# ROAD EDGE TRACKING FOR ROBOT ROAD FOLLOWING

A.D. Morgan E.L. Dagless D.J. Milford B.T. Thomas

Faculty of Engineering, University of Bristol, Bristol, UK.

*The problem of navigating a robot along a road is approached by means of creating and updating a simple representation of the road from a sequence of images. The representation chosen is a 4-parameter model that describes the width, direction and simple curvature of the road in a vehicle centred (X, Y, Z) world coordinate system. The model is created from tracking along major edge features in an image and applying constraints to select road edge candidates. Updating consists of tracking a set of measured edge points from frame to frame (assuming that vehicle motion is known) and using a weighted least squares process to find the 4 parameters of the road model. A number of constraint and filtering processes representing knowledge of how a vehicle moves on a road have been applied.*

The application of computer vision techniques to the automatic guidance of mobile robots is a very challenging task. In the case of a road vehicle this is especially so considering the competence of human vision and the variety of other human skills that are used and the efforts taken to learn how to coordinate them. In addition the enviroment can become very complicated from simple well-defined roads to complex ill-defined roads that include lay-by's, junctions and roundabouts all with moving potential obstacles. Initial research in the USA [1,2,3] and [4,5,6] has demonstrated the automatic navigation of road vehicles over simple, well-defined roads at a modest speed using image processing techniques that process large regions of the image. Research in Germany [7,8] has shown that the use of dynamic and geometric models of the road and the vehicle can be used to focus the image processing into gathering information about the scene from small windows. These techniques have been applied to tracking edges in very well defined roads at much higher speeds, an approach supported by human factors reasearch into driver's eye movements while negotiating curved roads [10]. However, it seems likely that a more competent visual navigation system would have to include a great wealth of information about the enviroment in the form of hierarchical models that guide and are updated by image processing of large parts of the image. An important component of such a system would be a model of the road edges.

In subsequent sections, a simple model of a pair of road edges is described which is used to guide the tracking of edge points through a sequence of images. An orientated 'edge-finder' operator is descibed that determines the edge point positions which are the data for a weighted least squares curve fitting process to determine the parameters of the model at each frame. Finally, a description of the overall algorithm is given with initial results and conclusions.

## ROAD MODEL

In computer vision, a model is an application specific representation of features of interest in a scene that is capable of predicting the form of features in an image or the behaviour of features in a sequence. The use of a model is important as it allows the results from processing of previous images to influence the current processing.

### A Four-Parameter Road Model

The simplest possible model of a road is a pair of straight lines in an (x,y) image coordinate system. This crude representation is obviously inadequate when the road begins to curve but also is not convenient to accomodate edge position changes in the image due to measured vehicle motion. A more powerful and convenient representation is to think of the road as comprising a flat surface bounded by two road edges that are each described by a circular arc in a vehicle centred (X,Y,Z) world coordinate system (initially assume that the earth is flat so that Y is a constant) as in Figure 1. In the first instance it is reasonable to assume that the road-width is constant or varies slowly from frame to frame. This means that the left and right circular arcs having radii $R_{left}$ and $R_{right}$ respectively share a common centre $(Z_0, X_0)$. These are the four parameters of the model which are related by,

$$(X_{left} - X_0)^2 = R^2_{left} - (Z_{left} - Z_0)^2 \quad (1)$$

$$(X_{right} - X_0)^2 = R^2_{right} - (Z_{right} - Z_0)^2 \quad (2)$$

$$roadwidth = \mid R_{right} - R_{left} \mid \quad (3)$$

The radius of the circular arc determines the curvature, with a very large radius giving an almost straight section and the position of the centre of the circle determining the general direction of the arc. The lateral offset of the vehicle from the left edge for example, is obtained when $Z = 0$. This 4-parameter model is capable of representing the position of the robot on the road, the general direction of the road and a simple curvature. This representation has proved to be adequate for the majority of road scenes that we have encountered. However, there are situations in which this representation is inadequate. For example, a long straight section followed by a curve or a section with two or more curves. A more powerful representation would be to use cubic spline approximations or other higher order polynomials to model the

