Teh and Chin[8] surveyed several methods for finding dominant points on a curve suitable for creating a linear approximation of the curve, and suggested their own. All but one algorithm surveyed in the paper require explicit input parameters. These parameters usually control the region of support for the measurement of local properties (e.g., curvature) at each point on the curve and so serve to measure the changes between the original curve and its linear approximation. The algorithm must be tuned to the level of detail represented by the digital curve. Generally speaking, it is difficult to find a set of parameters suitable for an image that consists of multiple size features.

Teh and Chin suggested their own dominant-point finding algorithm, which requires no input parameters. They therefore claimed that the algorithm is suitable for different scales of curves. We have checked their method in our images and found that it favours small curves. For larger curves, say with more than 100 edgelets, it often produces too many dominant points. The approximation of the curve is good but the over-segmentation reduces its usefulness for further processing. We have therefore developed methods better suited to finding salient features, based on the chain code (Freeman, 1974).

The connected curve is first divided into short straight lines comprising compact sequences of horizontal, vertical and diagonal vectors. Further grouping deletes

BMVC 1990 doi:10.5244/C.4.39

repeating patterns in the Freeman chain. For example, if pattern 0 1, appears repeatedly, 0 1 0 1 0 1 0 1, we can group the corresponding edgelets into a straight line sector. Only 8 such patterns, 01, 12, 23, 34, 45, 56, 67 and 70 are used, together with their equivalent reversed forms.

This stage results in over-segmented curves. The straight line sectors can be combined further subject to two constraints: the length of a line sector, and the angle between adjacent lines. Since the main object of the present work is to find highly salient features in the image, we are uninterested in lines shorter than one tenth of the length of the longest one in a curve. Let curve C be approximated by straight lines $\{l_1, l_2, ..., l_t\}$, among which l_k is the longest. If there is a line l_i whose length is shorter than one tenth of that of l_k, we can check the angles θ_i between $l_{i-1}$and l_i, and θ_{i+1} between l_i and l_{i+1} and combine l_i with l_{i-1} if $\theta_i < \theta_{i+1}$, and combine l_i with l_{i+1} otherwise. Angles between adjacent lines can be treated similarly. If the angle between two lines is smaller than a certain threshold, those two lines can be combined into one.

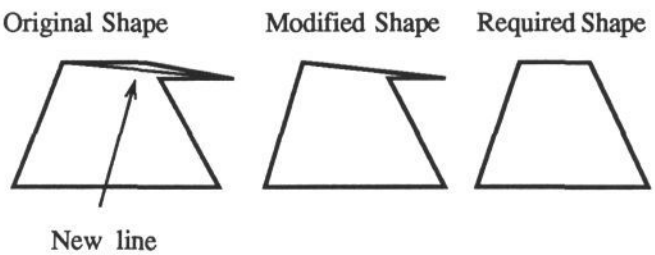

Figure 1. Errors introduced by line merging

A common method to reduce further the number of points on the curve is to measure the distance from a point to the line connecting its two neighbours. If this distance is under some threshold, the point can be deleted from the curve. This sometimes fail badly as shown in Fig. 1. We seek compact feature, such as the trapezium (Fig.1, right) and these are not explicitly favoured by the distance algorithm. Instead, we have developed a method based on an area measurement.

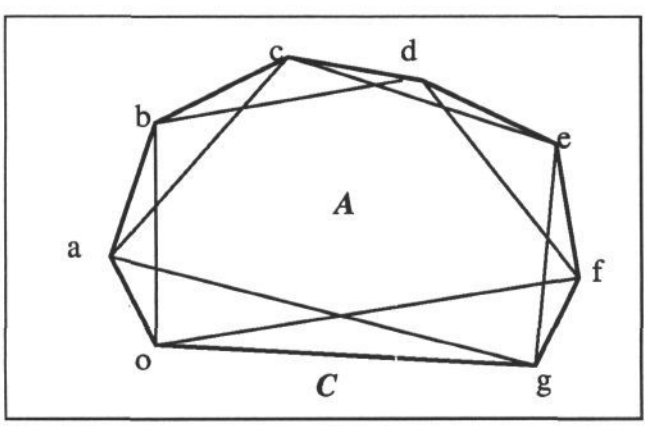

Figure 2. Area-based merge algorithm

The areas of convex portions of a curve can be used to measure the difference between the curve and its linear approximation. If a curve is not convex we break it into sub-curves which are convex. Let the area of curve C be A (if it is not closed, then connect the two ends points). Each vertex on C defines a triangle with its immediate neighbours, as shown in Fig.2. We select the smallest triangle and remove the corresponding vertex if the area of the triangle is less than a certain percentage (say 5%) of A. This is iterated until no further removals are possible. This method is very effective at removing small protuberances such as shown in Fig.1.

