Open access • Proceedings Article • DOI:10.5244/C.2.29

# Road Edge Extraction Using a Plan-View Image Transformation. — Source link

Roberto de Alencar Lotufo, Erik L. Dagless, D.J. Milford, BT Thomas

**Published on:** 01 Jan 1988 - Alvey Vision Conference

**Topics:** Image gradient

# ROAD EDGE EXTRACTION USING A PLAN-VIEW IMAGE TRANSFORMATION

R. A. Lotufo E. L. Dagless D. J. Milford B. T. Thomas

Faculty of Engineering, University of Bristol, Bristol, UK

*A new technique to extract road edges in the road-following algorithm for autonomous road vehicle navigation is described. It is based on finding road edges on a subsampled plan-view of a portion of the road ahead of the vehicle. The method is illustrated in the real-time identification of road edges using a fast vertical edge detector and link operator applied to the transformed plan view. Location of both road edges at 20 frames per second is demonstrated.*

Research on autonomous navigation of robot vehicles has been increasing in the last few years [1,2,3]. Part of this research consists of identifying the road from a digitised image received by a camera positioned on the robot vehicle.

The problem of identifying roads to drive an autonomous vehicle is a challenge due to its real-time characteristic. To achieve real-time processing, it is crucial to reduce the amount of data analysed by the system. A solution which processes only small windows placed over the predicted road edges [1] has the disadvantage of lack of global information on the road scene. This paper describes a new method of reducing the amount of data while preserving the global features of the road. It is based on generating a subsampled plan-view of the portion of interest on the original road image. The transformation is performed using the concept of a Vanishing Point [4], making the road edges nearly vertical in the plan-view even in a curving road.

Section 2 describes how the plan-view image is generated comparing some aspects of the pre-filtering of the road image. In section 3, we describe the edge detector, segment linking and sorting of the edge magnitudes. In section 4, details of the implementation and some real-time performance figures for the algorithms are reported.

The objective in the present paper is to introduce the plan-view transformation method and to demonstrate its application in the real-time road identification problem.

## IMAGE TRANSFORMATION

In order to build an image of the road as seen in plan-view, a transformation is defined that maps the road coordinates into the digitised image pixels.

These transformation equations are determined using a linear transformation to change the road coordinate system to the camera coordinate system followed by a non linear perspective transformation related to the image formation inside the camera.

Figure 1 shows the road coordinate system defined by the axis $X_r$, $Y_r$ and $Z_r$ with the road on the plane $X_r$ $Z_r$ and the robot vehicle moving in the $Z_r$ direction. The camera is located above the vehicle with its optical centre at a height $H$ above the road with a tilt angle $\tau$. The camera coordinate system is represented by $X_c$, $Y_c$ and $Z_c$. The equations which relate a point in the road system ($x_r$, $y_r$, $z_r$) to a point in the camera coordinate system ($x_c$, $y_c$, $z_c$) are obtained from a translation of $H$ in the $Y_r$ direction and a rotation of $\tau$ on the plane $Y_r$ $Z_r$:

$$\left\{ \begin{array}{lcl} x_c &=& x_r \\ y_c &=& (y_r - H)\cos\tau + z_r \sin\tau \\ z_c &=& -(y_r - H)\sin\tau + z_r\cos\tau \end{array} \right. \qquad (1)$$

We use a pin-hole model of the camera with a focal length $f$ to define the perspective transformation. Figure 2 shows a point in the camera coordinates system ($x_c$, $y_c$, $z_c$) projected onto the image sensors ($x_i, y_i$). As the $z_c$ coordinate is measured from the optical centre of the lens, the perspective projection is given by the equations:

$$\left\{ \begin{array}{lcl} x_i &=& -\frac{f x_c}{z_c} \\ y_i &=& -\frac{f y_c}{z_c} \end{array} \right. \qquad (2)$$

By combining (1) and (2) we can write the equations to transform points on the plane of the road, $y_r = 0$, in terms of raster image coordinates $x$ and $y$:

$$\left\{ \begin{array}{lcl} x &=& \frac{X_{size}}{2} - S_x \frac{f x_r}{(H\sin\tau + z_r\cos\tau)} \\ y &=& \frac{Y_{size}}{2} - S_y \frac{f(z_r\sin\tau - H\cos\tau}{(H\sin\tau + z_r\cos\tau)} \end{array} \right. \qquad (3)$$

where $X_{size}, Y_{size}$ are the dimensions of the image in pixel units and $S_x, S_y$ are the conversion factors in pixels per metre in each coordinate.