AVC 1988 doi:10.5244/C.2.28

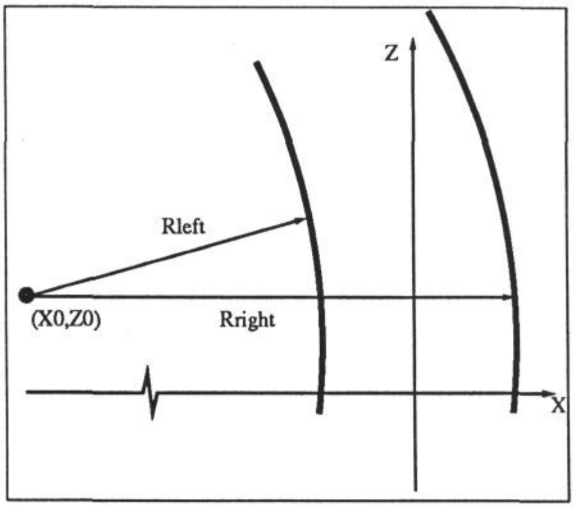


*Figure 1: Two circular arcs with a common centre* $X_0, Z_0$.

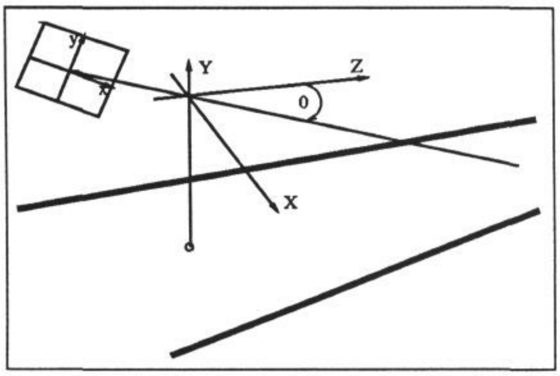


*Figure 2: The coordinate system.*

road edges. Other representations [11,12] based upon a 3D "ribbon" have been described that model the road with both a horizontal and vertical curve. The difficulty with these representations is that the uncertainty in the parameter values is increased as more parameters have to be identified from the information extracted from the image.

### Using the Model

In order to relate information extracted from the image to features in the world and to the road model it is necessary to be able to transform between the (x,y) image and (X,Y,Z) world coordinate systems. The transformation is direct if it assumed that the earth is locally flat and the height and look angle of the camera are known. Figure 2. shows the two coordinate systems. The following expressions assume that the camera is looking straight ahead with a tilt angle $\theta$. This was found to be necessary in order to have the road appear in the majority of the image.

(x,y) $\Leftarrow$ (X,Y,Z)

$$x = \frac{FX\sec\theta}{Z - Y\tan\theta} \tag{4}$$

$$y = \frac{F(Y + Z\tan\theta)}{Z - Y\tan\theta} \tag{5}$$

(X,Y,Z) $\Leftarrow$ (x,y)

$$Z = \frac{Y(F + y\tan\theta)}{y - F\tan\theta} \tag{6}$$

$$Y = \mathit{constant} \tag{7}$$

$$X = \frac{(Z\cos\theta - Y\sin\theta)x}{F} \tag{8}$$

where F is the focal length of the camera. Under the assumptions made these transformations allow a depth and lateral offset to be assigned to each pixel in the image.

In the situation where a road model has been initialised from edge information extracted from a particular image (described in a later section), then the processing of subsequent images should take into account information from the previous image. It has been assumed that the forward velocity of the vehicle can be measured which provides distance travelled $\Delta Z$ in an algorithm cycle time. The (X,Y,Z) edge point data can then be updated with this $\Delta Z$ to provide predicted edge locations in the next image. There are two ways of updating the model: Firstly searching for edge points in a fixed grid ahead of the vehicle. In this case, to use the edge information from the previous frame it would be necessary to interpolate a distance $\Delta Z$ between known (X,Y,Z) edge points to find the predicted points on the grid. An alternative method is to just update the (X,Y,Z) edge positions with $\Delta Z$ to provide the predicted edge locations in the next image. The effect of this is have fixed edge points in the world that sweep into and then out of view as the vehicle moves forward. This scheme was used in the algorithms described below.

## EDGE TRACKING IN AN IMAGE

The appearence of road edges in an image can vary greatly depending upon such things as:

- road design, age and condition, painted lane markers, kerb-stones, grass or dust verges,
- daily weather conditions such as intermittent sunshine, shadows and poor contrast, rain and reflections, and
- seasonal effects such as snow, ice, leaves, and blown dust.

