Lane Boundary Tracking for an Autonomous Road Vehicle

N. W. Campbell and B.T. Thomas
Advanced Computer Research Centre
University of Bristol
Bristol, UK

Abstract

We describe an algorithm by which an autonomous land vehicle is able to navigate along roads utilising lane boundary markings. This is achieved by defining a six-parameter model of the lane markings and fitting this to the processed monochrome image using non-linear least squares techniques. By qualitative as well as quantitative analysis over many hundreds of images the model used is shown to be justified, resulting in a very robust method, finding lane markings to sub-pixel accuracy.

1 Introduction

There has been a great deal of interest in autonomous road vehicles especially in the last decade, during which real-time image processing has become a reality. Road following is better defined than the more difficult problem of general terrain navigation, but is still extremely complex. There are several approaches to the problem, including:

- Detecting and tracking road-edges.
- Road surface segmentation.

Both of these suffer difficulties. Edge tracking [1] may fail for many reasons, including loss of road edge due to occlusion by other vehicles, its disappearance at junctions or its moving when the road widens or narrows. Road surface segmentation [2] is often unreliable because shadows, other vehicles and changes in surface texture may cause mis-classification.

It is our belief that any effective system will not only need to use several of the above techniques in parallel, but that it will be greatly enhanced by the little-used but very powerful cue of lane markings.

In [3] lane markings are detected in a 'bootstrap' mode using a high level vision approach. Candidate segments are extracted from the image based on their local geometric properties having a high probability of belonging to a lane marking. These segments are then grouped globally to identify possible lane markings.

However approaches such as this are often not robust enough to be relied upon by a road-following vehicle since the local segmentation process is subject to noise and the global grouping has little knowledge of the overall structure of lane markings. Therefore other white objects in the scene i.e. window frames, lettering on signposts or cars may cause mis-tracking to occur.

BMVC 1992 doi:10.5244/C.6.17

The approach taken here is to define a model powerful enough to be capable of encapsulating the inherent structure of the lane marking, including its regular mark:space ratio and geometry. The model is then matched to the image and updated in a frame to frame manner using a least squares technique to overcome any problems of noise and to increase robustness.

The theory of the non-linear least squares process is described next, followed by a discussion of the model used and the results obtained using this technique.

2 Theory of Non-Linear Least Squares

A two-dimensional data set $I_{(X,Z)}$ may be modelled using the function $S_{(X,Z)}(\underline{\mathbf{a}})$, where $\underline{\mathbf{a}}$ is a set of parameters $a_0 \ldots a_{n-1}$. This n parameter model need not give an exact reconstruction of the original data set but, in general, it should be true that for all points:

$$S_{(X,Z)}(\underline{\mathbf{a}}) - I_{(X,Z)} \approx 0 \tag{1}$$

If we have an approximation $\underline{\mathbf{a}}^k$ to the required vector of parameters $\underline{\mathbf{a}}$, then we obtain a better estimate $\underline{\mathbf{a}}^{k+1}$ using:

$$\underline{\mathbf{a}}^{k+1} = \underline{\mathbf{a}}^k + \delta\underline{\mathbf{a}} \tag{2}$$

To determine the $\delta\underline{\mathbf{a}}$ increment in general requires solving a set of non-linear equations. This is avoided by expanding $S_{(X,Z)}(\underline{\mathbf{a}})$ as a Taylor series about $\underline{\mathbf{a}}$ and neglecting all terms which are not linear in δa_i. Hence $\delta\underline{\mathbf{a}}$ can be found from:

$$\underline{\mathbf{A}}.\delta\underline{\mathbf{a}} = \underline{\mathbf{b}} \tag{3}$$

where the normal matrix $\underline{\mathbf{A}}$ is assumed non-singular and defined by:

$$\underline{\mathbf{A}}_{ij} = \sum_{X,Z} \left(\frac{\partial S_{(X,Z)}}{\partial a_i} . \frac{\partial S_{(X,Z)}}{\partial a_j} \right) \tag{4}$$

and

$$\underline{\mathbf{b}}_i = \sum_{X,Z} \left(\frac{\partial S_{(X,Z)}}{\partial a_i} [I_{(X,Z)} - S_{(X,Z)}(\underline{\mathbf{a}})] \right) \tag{5}$$

Now that $\underline{\mathbf{a}}^{k+1}$ has been calculated, it is refined iteratively using equation (2), the process being repeated until convergence occurs with the desired values $\underline{\mathbf{a}}$. This algorithm, known as the Gauss-Newton iterative technique, is a generalised version of the one-dimensional Newton-Raphson. It is suitable for any system in which the model $S_{(X,Z)}(\underline{\mathbf{a}})$ is smooth and continuously differentiable and for which it is possible to obtain an initial guess 'sufficiently close' to the solution $\underline{\mathbf{a}}$ [4].

