A Method for Following Unmarked Roads

Ola Ramström and Henrik Christensen

***Abstract*— Road-following for ground vehicles has been widely studied and the solutions normally tracks line markings or a salient road edge. Most of the presented approaches are thus based on geometric or spectral models and does a fit of image data to the model to ensure robustness. For operation on unmarked roads under highly varying weather and light conditions a strategy is presented which is based on a probabilistic generative model. The method is evaluated on a number of test scenes and compared to well known methods from the literature.**

I. Introduction

Since the early eighties several road-following applications have been developed by research groups, such as [1]. Long-term evaluation of the techniques has also been performed and reported, e.g., across USA [2], through Germany [3], and around northern Italy [4].

The bulk of the road-following research is focused on following marked roads at high speed. Universität der Bundeswehr München (UBM) has developed an advanced system based on the 4D approach [5] [6], which extracts line markings or road-edge information from attentional windows in multi-focal cameras using predictive feed-forward to estimate the layout of the road based on reference models of the scene [7] [8]. In most of the reported approaches it is common to exploit the longitudinal patterns of roads: line markings, wheel tracks, and the road edge, e.g. [9], [10], [11], and [12].

For operation on unmarked roads the availability of prior models is limited. In such environments the road might not follow any standard model and in addition the contrast between road and non-road might vary considerably. In such a setting there is a need to adaptively acquire a model of the road to enable tracking. Such work has for example been reported by [13], [14], and [15]. In [13] and [14] an accurate probabilistic model is used for the image variation but the associated road model is very simple. At the other end of the spectrum a detailed road model is used by [15] for the mapping and estimation, but the segmentation of the road is based on an ad-hoc approach. Consequently it is natural to try to combine the two strategies. Through adaptive modeling of the road-/non-road regions and fitting these to a detailed (adaptive) road model it can be expected that a higher degree of robustness can be achieved. The present paper consequently presents a solution inspired by [14] and [15]. In addition the developed method is benchmarked against [14]

II. Overview

For the research a skid-steering all-terrain vehicle is used. The vehicle is equipped with a standard color camera and can operate at speeds up to 1.5 m/s. A standard visual front end model is used. For each image a number of feature maps is computed (see Section III). For the segmentation into road/non-road two Gaussian Mixture Mode (GMM) are used for each feature map. Based on the feature maps a generative model of the road layout is maintained.

The two GMM are computed using the road shape model and the road shape model is computed using the two GMM. To resolve this chicken-egg problem an initialization process computes an initial value (see section V). After initialization the road shape model and the two GMM are maintained by performing the following processes iteratively for each feature map:

1) Acquire a new observation from the camera.
2) Update feature map (see section III).
3) Update Gaussian models according to road shape (see section III).
4) Update road shape according to Gaussian models (see section IV).
5) Compute fitness of the road model (see section IV).
6) Select the road shape model with highest fitness across all features.
7) Steer toward the road center further up the road.

If the probability of having a correct road model is lower than 0.8 two iterations in row, the robot slows down and the model is reinitialized.

Note that the separate feature map-processes only communicate the parametric road shape model in step 5 above. Thus, a model that is well suited for distribution is available so as to ensure real-time operation.

III. Image features

The robot camera provides color images which are decomposed into a set of feature maps. For each separate feature map two GMM are computed using the road shape.

Images from the camera are resampled to be of size 61×50, as this is adequate for the steering of the vehicle. For each image three feature maps are computed:

1) f_{UV}, the UV channels from a YUV color coding reduces the effect of illumination variations by a linear transformation.
2) f_{rg}, red and green channels from normalized RGB (chromaticity) reduces the effect of illumination variations; $r = R/(R+G+B)$ and $g = G/(R+G+B)$
3) f_{int}, the intensity is useful when the color contrast is low. Intensity is defined as $(R+G+B)/3$.

The feature maps are correlated and one might expect that the overlap in distribution would pose a problem but each of the maps are invariant to different perturbations which increases robustness.

Two GMMs, one of the road and one of the background, are computed from each feature map. The GMM from feature map f_q, $q \in \{uv, rg, int\}$, of the road is denoted $\{G^i_{rd,q}\}^2_{i=0}$ and of the background $\{G^i_{bg,q}\}^2_{i=0}$. Hence, we use a mixture of three Gaussian for each segment. The size of the mixture was determined experimentally. The GMM of the road is computed within the road area, according to the current road shape model, and the GMM for the background at the surrounding area. A safety region around the road edge is omitted in the two computations due to the uncertainty of the exact road shape. The two GMM are computed using the EM-algorithm, see [16] for details.

IV. Road shape

The road shape is used to maintain the GMM and to steer the vehicle. The shape of a normal road is simple and can be expressed by a parametric model, such as a clothoid or a sequence of constant curvature segments. In the present work we have adopted a second order image based model, as it captures both road models adequately and the relation between the image and the surface is assumed to be a homography.

The road shape is describe by 5 parameters: $Rd = \{rw, hn, k_0, k_1, k_2\}$; where rw is the road width at the bottom of the image, hn is the horizon, k_0 lateral offset of the road center, k_1 yaw at bottom of the image, and k_2 curvature of the road. Hence the centerline of the road is described by the equation:

$$x = k_0 + k_1 y + k_2 y^2 \tag{1}$$

as depicted in figure 1 and motivated in [7] and [15].