Eventually it will be necessary to be able to detect road edges under all possible conditions but in order to make progress with the use of edge information it is necessary to make the assumption that the road scene is simple and that the road edges are reasonably well defined.

### An Orientated Edge-Finder Operator

The usual approach to detecting an edge in a region of an image is to firstly highlight those pixels that could be edge pixels and then to group them together into edge segments. The highlighting can be achieved using convolution operators that vary in complexity, ranging from a simple 3x3 Sobel up to the Marr-Hildreth DOG operator. The grouping process could be a test for connected pixels or commonly a Hough Transform as used in [6]. Whichever combination of methods is chosen there is always a large number of arithmetic operations to perform. By designing novel, special purpose computer architectures that are well suited to such low-level image processing tasks, the execution times can be significantly reduced. At present though realistic algorithm cycle times are between 10 and 1000 times greater than video frame rates [3,6]. One approach to detecting edges that is less computationally intensive is to assume that the approximate location and orientation of the edge can be predicted from previous knowledge stored as a model. Then, after correlation of an edge template over a region positioned by a model, search the resulting surface for the best match. This also requires a large number of arithmetic operations if performed in two dimensions. However, the appearence of a road edge in a region of an image is essentially just one dimensional and its orientation predictable from a model. An edge template that is sensitive to an edge of a particular orientation need only be correlated in a direction normal to the expected direction of the edge. This approach was first proposed in [8]. The simplest edge template, $[1, 0, -1]$ using integer coefficients, is sensitive to vertical edges. The first problem with simple operators of this kind is that they are sensitive to edges of a particular size at a particular image resolution. The operator needs to be designed so that it is sensitive to the size of edge feature of interest. The second problem is that they are very sensitive to the effects of random noise in an image. In order to improve the signal-to-noise ratio, the operator can be replicated along the extent of the expected edge direction. For example, an operator sensitive to edges of a particular scale at 45 degrees,

$$\begin{array}{cccccc}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & -1 \\
1 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0
\end{array}$$

A set of such operators can be defined so that an edge of any orientation can be detected. In addition, [8] also suggested a scheme whereby further unnecessary calculation is avoided by forming intermediate arrays of the sums of appropriate pixels $P_{i,j}$ as shown,

$$\begin{array}{cccccccccc}
P_{11} & P_{12} & P_{13} & P_{14} & P_{15} & P_{16} & P_{17} & P_{18} & P_{19} & \cdots \\
P_{21} & P_{22} & P_{23} & P_{24} & P_{25} & P_{26} & P_{27} & P_{28} & P_{29} & \cdots \\
P_{31} & P_{32} & P_{33} & P_{34} & P_{35} & P_{36} & P_{37} & P_{38} & P_{39} & \cdots \\
P_{41} & P_{42} & P_{43} & P_{44} & P_{45} & P_{46} & P_{47} & P_{48} & P_{49} & \cdots \\
P_{51} & P_{52} & P_{53} & P_{54} & P_{55} & P_{56} & P_{57} & P_{58} & P_{59} & \cdots \\
P_{61} & P_{62} & P_{63} & P_{64} & P_{65} & P_{66} & P_{67} & P_{68} & P_{69} & \cdots \\
P_{71} & P_{72} & P_{73} & P_{74} & P_{75} & P_{76} & P_{77} & P_{78} & P_{79} & \cdots \\
P_{81} & P_{82} & P_{83} & P_{84} & P_{85} & P_{86} & P_{87} & P_{88} & P_{89} & \cdots \\
P_{91} & P_{92} & P_{93} & P_{94} & P_{95} & P_{96} & P_{97} & P_{98} & P_{99} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{array}$$

The final results are easily obtained from addition of pairs of these intermediate array elements. In forming these intermediate arrays, further computation is avoided if the array indexing for the differently orientated templates is pre-computed and stored in look-up tables. The result of combining these techniques is a very fast, scaleable operator that, when scanned over a region of an image, is capable of locating the position of an edge. There still remains the possibility that the edge operator can produce a spurious result, due to the effects of noise in the image or a variation in the road edge itself. In this case, greater consistency can be achieved if the appropriately orientated operator is scanned over a region in a number of channels. The operator results can