At convergence, a quantitative measure of the accuracy of the fit over m observations can be obtained from the standard deviations of the n parameters. These are given by the following formula, involving the leading diagonal of the inverse of the normal matrix $\underline{\mathbf{A}}$:

$$\sigma^2(a_i) = \underline{\mathbf{A}}_{ii}^{-1} \frac{\sum [I_{(X,Z)} - S_{(X,Z)}(\underline{\mathbf{a}})]^2}{(m-n)} \tag{6}$$

3 The Model

Although the number of parameters in our model is large enough to accurately model the marking uniquely, it was necessary to keep the number down to reduce processing time. For this reason the model is of a binary, overhead view of the road as shown in Figure 3, so that perspective and grey level effects can be ignored.

Another factor in choosing our model is that, as mentioned above, the function should be differentiable and this prevents the use of step-edge functions.

The model is split into two parts each having three parameters. The first defines the spatial location and curvature of the marking on the road in the form of a quadratic term similar to that used in [5] to model a road edge. The shape of the lane marking in cross-section, as shown in Figure 1, is approximated by a sech() term. The long 'tail' on this function will be shown later to be necessary during convergence. The second term models the size and spacing of the lane markings as shown in Figure 2. In plan view (co-ordinates X, Z) a plot of lane marking intensity vs. distance from the camera may be considered as a square-wave having a period $\mathcal{P}$, duty cycle d and phase shift θ.

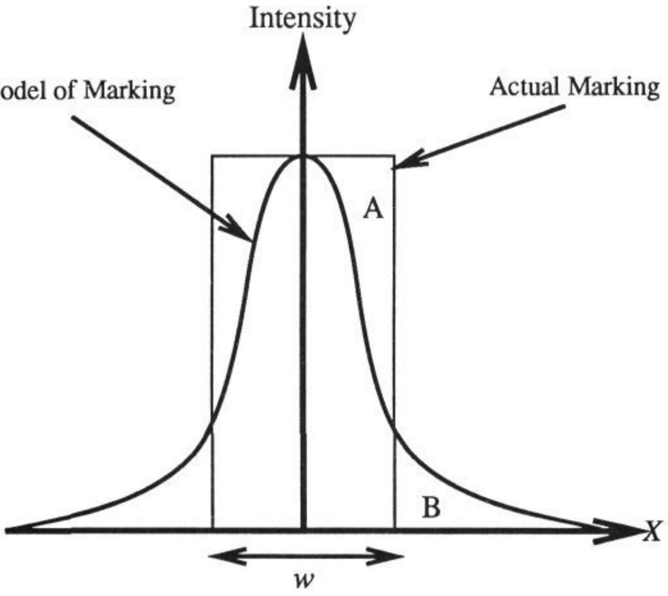

Figure 1: Plot of $\text{sech}(K_s X)$, which Models Cross-Section of a Lane Marking

An example plot of the model is shown in Figure 3 and the model itself is given by:

$$S_{(X,Z)}(\underline{\mathbf{a}}) = \mathcal{C}_{(X,Z)}(a_0, a_1, a_2)\mathcal{T}_{(Z)}(a_3, a_4, a_5) \tag{7}$$

where

$$\mathcal{C}_{(X,Z)}(a_0, a_1, a_2) = \text{sech}\left(K_s\left[X - a_0 - a_1 Z - a_2 Z^2\right]\right) \tag{8}$$