2. Classification

The polylines are searched to find quadrilaterals, U-shape curves, and S-shape curves to serve as cue features. On the basis of experience with common feature extractors, we believe that the most salient features for recognizing a car are edges at the windows, hatchback, and bonnet. These retain identifiable shapes under a range of view-points and therefore can be used as "focus features" to make hypotheses about the car.

For each linear approximation L of a curve C, we see if it matches with one of the models of S-shape curve, U-shape curve, or quadrilateral. These models are expressed by sets of rules in a rulebase. When L is matched, we classify C into that category of features. The rules for a quadrilateral are listed below; those for an S-shape and U-shape are similar.

A polyline P is reduced to a quadrilateral by line merging under the following constraints:

1. P is closed, or the distance between the two end points of the curve is less than one fifth of the length of the longest line segment in P.
2. Four of the exterior angles between successive lines must be significantly larger than other angles (5 times as large as others) after the line merging process. The sum of the four largest angles is within the range 320° to 400°.
3. The line segments are merged between the four largest angles into 4 line segments. The longest grouped line segment must not be longer than 5 times the length of the shortest one.
4. At least one pair of the non-adjacent line segments is parallel, as defined by the overlap, distance and orientation of the line segments.

3. Hypothesis Generation

Salient features are used as "focus features" or "seed features" to make hypotheses about the car. In the system so far, we use only quadrilateral as seed feature. The others are used to support the hypotheses. Correspondingly, we use car windows as model features to make the match. To limit the search space we assume that the car is upright and the camera axis is between 0° to 45° from the ground.

Let P be the set of all the quadrilaterals extracted from the image. For every quadrilateral p_i, we can associate any of the six windows $W=\{w_1, w_2, w_3, w_4, w_5, w_6\}$ of the car and thus generate a hypothesis

$$H_{ij}=\{p_i,w_j\}$$

This in turn can be used to make predictions about the car by using the quantitative model discussed in the following section.

To reduce the number of hypotheses, we define a function f: H->I where H is the set of hypotheses, H=P*W. For a quadrilateral p_i we define a subset F_i of image features which includes all the image features in a small area centered at p_i. The area is determined by the vertical extent in the image of p_i, since this changes little in most of the view-points. For window w_j we define a subset W_j of model features. When we find one feature in P_i matching one feature in W_j we increase the value of $f(H_{ij})$ by one.

A hypothesis H_{ij} is said to be a good hypothesis (GH) if $f(H_{ij})$ is greater than a certain threshold (say 5). The GHs are used as input to the quantitative model to give predictions of the car. The number of GHs is usually smaller than 5.

POSE ESTIMATION

We estimate the view direction using pre-computed maps of single features, represented on the view sphere, in a way similar to Goad (1987). The cue features delivered by the above are trapezoids, and the following relational two attributes are computed.

Let line L_1 be the lower of the two parallel lines and line L_2 be the line to its left. We calculate

- r -- the ratio of length between L_1 and L_2.
- α-- the angle between L_1 and L_2.

The feature attributes are as shown in Fig. 3.

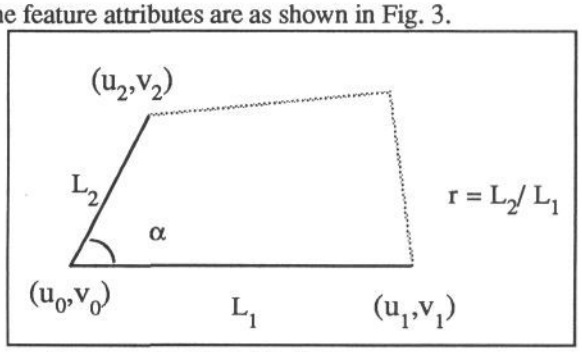

Figure 3. Image Feature Attributes

In an object-centred coordinate system, the camera position is specified by 3 translational parameters and 3 rotational parameters. We use the spherical-polar representation with two angles θ and φ and one distance measurement r for the position of the camera nodal point in the object frame. A camera coordinate system is then established with its origin at the nodal point and its axis Y_c intercept the object centre. The camera rotations are represented by a roll γ_{yc} (around camera axis Y_c), and tilt, γ_{xc} (around axis X_c), pan γ_{zc} (around axis Z_c). Therefore, in our new representation the parameters are:

(1) R -- spherical distance.