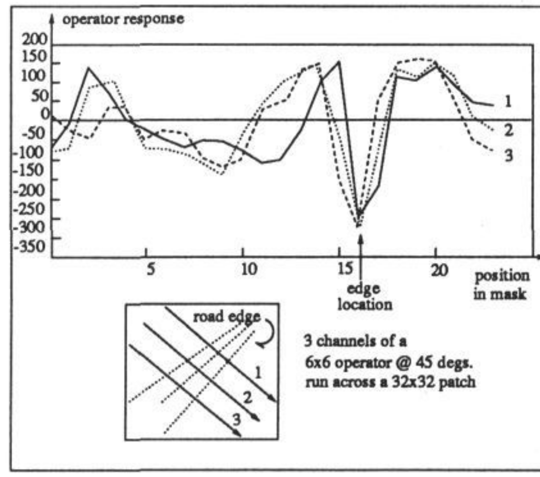


*Figure 3: Correlation in 3-channels and the correlation arrays.*

then be tested for consistency since they should agree on the position offset. Figure 3. shows these channels and the corresponding correlation arrays obtained from scanning the 6x6 operator above across a road edge feature. Since it is important to design an operator that is appropriate for the scale of the edge feature and the image resolution, a number of tests were made on typical road edges in images at three different resolutions. The 6x6 operator shown above was found to be very sensitive to well-defined edge features appearing at 45 degrees in a 256x256 resolution image.

### Using the Edge-Finder Operator

The orientated edge finder has been designed to locate a road edge whose approximate position and orientation in an image are known. This situation arises in the following two cases,

- Register predicted edge positions - where a model of a road has predicted the approximate position of an edge in an image and selects the appropriate operator to accurately locate the edge. In this way, the same edge points are tracked through a sequence of images.

- Locate new edge points from extrapolation - where two or more known edge points are used to extrapolate the approximate position of another edge point. The appropriate mask can be selected from measuring the angle made in the image by these known points. Linear extrapolation to find the $n^{th}$ edge point using the (X,Y,Z) coordinates of the known edge points by,

$$X_n = X_{n-1} + (X_{n-1} - X_{n-2}) \quad (9)$$
$$Z_n = Z_{n-1} + (Z_{n-1} - Z_{n-2}) \quad (10)$$

## WEIGHTED LEAST SQUARES FITTING

The purpose of fitting a curve to the (X,Y,Z) data points is to initialise or provide information to update a model of the road.

## Approximation to a Circular Arc

For the sets of edge points $\{X(i), Z(i)\}_{left}$ and $\{X(j), Z(j)\}_{right}$ it is required to find a pair of circular arcs that are constrained so that the road is a constant width as defined in equations (1),(2) and (3). However, these equations are not linear in the unknown parameters and it is necessary to reduce them to a linear function. To do this, re-arrange and expand as a binomial series:

$$(X - X_0) \;=\; \pm\{R - \frac{(Z - Z_0)^2}{2R} + \cdots\} \qquad (11)$$

the higher order terms in the expansion can be neglected because, in general $R \gg Z$. Re-arrange the last equation into a quadratic form,

$$X \;=\; (X_0 \pm R \mp \frac{Z_0^2}{2R}) \pm \frac{Z_0}{R}Z \mp \frac{1}{2R}Z^2 \qquad (12)$$

and comparing this with $y = a + bx + cx^2$ it can be seen that the parameters $X_0, Z_0$ and $R$ can be determined from the least squares fitting of a quadratic function.

## Weighting Function

In a typical road scene under perspective transformation there is obviously a relationship between the the position accuracy of an edge in the image and physical distance from the vehicle (assuming a constant image resolution). It is possible to include a 'weighting function' into the least squares fitting process to account for this relationship. This is achieved by weighting the original sets of edge points $\{X_i, Z_i\}$ according to the distance from the vehicle $Z_i$. Another reason for introducing a weighting function to bias nearer points is that deviations of the road edge appearence from a circular arc representation will be larger the further from the vehicle. If it assumed that the total measurement error $\Delta x$ of the edger-finder operator, due to such things as noise, camera wobble etc., has a normal distribution $N(0, \sigma^2)$, then the corresponding position error $\Delta X$ in the real world will also have a normal distribution $N(0, f(\sigma^2))$, the variance being scaled by a simplification of the perspective transformation,

$$f(\sigma^2) \;=\; \left(\frac{Z}{kF}\right)^2 \sigma^2 \qquad (13)$$

where k is a normalising constant and F is the focal length of the camera. The coefficient of $\sigma^2$ corresponds to a 'weight' in the least squares fitting process and with appropriate choice of constant the 'weight' for the furthest point of interest at $Z_{max}$ can be normalised to 1, increasing as the distance from the vehicle $Z_i$ decreases.