AVC 1988 doi:10.5244/C.2.29

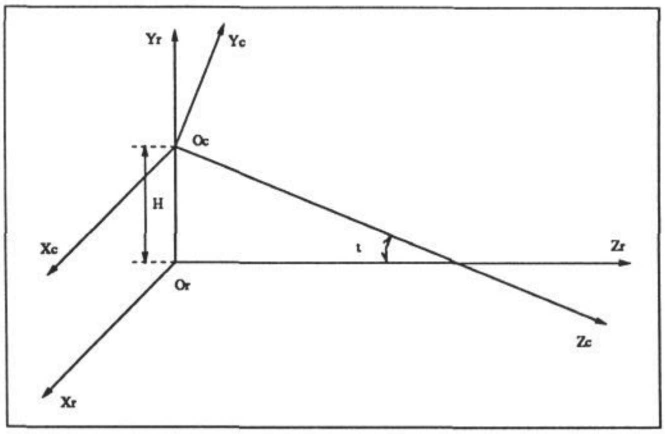


*Figure 1: Two world coordinates system: $(X_r\ Y_r\ Z_r)$: road system, $(X_c\ Y_c\ Z_c)$: camera system. The road is assumed to be on $X_r\ Z_r$ plane, i. e. $Y_r = 0$. $H$ is the height of optical centre of the camera lens from the ground. $\tau$ is the tilt angle of the camera.*

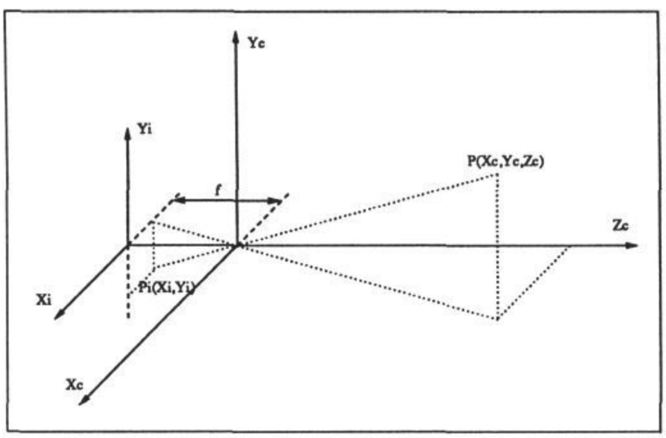


*Figure 2: Perspective transformation. Pin-hole model of the camera. $f$: focal length, $(x_i, y_i)$: image plane.*

## Generating a plan-view image

The method consists of compressing the raw image while preserving all the main road features of interest in the field of view, making each pixel have uniform physical dimensions and normalising the road edges to a vertical direction or near vertical direction in the case of bending roads.

These issues benefit any later computation both in terms of speed and in simplifying the algorithms by limiting the range of inclination of the road edges. Two examples of these algorithms are a gradient edge detector and edge linking operators discussed later.