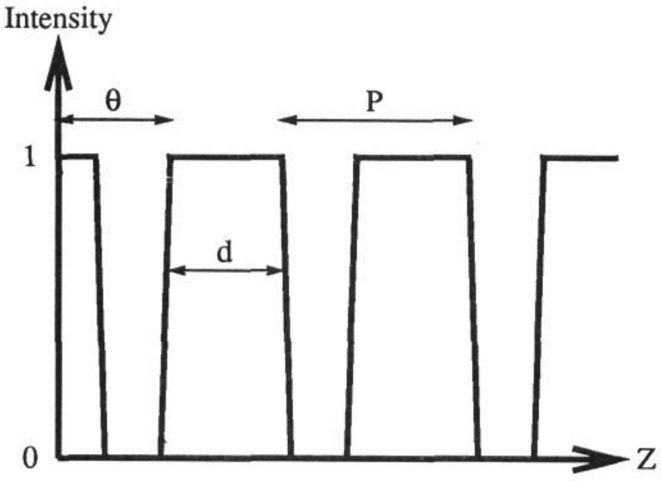

Figure 2: Square Wave Model: Plot of $\mathcal{T}_{(z)}(\theta, d, \mathcal{P})$

and

$$\mathcal{T}_{(z)}(\theta, d, \mathcal{P}) = \tfrac{1}{2}\tanh\left(K_t\left[\operatorname{asin}(\cos(\frac{2\pi Z}{\mathcal{P}} - \theta - \pi d)) + \pi(d - \tfrac{1}{2})\right]\right) + \tfrac{1}{2} \quad (9)$$

Note that although we use parameters $a_0 \dots a_5$ it is more meaningful if we use $a_3 \equiv \theta$, $a_4 \equiv d$ and $a_5 \equiv \mathcal{P}$ in equation (9).

The constant K_s in equation (8) relates to the width of a lane marking. We require that the errors incurred by using the $\operatorname{sech}(K_s X)$ term to model the lane marking be the same inside and out, i.e. Area A equals Area B in Figure 1. Equating these two areas gives:

$$\frac{w}{2} - \int_0^{w/2} \operatorname{sech}(K_s X)\, dX = \int_{w/2}^{\infty} \operatorname{sech}(K_s X)\, dX \quad (10)$$

where w is the width of the lane markings in the plan view. This leads to the result: $K_s = \frac{\pi}{w}$. For the images shown here $K_s = 1$ has proven to be satisfactory.

Equation (9) models the square wave shown in Figure 2. The tanh() term models a mark/space boundary in a continuous manner and the asin(cos()) term makes these boundaries periodic. K_t is the constant which changes the slope of the edge of the square-wave so as $K_t \to \infty$ the model more closely resembles a true square-wave. However, if this parameter is too large then the continuous differentiability assumption is violated. A trade-off is required and in current versions of the algorithm $K_t \approx 50$.

4 Processing

The steps required in pre-processing the image into a form suitable for fitting with the model are described here.

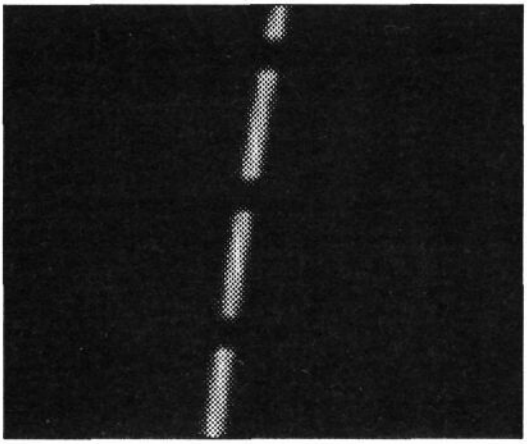

Figure 3: Model of the road scene $S_{(X,Z)}(\underline{a})$.

4.1 Plan Transform

Our model of lane markings assumes that the mark:space ratio is constant throughout the image. This is only true when considering the road in plan view since perspective effects are removed [6]. An image (x, y) may be transformed to a 'birds eye' view (X, Z) with knowledge of the camera geometry using:

$$Z = \frac{h(F_y + y\tan\alpha)}{F_y\tan\alpha - y} \qquad X = \frac{(h\sin\alpha + Z\cos\alpha)x}{F_x} \tag{11}$$

where h is the height of the camera above the road, F_y and F_x are the focal lengths of the camera lens and α is the tilt angle of the camera. This assumes that the tilt angle α remains constant (or is known) for each frame, and that the scene being transformed is locally flat.