## Weighted Least Squares Process

In the weighted least squares fitting process, there are a number of measurements $y_i$, in error by a random amount $e_i$ and it is required to minimise the sum of the squared errors,

$$y_i \;=\; a + bx_i + cx_i^2 + e_i \qquad (14)$$

with a weighting function incorporated into the minimisation by scaling the variance of each $e_i$ by the appropriate weight $w_i$ and applying the transformation $e_i' = e_i\sqrt{w_i}$ to the $y_i$'s so that the variances are equal. The corresponding normal equations are obtained by partial differentiation with respect to each parameter and after being set to 0, these are,

$$\begin{array}{llll} a\sum w_i & +b\sum w_i x_i & +c\sum w_i x_i^2 & = \sum w_i y_i \\ a\sum w_i x_i & +b\sum w_i x_i^2 & +c\sum w_i x_i^3 & = \sum w_i x_i y_i \\ a\sum w_i x_i^2 & +b\sum w_i x_i^3 & +c\sum w_i x_i^4 & = \sum w_i x_i^2 y_i \end{array}$$

The solution to this system of linear equations $(a, b, c)$ can be found using Cramer's rule. An estimate of the quality of the fit or the accuracy with which the parameters have been identified can be found from the mean error $\mu$, the 'sum of the residual squares' $S_0$ and the corresponding variance $\sigma^2$, for a quadratic curve given by,

$$\begin{aligned} \mu &= \frac{1}{N}(\sum w_i y_i - a\sum w_i - b\sum w_i x_i - c\sum w_i x_i^2) \\ S_0 &= \sum w_i y_i^2 - a\sum w_i y_i - b\sum w_i x_i y_i - c\sum w_i x_i^2 y_i \\ \sigma^2 &= \frac{S_0}{(N-3)} \end{aligned} \qquad (15)$$

## Filtering Using the Fitted Curve

It has been assumed that the measurement errors have a normal distribution which allows for the identification of 'rogue' edge points in the original data set using the following test,

$$\mid X_i - \mu \mid \;\geq\; 3\sigma \qquad (16)$$

In this way the edge point set can be filtered with 'rogue' points being adjusted to lie on the fitted curve. Further, because 'rogue' edge points would have adversely affected the initial fitting procedure, the whole fitting process can be repeated. This recursive operation can continue until the variance of the residual errors stops decreasing.

# ALGORITHM DESCRIPTION

## Model Initialisation

There are two situations where a road model would need to be initialised.

Firstly, when the vehicle is stationary on the road there is almost unlimited processing time to analyse the scene. In this case, it is most likely that a number of methods would be used to account for failures and resolve ambiguities in finding the road. One such method described in [6] is to apply a Hough transform to a set of edge pixels that have been found from histogramming of the gradient direction, in order to find dominant linear features.

Secondly, whilst the vehicle is moving a situation is encountered that causes the normal mode of operation to fail. In this case, a fast re-initialisation process is proposed. Assuming that one or both of the road edges are still in view, overlap the edge finding operator to produce a very long orientated operator. Then search the resulting correlation array for the 'brightest' n% and attempt to track along these using the extrapolation process.. Possible single tracks or track pairs can then be tested for consistency. Failure of this method would require the vehicle to stop and analyse the full scene.

These proposed initialisation methods have yet to be included and in their absence the initialisation is guided manually by providing the initial coordinates of two (x,y) edges on the left and right road edges. The result of this processing is to produce sets of edge points for the left and right road edges in both the image and world coordinate systems.

### Normal Frame-to-Frame Mode

In subsequent frames, the following sequence of operations are performed on the sets of edge points.Firstly, using the road model;

**vehicle motion correction** - knowledge of the vehicle's forward motion and the algorithm cycle time allow the distance travelled $\Delta Z$ to be calculated. The (X,Y,Z) edge coordinates are then updated. After transformation into (x,y) image coordinates, these then form the predicted locations of the same points in the current frame.