The plan-view normalised image of the road is generated by applying the transformation (3) on a grid of points over the road plane. Figure 3 shows the boundaries of the grid which are limited by $Z_{r_{near}}$ and $Z_{r_{far}}$ in the direction of the vehicle movement, by $X_{r_{LEFT_{near}}}$ and $X_{r_{RIGHT_{near}}}$ in the lateral distances and inclined by $\theta$. $\theta$ is chosen in such a way that the road edges are nearly parallel to the lateral boundaries of the grid. This is

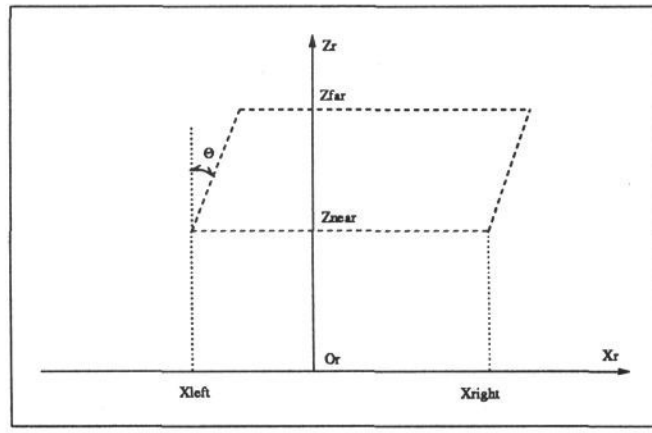


*Figure 3: Definition of the grid boundaries on the road plane $X_r\ Z_r$ where the transformation is applied*

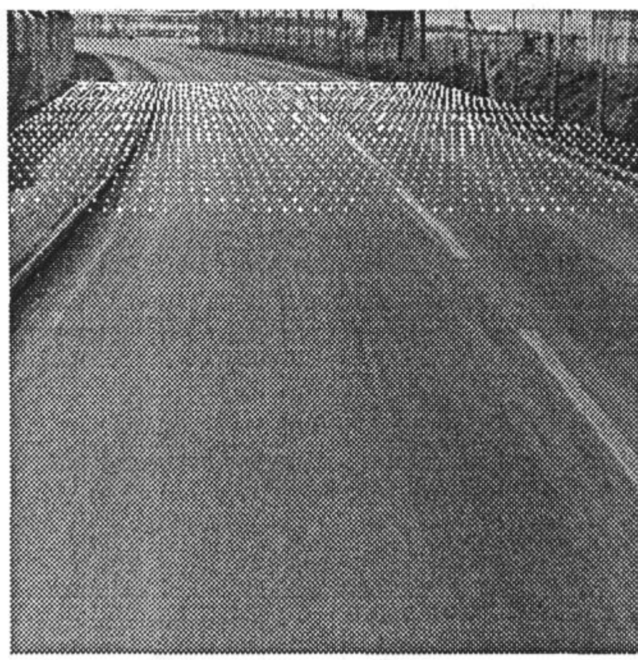

*Figure 4: original image (256 x 256, 256 grey levels), showing the grid of 64 x 32 points plotted on the road. Definition of the boundaries of the grid: $Z_{r_{near}} = 10.0m$; $Z_{r_{far}} = 30.0m$; $X_{r_{LEFT_{near}}} = -4.5m$; $X_{r_{RIGHT_{near}}} = 6.0m$; $\theta = -8^o$.*

performed automatically by selecting a pan angle, from a number of precomputed values, which gives a preponderance of near-vertical edges.

In figure 4 this grid is plotted on a road scene digitised to 256 x 256 pixels with 256 grey levels. The camera parameters have been calibrated manually and set to: focus $f$ = 50mm, height $H$ = 2.05m, tilt $\tau = 18.0^o$, and the conversion factors $S_x = S_y$ = 7900 pixels/m. The grid boundaries parameters are $Z_{r_{near}}$ = 10.0m ; $Z_{r_{far}}$ = 30.0m ; $X_{r_{LEFT_{near}}}$ = −4.5m; $X_{r_{RIGHT_{near}}}$ = 6.0m and $\theta = -8^o$.