(2) θ -- angle of the camera axis formed with the Z axis of the object frame.

(3) φ-- angle of the projection of camera axis on the X-Y plane with X axis.

(4) γ_{yc}-- roll angle of the camera.

(5) γ_{xc}--tilt angle of the camera.

(6) γ_{zc}--pan angle of the camera.

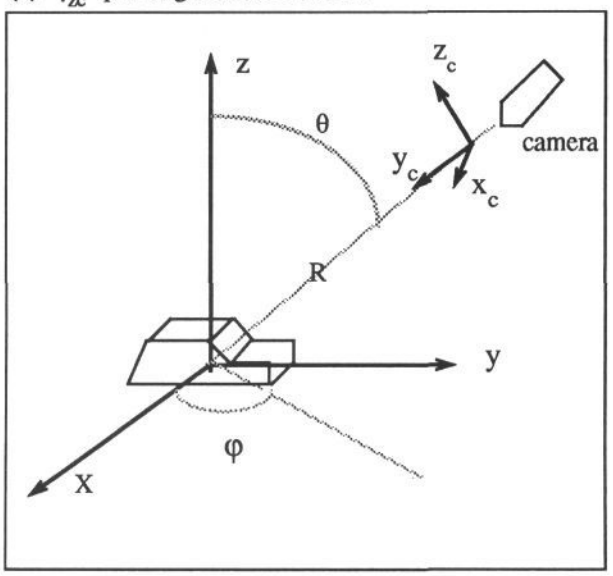

Figure 4. Coordinate Systems and Symbols

The position of the cue in the image constrains two of the translation components (in X_c and Z_c). To simplify the problem, the Y_c axis initially assumed to intercept the centre of the cue. The residual 4D problem is then decomposed into the following steps.

(1) Estimate view direction as a point on the viewsphere.

(2) Estimate the viewing distance.

(3) Estimate the roll angle.

(4) Relocate the Y_c axis to intercept the model centre and re-estimate the pan and tilt angles and the translations in X_c and Z_c.

1. View direction from shape attributes

Under the above assumptions, the determination of view direction parameters is decoupled from the other parameters. The feature attributes defined in Figure 3 are functions of the view direction as shown in Figure 4.

$$\alpha = f(\theta, \varphi) \qquad r = g(\theta, \varphi)$$

The recovery of view direction from given feature attributes requires inverting the functions f and g. A numerical solution is used. We represent the functions f and g numerically as two tables of sample points with small intervals for θ and φ. The inversion operation is carried out in table-look-up fashion. Sub-interval accuracy of inversion is achieved through linear interpolation of the functions f and g. These functions are tabulated for each object feature, in this report, the

windows.

A coarse view direction estimation is produced by looking up the corresponding table for the feature under investigation. Further on, this estimation is refined to sub-interval accuracy by using linear interpolation of the table in the immediate neighbourhood of the coarse view estimate.

In building the view-shape table for the car recognition problem, θ is restricted to the range of 45°-90° (implying that the car is not viewed from high above nor from below.) This is not essential to the approach, but a simplification for efficiency. The range of φ in which any feature is visible is 180 degrees. We partition it into two parts, since the near symmetry of the functions f and g means that a table covering 180 degree of longitude is ambiguous.

2. Estimation of View Distance (R)

Under the assumption that the camera axis is pointing towards the object centre, the distance R can be determined by size-depth scaling.

Let u, v be the position of a image point corresponding to (X_c, Y_c, Z_c) in the camera centred system onto the image plane, f be the focal length. u and v are in pixels. We have

$$u = f X_c / Y_c$$

$$v = f Z_c / Y_c.$$

In building the table, $Yc = R_{tbl}$ is used. Suppose, point (X_{c1}, Y_{c1}, Z_{c1}) and point (X_{c2}, Y_{c2}, Z_{c2}) are the two end points of a line in consideration. The length the projection of the line is,

$$LEN_{tbl} = f / R_{tbl} \; [(X_{c1} - X_{c2})^2 + (Z_{c1} - Z_{c2})^2]^{1/2}$$

When R changes with camera axis kept pointing towards the object centre. X_{c2}, Z_{c2} and X_{c2}, Z_{c2} will not change. This gives,

$$LEN = f / R \; [(X_{c1} - X_{c2})^2 + (Z_{c1} - Z_{c2})^2]^{1/2}$$

for the length of the same line projection. Therefore, the distance between the camera and the object centre can be estimated by the following,

$$R = R_{tbl} * LEN / LEN_{tbl}$$

3. Calculation the Roll Angle (γ_{yc})

Having estimated the view direction (θ and φ) the roll angle can be easily computed as the angle needed to bring line 1 from image parallel to its counterpart in the predicted template.