**extrapolating new points** - because the vehicle has moved forward, new portions of the road edge are visible and are found through the (X,Y,Z) extrapolation process described in the previous section. In addition, a constraint on the maximum allowable 'curvature' can be applied to limit the extrapolation process, where the model state in the previous cycle is used to find the maximum.

**filtering old points** - as the vehicle proceeds edge points that are near the vehicle will disappear from view and thus this condition needs to be tested and the edge points filtered.

**register predicted edge positions** - each predicted edge position is used to position the edge finder operator described in the previous section. The correct orientation of this operator is chosen by using the state of the model in the previous cycle. The new (x,y) edge points that have been consistently located using this method are then used in the next stage of updating the model.

and then to update the road model;

**curve fitting in (X,Y,Z)** - after transforming into (X,Y,Z) coordinates, use the weighted least squares procedure to fit a curve to the sets of left and right edge points. As described earlier, the statistics of the residual errors can then be used to identify 'rogue' points and filter the sets. The coefficients obtained in the final fit for each side provide values for $\{Z_0, X_0, R\}_{left}$ and $\{Z_0, X_0, R\}_{right}$. By the earlier definition, the coordinates of the centres of the two circular arcs are the same and the difference in the radii is the road width.

**confidence measure** - a change in the model parameters can be attributed to a combination of a change in the road appearence and/or a change in the accuracy of the fit. In order to allow the model to change to an extent controlled by the quality of the information, a confidence measure is used based upon the standard deviation $\sigma$ of the residual errors in the curve fitting. That is, if the $\sigma$ in the current cycle is less than the previous then use the parameters from the fit. Otherwise make a percentage change in the previous model parameters. Two reasons for a low confidence measure are firstly the uncertainty of the edge data at larger distances from the vehicle and secondly the inadequacy of the simple circular arc representation at large distances. Consequently, a maximum 'look ahead' distance related to the confidence measure is used.

### Failure Recovery

A failure is defined as a low confidence measure from the fitting procedure or a loss of edge points. If the failure is only on one side then the previous state of the model is used to predict the positions in the image of two edge points. If they are found using the edge-finder operator then attempt to track along the extent of the edge using the (X,Y,Z) extrapolation process. If they are not found then revert to initialisation procedure detailed above.

## EXPERIMENTAL RESULTS

The algorithm described was coded in OCCAM to run on the Transputer hardware that has been designed and built at Bristol University [9]. The system comprises of a video interface board that provides digitised video data from camera or VCR to a number of framestores. These are accessible by a number of 20 MHz T414 32-bit Transputers via a VME-like bus. An IBM compatible PC with plug-in B004 is used for program development. A number of sequences of road images have been collected from a video camera mounted on the roof of a car travelling at an average speed of 20 mph around public roads in the Bristol area.

Firstly, the initialisation procedure is manually provided with the initial (x,y) coordinates of two points on the left and the right road edges. These are sufficient for the (X,Y,Z) extrapolation process to track along the extent of the road edges up to a fail condition or a maximum distance of 50 metres. The curve fitting process is then performed to filter the edge points and to initialise the model. If this completes successfully on this frame then the normal frame-to-frame mode begins.

Initial results show the successful tracking of the same edge points in the real world until they disappear from view due to the vehicle moving forward and also the (X,Y,Z) extrapolation process searching for and finding new edge points. The curve fitting process at each stage is monitored by drawing the curve corresponding to the model parameters in the current frame. Figure 4. shows the set of edge points and the fitted curve at a snapshot in the sequence. The algorithm begins to fail when the edge features in the image become indistinct or fragmented and when the road becomes more complex than the representation, for example at a short straight section followed by a sharp curve.

The overall algorithm cycle time is dependant upon the numbers of edge points in the sets, but for a typical value of 15 each side the cycle time is 90 millisec. which corresponds to an update rate of 11 frames/second. The processing of image data by the edge-finder operator is very fast at nearly 2 millisec./ edge point. The transformations and curve fitting take a significant amount of processing time ($\approx$ 30 millisec.) due to the necessity of working with real numbers. This should improve dra-