The plan-view image formed is as shown in figure 5. Its dimensions are 64 pixels in the $X_r$ direction and 32 pixels in $Z_r$ direction, reducing the raw image data by a factor of 32. The lateral dimension of the image is greater than

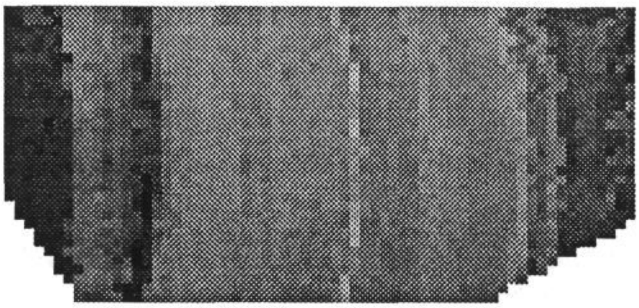

*Figure 5: Plan-view image processed using the grid plotted on figure 4: $\theta = -8^o$, 64 x 32 pixels.*

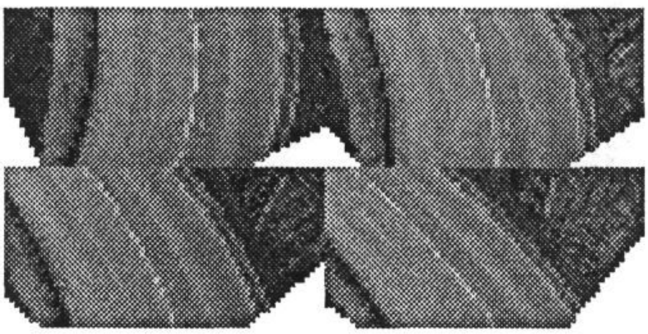

*Figure 6: Plan-view images processed by varying the angle $\theta$ on figure 4: $\theta = -10.0^o$ (top left), $\theta = -5.0^o$ (top right), $\theta = 0.0^o$ (bottom left), $\theta = 5.0^o$ (bottom right).*

the longitudinal direction giving a higher resolution in that direction in order to make the extraction of the nearly vertical road edges more accurate.

To illustrate the effect of varying the pan angle $\theta$, several plan-view images are, as shown in figure 6, built up from the same road image (figure 4), with the same boundary parameters and $\theta = -10.0^o$ (top left), $\theta = -5.0^o$ (top right), $\theta = 0.0^o$ (bottom left) and $\theta = 5.0^o$ (bottom right) .

The image is generating by subsampling the individual pixels specified by the transformation without any smoothing so the resultant image is slightly noisy. To study the effect of this noise on the edge detector operator applied later, two methods of building the image are compared to the sampled image method described.

First a gaussian smoothing filter is applied to the raw image in the neighbourhood of the points selected by the grid [5]. As the sampling distance in horizontal direction is different from the vertical, the following gaussian operator is used:

$$g(x, y, \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2})} \qquad (4)$$

where $\sigma_x$ and $\sigma_y$ are equal to half of the horizontal and vertical sampling distances respectively.

The second method of smoothing was a straightforward

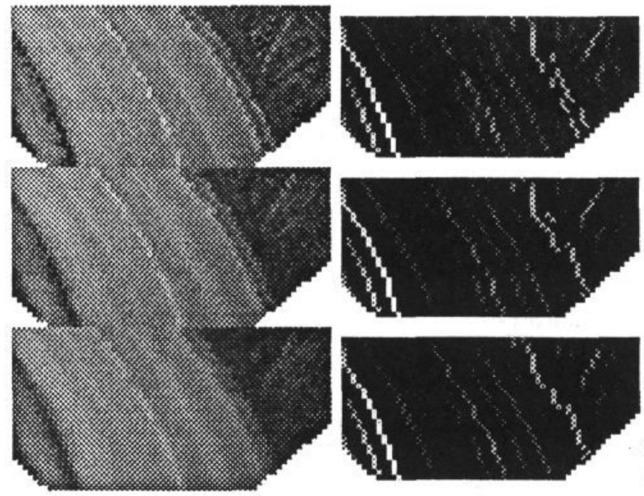

*Figure 7: Comparison of smoothing methods used to compute the plan-view image: top left, without smoothing; centre left, rectangle averaging; bottom left, gaussian smoothing. The corresponding edge detector output is on the right.*

pixel averaging of a rectangular area in the raw image around the selected point. The horizontal and vertical dimensions of the rectangle were the corresponding sampling distances.