4. Correction for pan(γ_{xc}) and tilt(γ_{zc})

We initially assumed that the displacement of the cue feature in the image was only due to the camera pan and tilt. This neglects any displacement of the feature with respect to the centre of the model. Having established an estimate of θ, φ, R and the roll angle the initial assumption can be corrected by computing the pan and tilt angles which make the cue feature coincide with the predicted model feature.

5. Error Assessment by Simulation

The error of estimation has been assessed by simulation. The image features are generated from the model with given view directions (θ and φ) and distance, together with translation of Z_c, X_c. The difference between the given view parameters and the estimated parameters is the error. The typical error is less than half of the sampling interval of θ or φ. The error of r is less than 5%.

The super-imposed estimated template and the simulation input appear to match well without significant perceptible error.

MODEL BASED GROUPING

Figure 5 shows an original image which is a typical outdoor scene including several cars and other irrelevant objects. Figure 6 shows the result of the Canny edge detector. Figure 7 shows the polyline descriptions of the main connected curves, and figure 8 shows the quadrilaterals detected in it which are used as seed cues.

Figure 9 shows the polylines around one seed cue in greater detail, with the seed cue emphasised. A single quadrilateral is a candidate for any one of 6 windows in the model, and each window may lead to several solutions of the viewsphere functions (see above). Figure 10 shows the solution obtained near the visibly correct position, and figure 12 shows an incorrect solution, corresponding to a mirror confusion in the feature attributes.

The projected instances of the model in figures 10 & 12 are used to initiate a search for fragments of lines in the polyline database, which closely match the model features. The match seeks lines that are close and parallel (according to simple criteria). Figures 11 and 13 show the lines identified in the two cases. Each of the lines grouped by model-based reasoning is associated with known model lines, which can be used for subsequent view-point inversion (Lowe, 1987; Worrall *et al*, 1988).

CONCLUSION

Improvements to an existing model-based system for recognising cars have been presented. Data-driven feature analysis stages have been developed which derive an accurate description of the most conspicuous edges in the image. Cue extraction rules are able to derive application-dependent feature groups indicative of cars, which are robust against changes in viewpoint.

It has also been shown that the viewpoint can be estimated with fair accuracy from simple polyhedral cues, and that this allows 3-D model-based reasoning to be used to group additional fragmentary evidence from the line database. The additional evidence may in turn be used to compute the pose more accurately, or to discriminate between likely and unlikely hypotheses for full evaluation.