4.2 Edge Detection

To fit our model to the image it is necessary that the image be in a similar form i.e. pixel values in $[0, 1]$, where a value of 0 represents road pixels and a value of 1 represents lane markings. Simple thresholding techniques prove too unreliable to achieve this, especially in areas of shadow or where the lane marking becomes obscured by tyre marks or surface dirt. Therefore the image is convolved with an edge detector and the values obtained appropriately scaled so that a typical marking/road boundary has the value 1 and others have smaller values. This method copes well with most shadows since the *magnitude* of edges are similar in shadow or out. The algorithm has worked satisfactorily using many different edge detectors including Canny and Sobel.

A morphological dilate/erode algorithm is now applied so that any gaps between edges, or those caused by noise, are removed. Figure 5 shows the plan-view transformed, pre-processed result of Figure 4.

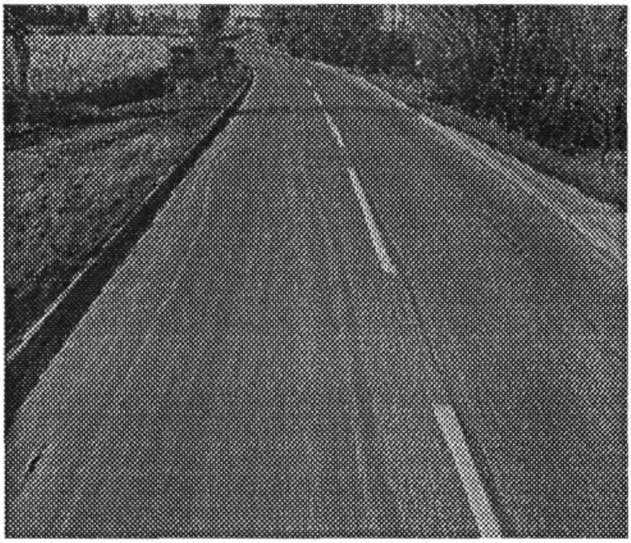

Figure 4: A Typical Road Scene.

4.3 Solution

A bootstrap algorithm as described in Section 1 obtained initial values for the model at the beginning of every sequence. Then for each frame to be processed the previous frame provided an initial guess of the six parameters in $\underline{\mathbf{a}}$. In general, however, this guess was not good enough to give reliable, fast convergence. Any local minima between the initial guess and the global minimum in the 6-dimensional search space lead to an incorrect solution being generated. In practice the approach of updating all six parameters at once failed for this reason. Therefore, as stated in section 2, it is vital that our initial guess be improved before this approach is taken.

In our six parameter model the only parameter almost certain to change from frame to frame is θ, and so it is necessary to estimate this as accurately as possible before any attempt at minimising the other parameters is made. This is achieved by keeping all the parameters of our initial guess, $\underline{\mathbf{a}}^k$ say, constant expect for θ and minimising:

$$\sum_{X,Z} \left[I_{(X,Z)} - S_{(X,Z)}(\underline{\mathbf{a}}^k(\theta)) \right]^2 \tag{12}$$

as a function of θ. This new value is used for the parameter a_3 in our refined guess $\underline{\mathbf{a}}^{k+1}$.

Now that we have a set of parameters very close to the correct answer, we can update all six parameters using the Gauss-Newton technique to obtain our final answer. By taking advantage of knowledge of the form of the function $S_{(X,Z)}(\underline{\mathbf{a}})$, convergence was achieved reliably without the need to resort to more

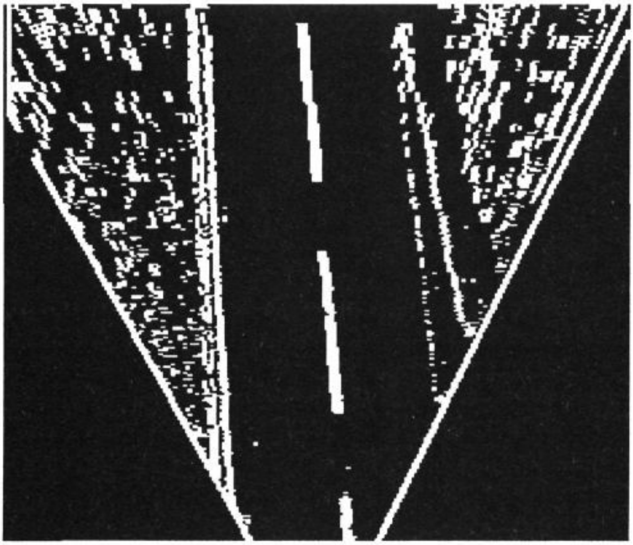

Figure 5: Plan-View Transformed Road Scene after Pre-Processing

computationally expensive methods such as damping [7] or filtering using a spectral analysis of $\underline{\mathbf{A}}$.