The methods are compared visually in figure 7. It shows three processed images of the same part of the road image applying the gaussian smoothing (bottom left), averaging smoothing (centre left) and without smoothing (top left). The respective vertical edge detector outputs are shown on the right of the figure and are described in a later section.

Although the gaussian smoothing shows the best noise figure of the three methods, the unfiltered image is used to achieve real-time processing since the edge detector output still gives good results.

In this paper it is assumed that the road is locally plane and the camera is calibrated, i.e. the parameters $f, S_x, S_y, H$ and $\tau$ are known and fixed during the experiments. In a non-level road surface or in case of pitching of the camera platform, the road edges will appear as non-parallel edges in the plan-view image. It is possible to compensate for these effects by determining the new $\tau$ from the angle between the transformed road edges.

## EXTRACTING ROAD EDGES

The process of extracting road edges from the plan-view road image involves several steps: convolving the image with a vertical edge mask, thinning the edges retaining magnitude values, linking them and finally selecting a set of the strongest edge segments among all segments detected. This method is based on one proposed by Nevatia and Babu [6] with important simplifications due to the characteristics of the image. The process does not involve the choice of critical thresholding parameters as is usual in most edge detection schemes.

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \end{bmatrix}$$

*Figure 8: Prewitt 3x3, Sobel and '5 x 5 difference of boxes' masks*

### The edge detector mask

The selection of the edge detector mask is difficult and crucial. The literature on edge detectors is very large and their performance is very dependent on the features to be detected in the picture. Comparison of edge masks is hard and can be misleading when different features are considered.

Figure 8 shows three different vertical edge detectors: the Prewit 3x3 on the left, Sobel in the centre and a '5 x 5 difference of boxes' on the right.

The performance of these masks are compared for a series of processed plan-view images with the edges nearly vertical and also slightly inclined. The results showing the input images and the outputs of the edge operators after thinning are in figure 9. From top to bottom, they are the Prewitt 3x3, the Sobel 3x3 and the '5x5 difference of boxes'.

The results confirm that although the plan-view image is noisy due the fact it was generated by subsampling the raw image without smoothing, the results of the vertical edge detectors are reliable and the strongest intensities correspond to the road edges and lane markings.

The '5x5 difference of boxes' mask was chosen because its outputs are easier to link in the subsequent processing. Due to its low pass spatial frequency characteristic, the number of edges is smaller and the edges are farther apart making the search process in the linking phase trivial.

### Thinning the edges

After the vertical mask convolution the edges are thinned to extract the position of each edge. One of the simplest ways of doing this is to suppress non-maxima in the direction of the gradient. As the processing of the image only has to identify vertical edges the suppression is performed by comparing the two neighbours in the horizontal direction and if either has a higher value, its edge output is zeroed. If there are multiple edges with the same value, the left one is selected. This simplification has presented no problems. As the direction of the gradient is constant, the thinning can be combined with the convolution process in a left to right scanning of the image thereby improving the speed of the edge operator.