Fig.5

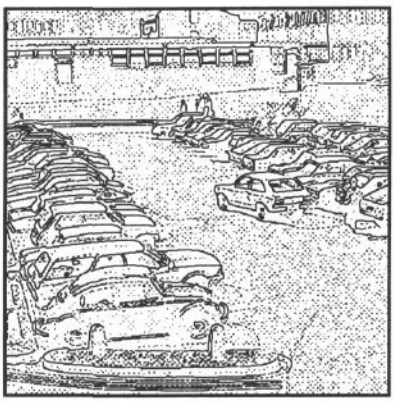

Fig.6

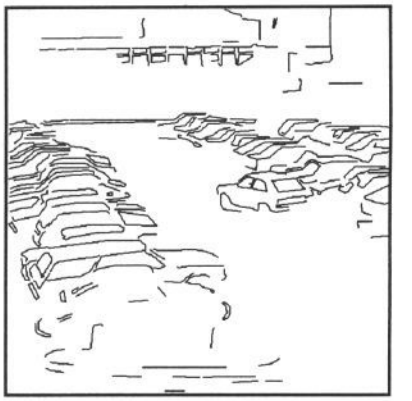

Fig. 7

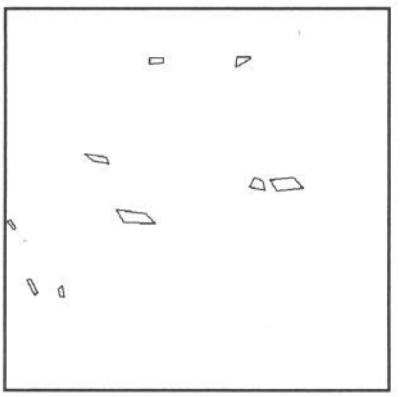

Fig. 8

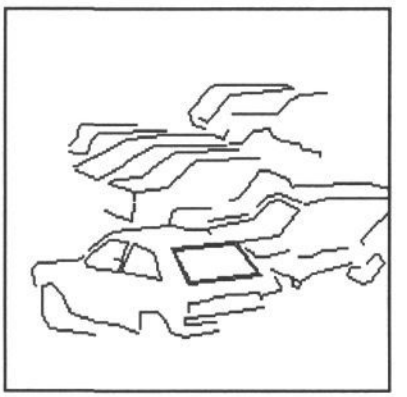

Fig. 9

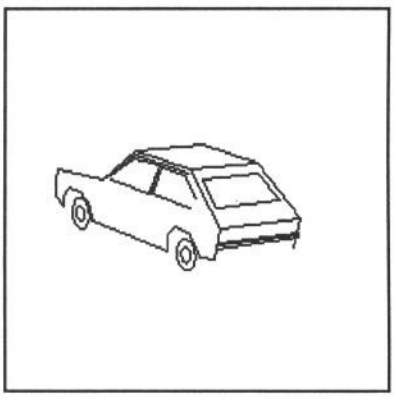

Fig. 10

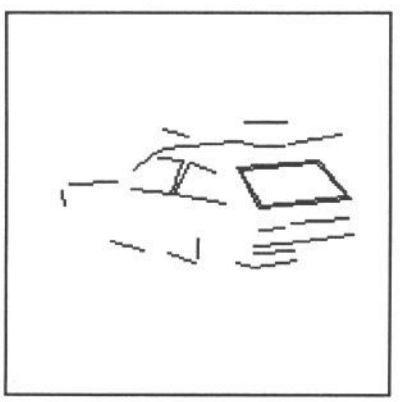

Fig. 11

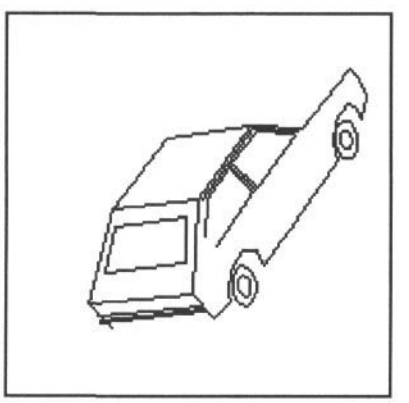

Fig. 12

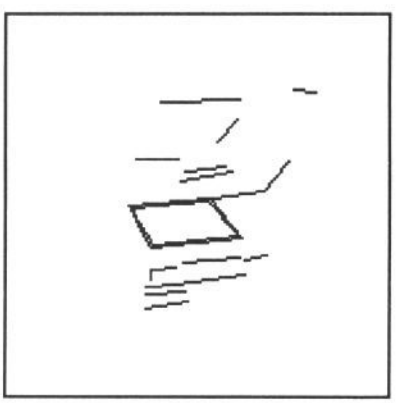

Fig. 13

REFERENCES

1. Angelikaki,T., Internal report, Dept. of Computer Science, University of Reading, 1988
2. Bodington, R. M., Sullivan, G, D, and Baker, K. D, "Experiments on the Use of the ATMS to Label Features for Object Recognition", Proceedings of First European Conference on Computer Vision, April, 1990
3. Bodington, R. M., Sullivan, G, D, and Baker, K. D, "The Consistent Labelling of Image Features Using an ATMS", Image Vision & Computing, Vol. 7, No. 1, Feb., 1989
4. Brisdon, K., Sullivan, G. D., Baker, K. D., "Feature Aggregationin Iconic Model Evaluation", AVC-88 Manchester(1988)
5. Canny, J. F., "Finding Edges and Lines in Images", Ph. D Dissertation, AI-laboratory, MIT, Cambridge, MA, 1983
6. Freeman, H., "Computer Processing of Line Drawing Image", Computer Surveys, 6, 1974
7. Goad, C., "Special Purpose Automatic Programming for 3D model-Based Vision", Readings in Computer Vision" (ed) Martin, A. Fishler, 1987.
8. Lowe, D. G., "The Viewpoint Consistency Constraint" International Journal of Computer Vision, Vol. 1 (1987), pp 57-72
9. Rydz., A. E., Sullivan, G., D., Baker., K. D., "Model-Based Vision using a Planar Representation of the Viewsphere", AVC-88 Manchester (1988)
10. Sullivan, G. D., "Alvey MMI-007 Vehicle Exemplar: Performance & Limitations", AVC-87, Cambridge,England (1987)
11. Teh, C., Chin, R. T., "On the Detection of Dominant Points on Digital Curves", IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-11, No.8, pp. 859-872, Aug., 1989
12. Worrall, A., D., Baker, K. D., "Model Based Perspective Inversion", AVC-88 Manchester (1988)