5 Results

The results of the above techniques are described here.

Figure 6 shows an initial guess made for the road scene and Figure 7 shows the final answer after non-linear least squares fitting. The result obtained is very accurate despite the poor initial guess.

Figure 8 shows the convergence of the model for this image. The shape of this convergence curve and the large number of iterations required is typical for an image with such a poor initial guess. However, it is usual to have a better guess than this, resulting in more rapid convergence, often in ~ 15 iterations. The fact that convergence occurs despite a poor initial guess is due to the tail on the sech() function in equation 8. The value and standard deviation (equation (6)) for each of the six parameters at convergence is shown in Figure 9. The low value obtained for each of the standard deviations shows that the parameters are all well determined, hence justifying the model used.

The algorithm has been run successfully on many different sequences, each of which is hundreds of images long. The only cause of mis-tracking has been found to be if a marking becomes occluded by another vehicle. This situation is readily identifiable from the standard deviations of the parameters, which become very large in relation to the parameters themselves.

One unexpected problem was that any region away from the lane markings that is completely devoid of edges (i.e. black in Figure 5) satisfies the model

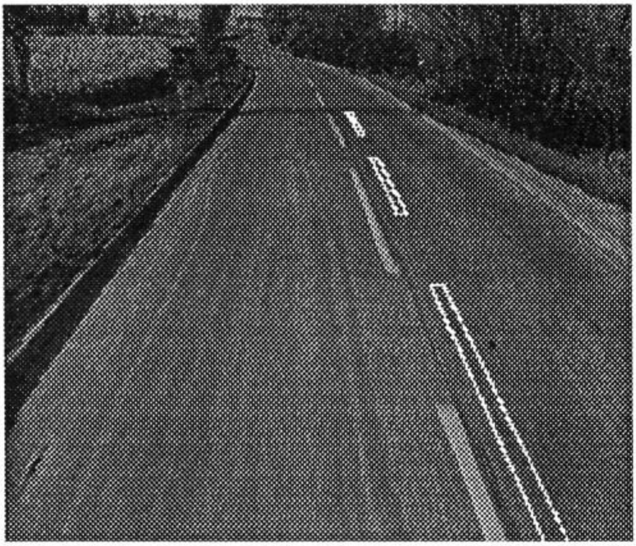

Figure 6: Initial Guess

$S_{(x,z)}(\underline{\mathbf{a}})$ exactly if $d = 0$. This meant that during convergence after a poor initial guess, $d \to 0$ and the position of the marking was not found correctly. This was solved by making the substitution:

$$d \equiv K_d \sin(d') + K_b + \tfrac{1}{2} \tag{13}$$

in equation (9). For our purposes $K_d = 0.45$ and $K_b = 0.05$ so that d was constrained to lie within the interval [0.1, 1.0].

6 Conclusions

An algorithm has been described which is suitable for use by an autonomous road vehicle to find lane markings in a scene reliably and with high accuracy. It uses a six parameter model to represent the portion of the image around the markings and a non-linear least squares technique is applied to refine these parameters to sub-pixel accuracy.

The algorithm has been tested on many sequences and has coped well with changes in luminance, sharp bends and changes in type of lane marking. It differs from other approaches which are used to guide autonomous vehicles (e.g. [8]) in that the completeness of the model ensures a high degree of confidence in the tracking result. If mis-tracking does occur then it is readily identified using the standard deviations of the parameters.

Further work currently under investigation includes improving the initial guess $\underline{\mathbf{a}}^k$ for the current frame by applying a predictive filter to previous values. The method of finding a minimum in a six dimensional search space using the Gauss-Newton method described here is robust but relatively computationally