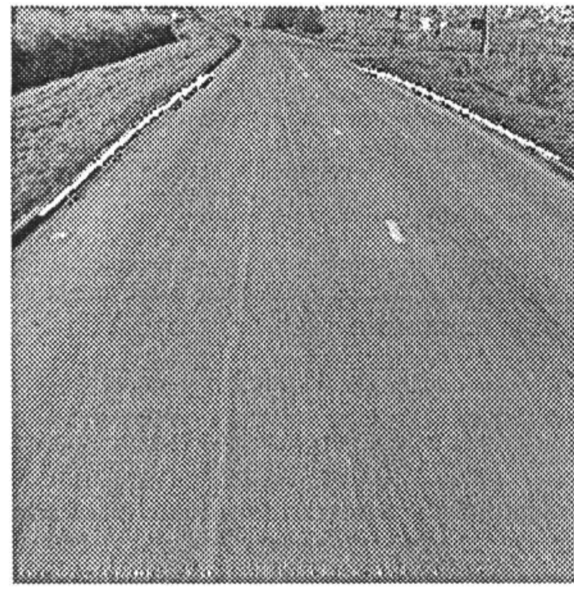

*Figure 4: Tracking of edge points and the result of curve fitting.*

matically when using T800 Transputers. In addition, we have yet to distribute the algorithm amongst a number of processors.

## CONCLUSIONS

Early results show that the use of the circular arc model representation of a road to guide selective image processing is sufficient in the case of simple roads and that the algorithms are fast and stable when the road edges are well defined. Situations arise whereby the model fails, due to inadequacy of the representation or failure of the edge-finder operator in cases where the road edges are ill-defined. On the occasions when edge information is not sufficient or not available, other forms of image processing from work described elsewhere should be used, for example region segmentation.

In order to improve the reliability and range of operation of the edge-tracking approach, planned developments include improvements to the image processing operators, an increase in the complexity of the representation, to identify further heuristics, to implement on a transputer network (using IMS T800's), and test the algorithms in a closed loop using a small tracked vehicle.

### Acknowledgements

This work is supported by MOD RARDE(Chertsey) under contract ER1/9/4/2034/085 to develop Transputer based hardware and algorithms for robot road vehicle guidance.

## REFERENCES

1. **Wallace R., Matsuzaki K., Goto Y., Crisman J., Webb J. & Kanade T.** "First Results in Robot Road Following." *Proc. IJCAI*, Los Angeles, USA, Aug., (1985).

2. **Wallace R., Stentz A., Thorpe C., Moravec H., Whittaker W. & Kanade T.** "Progress in Robot Road Following." *Proc. 1986 IEEE INT. Conf. Robotics & Automation.* (1986).

3. **Thorpe C. & Kanade T.** "1986 Year End Report for Robot Road Following at CMU.",*CMU-RI-TR-87-11*, (1986).

4. **LeMoigne J., Waxman A., Srinivasan B. & Pietikainen M.** "Image Processing for Visual Navigation of Roadways.", *Computer Vision Lab., Univ. Maryland, CAR-TR-138*, July (1985).

5. **Waxman A., LeMoigne J., Davis L.S., Liang E. & Siddalingaiah T.** "A Visual Navigation System." *Proc. IEEE INT. Conf. Robotics & Automation.*,(1986).

6. **Waxman A., LeMoigne J., Davis L.S., Srinivasan B., Kushner T.R., Liang E. & Siddalingaiah T.** "A Visual Navigation System for Autonomous Land Vehicles." *IEEE Journal of Robotics & Automation*, vol. RA-3, No.2, April (1987).

7. **Dickmanns E.D. & Zapp A.** "Guiding Land Vehicles Along Roadways by Computer Vision." *Proc. AFCET, Congres Automatique*, Toulouse, Oct., (1985).

8. **Kuhnert K.** "A Vision System for Real-Time Road and Object Recognition for Vehicle Guidance." *In: Advances in Intelligent Robotic Systems, Proc. SPIE*, vol.727, Cambridge, Mass., Oct., (1986).

9. **Milford D. & Lotufo R.** in preparation, *Bristol University Tecnical Report*, (1988).

10. **Shinar D., McDowell E.D. & Rockwell T.H.** "Eye movements in Curve Negotiation." *Human Factors*, 19(1), pp63-71. (1977).

11. **Ozawa S., Rosenfeld A.** "Synthesis of a Road Image as seen from a Vehicle.", *Pattern Recognition*, 19(2), pp123-145, (1986).

12. **DeMenthon D.** "Inverse Perspective of a Road from a Single Image." *Computer Vision Lab., Univ. Maryland, CAR-TR-210*, July (1986).