Normally the thinning process involves thresholding to

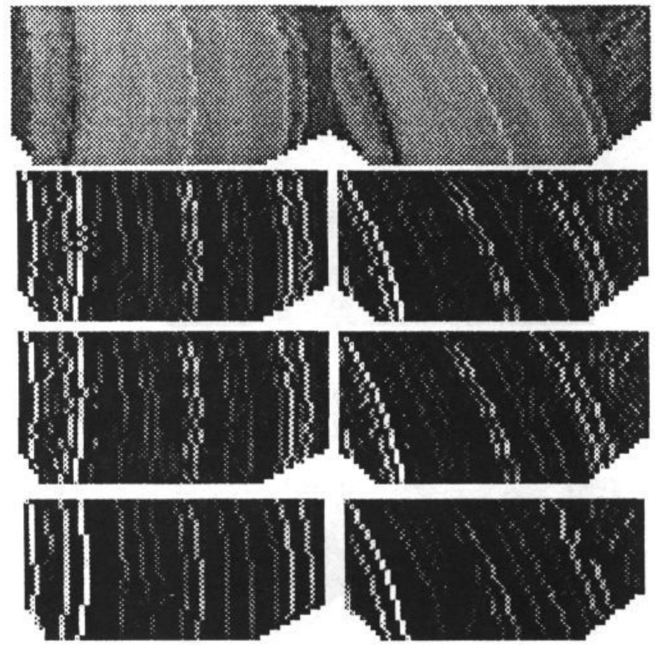

Figure 9: Comparison of edge detectors outputs (after thinning). From top to bottom: the plan-view road image (left hand side $\theta = -8^o$; right hand side, $\theta = 0^o$), the Prewitt 3x3, Sobel 3x3 and '5x5 difference of boxes' edge detectors.

eliminate unnecessary details or noise in the image: Canny [5] uses an adaptive threshold whilst Nevatia and Babu [6] use a fixed threshold. In our method no threshold parameter is used. The details are suppressed by applying a sorting process after the edge linking operator in order to find the strongest edges. The higher level process, responsible for the interpretation of the road boundaries, requests a small subset of the strongest segments to be analysed. We have found this approach to be very robust against the common problems in road edge extraction such as different types of road edges, shadow, variations on the illumination and surface conditions, etc. .

### Segment linking

Linking edges can be a complex task when the image contains edges in all directions. The choice of predecessors and successors of each edge can result in many options generating some faulty linked segments.

In this case the task of linking segments is easier as the edges are mainly vertical and have a clear gap between them. The search process looks for its successor among the three neighbouring cells in the row below. If more than one is found, a situation which is unlikely due to the large vertical mask used in the convolution step, the edge with the nearest magnitude value is chosen.

The search process computes the mean of the segment magnitudes, the length and the position of each segment. The linking process is performed in one scan of the image and the output is a list of all edge segments with the three parameters above.

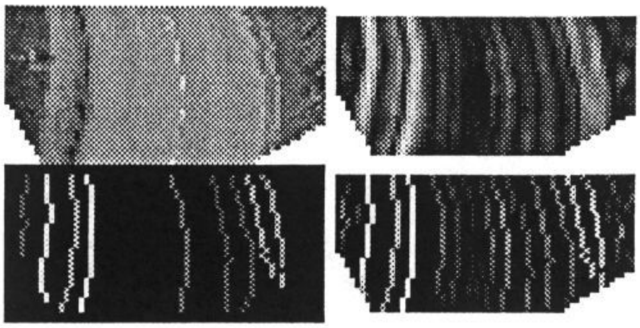

*Figure 10: top left: the plan-view road image, top rigth: edge detector output, bottom rigth: thinned edges, bottom left: ten strongest linked segments*

The processes of edge detection, thinning and segment linking is shown in figure 10. Clockwise starting from top left: plan-view image, '5 x 5 difference of boxes' vertical edge detector magnitudes, thinned edges without thresholding, and ten strongest linked segments.

### Extracting Road Edges

Extracting the road edges from the segment list is a complex problem. However, to aid evaluation, a simple technique was used to analyse the outputs of the edge and linking algorithms in real time. Assuming the road is well defined, plane, with parallel edges and obstacle free, then it can be identified by any strong parallel edge segments separated by its pre-established width.

The edge segments are sorted and the ten strongest are selected. Segments near the middle of the road are assumed to be lane markings and are rejected. For the remaining edges an average is performed of the left and right segments respectively. This is not an ideal grouping method but is used at present simply to illustrate the overall technique. In the final implementation we will be using a higher level model to assist in the selection process.