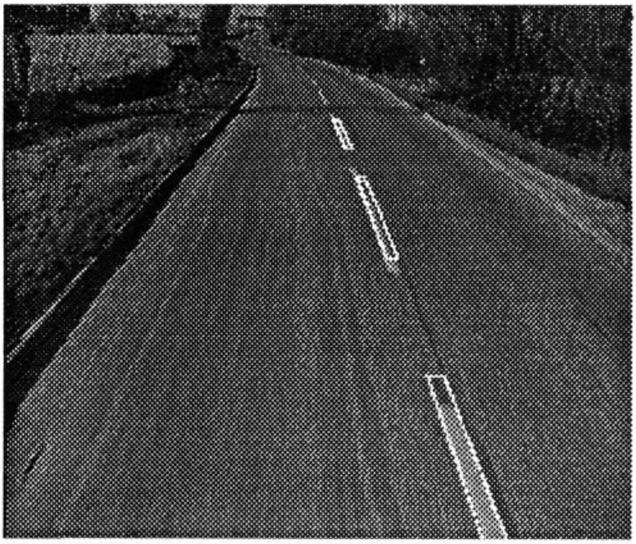

Figure 7: Final Result

expensive taking around 10 sec. per image on a single transputer. Other, faster techniques to achieve the same result must be investigated if this algorithm is to run in real-time on a small number of processors.

Thanks

We gratefully acknowledge the contribution made by Dr.B.R. Stonebridge to this work.

References

[1] A.D. Morgan, E.L. Dagless, D.J. Milford, and B.T. Thomas. Road edge tracking for robot road following: a real-time implementation. In *Image and Vision Computing*, pages 233–240, 1990.

[2] A. Grunes and J.F. Sherlock. Texture segmentation for defining driveable regions. In *Proceedings of British Machine Vision Conference*, pages 235–239, 1990.

[3] L. Schaaser and B.T. Thomas. Finding road lane boundaries for vision-guided vehicle navigation. In Ichiro Masaki, editor, *Vision Based Vehicle Guidance*, pages 238–254, Springer-Verlag, 1992.

[4] Carl-Erik Fröberg. *Numerical Mathematics : Theory and Computer Applications*, chapter 11. Benjamin/Cummings, 1985.

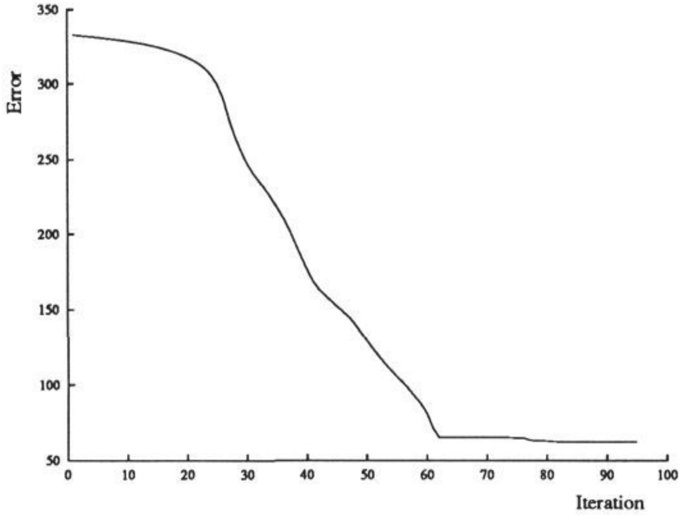

Figure 8: Convergence for One Image

Parameter	Value	$\pm\sigma$
a_0	160.110	± 0.1125
a_1	0.0129	± 0.0018
a_2	-0.000222	± 1e-6
a_3 (i.e. θ)	3.01966	± 0.0304
a_4 (i.e. d')	0.139263	± 0.0144
a_5 (i.e. $\mathcal{P}$)	90.701	± 0.2462

Figure 9: Average Value and Standard Deviation of Parameters

[5] A.D. Morgan. *Application of computer vision techniques to the visual navigation of a road following vehicle.* PhD thesis, University of Bristol, 1991.

[6] R.A. Lotufo, B.T. Thomas, and E.L. Dagless. Road following algorithm using a panned plan-view transformation. In *Proceedings 4th Alvey Vision Conference*, University of Manchester, 1988.

[7] B.R. Stonebridge and R.L. Soulsby. *Damped Nonlinear Least-Squares Computation of a Model for Scouring of the Seabed around a Vertical Cylinder.* Technical Report TR-91-34, University of Bristol, Department of Computer Science, December 1991.

[8] E. D. Dickmanns and B. D. Mysliwetz. Recursive 3-d road and relative ego-state recognition. In *Transactions on Pattern Analysis and Machine Intelligence*, pages 199–213, 1992.