### Choosing the pan angle $\theta$

To obtain nearly vertical road edges in the plan-view image, the pan angle, $\theta$, must be carefully chosen. As the processing time is fast compared to the speed of the vehicle, the predicted pan angle is determined from the results of processing the previous image. An average inclination is measured on both sides using the strongest segments. This new inclination is added to the current pan angle and the nearest precomputed value is selected to correct the transformation for the next frame.

The process described works on a well defined curving road, but it is sensitive to obstacles near the edges of the road like trees and people. This arises because as the obstacle appears in the view of interest, its edges are detected and if stronger than the road edges, a pan angle is selected that transforms the obstacle edges to the vertical and so makes them stronger.

A way to overcome this problem is to process, simultaneously, more than one transformed image each with a different pan angles, thereby providing more information for the road identification process.

## IMPLEMENTATION AND PROCESSING TIMES

The algorithms have been implemented in OCCAM on our own designed Inmos transputer (T414) based image grabber system using a single 20MHz transputer running with a 150 ns memory cycle. The 256 x 256 x 8 bits image is mapped into the transputer memory in a double buffer frame store.

The plan-view image generation is implemented using precomputed pixel coordinates referenced to the raw image. A set of 24 different pan angles, $\theta$, is stored with $\theta$ varying linearly for $18^o$ each side of the centre line giving a discrete step of $\delta\theta = 1.5^o$.

The vertical convolution and thinning are performed in the same scan of the plan-view image. The convolution is implemented keeping a temporary array with the sum of 5 rows updated as each line is scanned. This method can be applied due the simplicity of the 5x5 vertical edge detector chosen.

The linking process is implemented as described earlier and the segment sorted using the bubble sort algorithm.

The computation times for each step are as follows:

| | |
|---|---|
| - plan-view image computing: | 8.3 ms |
| - edge detector and thinning: | 19.0 ms |
| - linking segments: | 17.9 ms |
| - sorting ten strongest segments | 3.2 ms |

giving a total processing time of under 50 ms or less than three TV fields in a 50 Hz system.

Figure 11 shows a hardcopy of two screens of the transputer based real-time image processing system. The twenty four precomputed pan angles are represented at the top as a row of white dots with the chosen pan angle being indicated by the larger mark. Each screen shows the original image with the various processing stages displayed below. Inset in the centre is the plan-view image, bottom left is the edge detector output, and the ten strongest linked segments on the bottom right. The predicted road edges are displayed by two dotted lines superimposed on the original image.

## CONCLUSION

The main purpose of this paper is to describe a new low level image processing technique which plays an important role in solving the real-time road identification problem in visual navigation of an autonomous vehicle. The method consists of computing a reduced plan-view image of the portion of the road in the middle distance ahead of the vehicle. The plan-view image has the

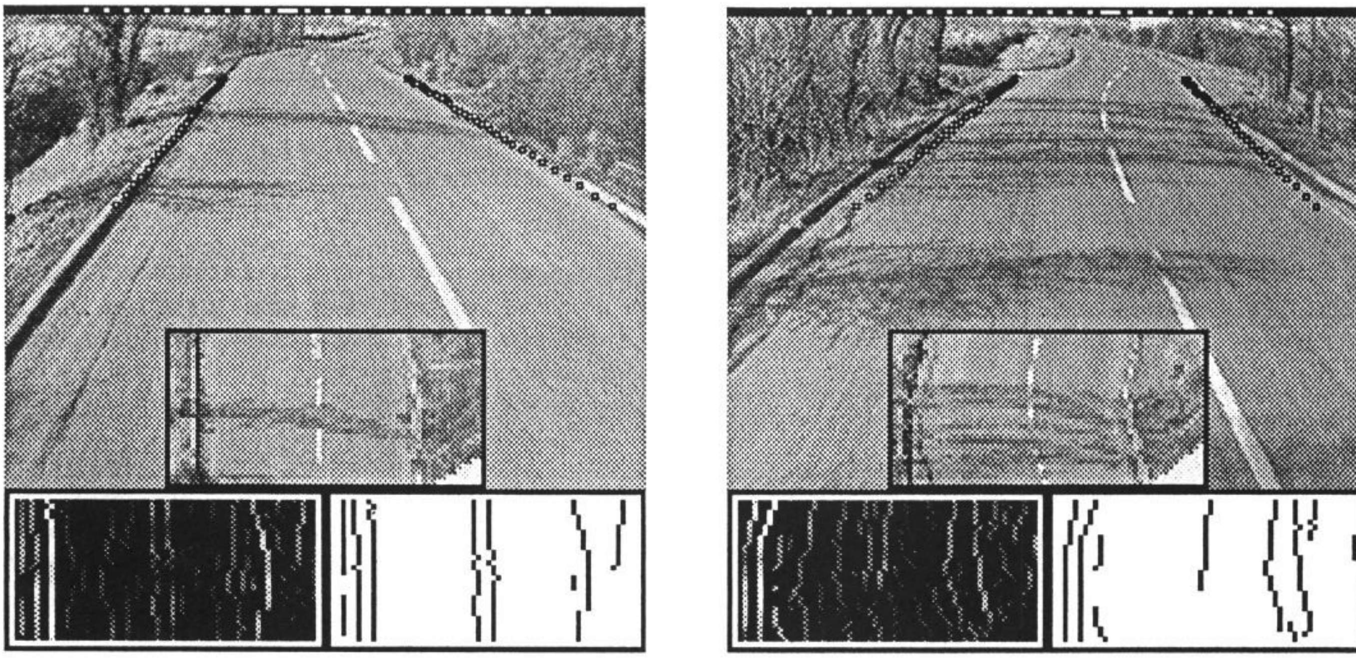

*Figure 11: Hardcopy of two screens each showing the original image and the intermediate steps in the road edge extraction process.*

following characteristics which improve later processing stages in the road identification problem:

- reduced size of data while preserving the main features of the road,
- road edges nearly vertical, and
- pixels having uniform physical dimensions.

The technique is illustrated in the real-time identification of real road edges using a fast vertical edge detector and link operator applied to the transformed plan-view image. Location of both road edges at 20 frames per second is demonstrated.

The results of the method seem very promising for autonomous road vehicle navigation. The planned future work using the plan-view transformation consists of processing many pan angle views in parallel, extracting area segmentation information from the plan-view and studying obstacle avoidance using successive plan-view frames.

## ACKNOWLEDGEMENTS

R. A. Lotufo is supported in part by the Conselho Nacional de Desenvolvimento Cientifico e Tecnologico - CNPq, Brazil, under grant n.200172/86-EE, and in part by the Universidade Estadual de Campinas - Unicamp, Brazil. This work has been funded by RARDE (Chertsey) under MoD contract ER1/9/4/2034/085.

## REFERENCES

1. **Mysliwetz, B. and Dickmanns, E.D.** "A vision system with active gaze control for real-time interpretation of well structured dynamic scenes", *Proc. Intelligent Autonomous Systems*, Amsterdam, (1986).
2. **Wallace, R., Matsuzaki, K., Goto, Y.,Crisman, J., Webb , J. and Kanade, T.** "Progress in robot road-following", *Proc. 1986 IEEE Int. Conf. Robotics and Automation* (1986).
3. **Waxman, A.M., Le Moigne, J.,Davis, L.S.,Liang,E. and Siddalingaiah, T.** "A visual navigation system", *Proc. 1986 IEEE Int. Conf. Robotics and Automation* (1986).
4. **Liou, S.P. and Jain, R.C.**"Road following using vanishing points", *Computer Vision, Graphics and Image Processing*, vol. 39 (1987) pp. 116-130.
5. **Canny, J.F.** *Finding edges and lines in images* Massachusetts Inst. Technol., AI Lab. Tech. Rep. 720, (1983).
6. **Nevatia,R. and Babu, K.R.** "Linear feature extraction and description", *Comput. Graphics Image Process.* vol. 13 (1980) pp.257-269.