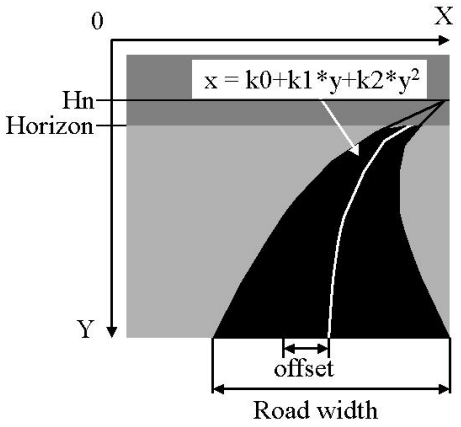

Fig. 1. Road shape

The road model can handle positive or negative curvature but is unable to handle both at the same time; e.g., only a right or left curve can be modeled, not an S-shaped road. The camera is in the present system mounted at a height of 1 m above the ground so there is a significant perspective effect, which implies that the limitation has a very limited effect, compared to the robustness achieved.

For the segmentation/adaptation knowledge of the horizon is embedded in the computations. The GMM of the road is computed below this horizon. However, the GMM of the background is not limited by the horizon, since the road often occupies the larger part of the area below the horizon. It has been empirically found that the background model benefits from this.

A. Road probability map

The road shape parameters are updated from the classification result using the GMM of the road and the background at each feature map. The classification result is temporarily represented by a road probability map, where each pixel reflects its probability of being a road pixel. The process is similar to [14], therefore only a short summary will be given here.

Using feature map f_q, $q \in \{UV, rg, int\}$, the probability that a feature vector $v_q = f_q(x, y)$ belong to the ith Gaussian model of $x \in \{rd, bg\}$ is denoted $G^i_{x,q}(v_q)$. Hence the probability that v_q belong to the class x is:

$$p(v_q|rd, q) = max_i(\{G^i_{rd,q}(v_q)\}^2_{i=0}) \tag{2}$$

$$p(v_q|bg, q) = max_i(\{G^i_{bg,q}(v_q)\}^2_{i=0}) \tag{3}$$

Using Bayesian statistics, the probability for v_q to belong to the road statistics is therefore:

$$p_{rd} = \frac{p(v_q|rd)P(rd)}{P(rd)p(v_q|rd) + P(bg)p(v_q|bg)} \tag{4}$$

The priors $P(rd)$ and $P(bg)$ is proportional to the road size in [14], one could improve this by using a probability map computed from the present road model. Such a prior could also involve prediction based on the control of the vehicle. There is much work done on prediction but this is outside the scope of this study and the $P(rd)$ is therefore set to 0.5 below the static horizon and 0 above.

Figure 2, middle image, illustrates a typical road probability map. We observe that there are two uncertain regions on the road, corresponding to specularities from the sun. In the next section we will discuss a method for compensation of this effect.

B. Illumination compensation

One common problem in road-following is specularities and shadows. In this study we are inspired by the work of [15].

A Gaussian model of the current road layout is computed from the intensity feature map and pixels deviating by more than twice its variance are considered as specularities (higher intensity) or shadows (lower intensities). The probability maps for UV (f_{UV}) and chromaticity (f_{rg}) are subsequently modified to include the outlier regions by setting all such pixels as road. Note that this is not done for RGB and intensity images, where illumination variation is an intrinsic property of the feature space. Figure 2 illustrates an observed image, the road probability map, and probability map after correction for specularities and shadows

However, it is not straight forward to extract the road shape from the road probability map. Several methods are applicable, here we have chosen to evaluate two: Voting [14] and a new

Fig. 2. Upper left to lower right: an observed image, the road probability map, and probability map after compensation for specularities.

method denoted max-edge, which is inspired by [15]. The two methods are presented in the next two sections.

C. Voting

[14] suggests voting to extract the road shape. A similar road probability map is used to vote for the most likely road shape; each pixel votes for shapes which are consistent with its value. The votes are weighted by their strength, i.e. their probability. Thus, a pixel (x, y) votes for each model according to its strength $p_{rd}(x, y)$ and each model excludes it with strength $1-p_{rd}(x, y)$. In this implementation the road probability map is down sampled to 30x25 pixels before voting and only pixels under the static horizon are used. The larger size of the road probability map is still used for extraction of GMM.

This method has a high stability due to the integration of large regions, but this is naturally at the cost of computational complexity. The complexity of updating the road shape is proportional to the image area. In order to decrease the complexity only a four parameter model is used for the road shape, $\{rw, hn, k_0, k_1\}$ and $k_2 \equiv 0$ in this implementation. Note also that [14] proposes a straight road model where only hn and k_1 are maintained.

D. Max-edge

[15] suggests a method for extracting the road edge using the road feature statistics. The solution is rather ad-hoc and therefore hard to qualitatively evaluate in comparison to others. However, it is clear that the approach to update the road shape model using edge estimates greatly reduces the complexity compared to the method presented in the previous section.

We propose a revised method for finding road edges, denoted max-edge. Max-edge extracts maximal edge responses in the road probability map. A derivative of Gaussian in x- and y-direction is applied to the road probability map, resulting in a D_x and a D_y image. Edges on the left side of the road are expected to have a positive Dx-value and edges on the right side a negative. Furthermore, the actual road edges are expected to have the largest gradient-magnitude along each horizontal line. This leads to the following algorithm:

Compute the horizontal gradient D_x, vertical gradient D_y, and the gradient-magnitude image G; where

$$G(x, y) = D_x(x, y)^2 + D_y(x, y)^2.$$

For each horizontal line y find the strongest left-oriented gradient-magnitude value

$$x_{l-max}(y) = \max_x \{G(y, x) | D_x(x, y) > 0\}$$

and select the set

$$x_l(y) \in \{x | G(x, y) > 0.99 * x_{l-max}(y) \text{ and } D_x(x, y) > 0\}$$

as left edge candidates. Similarly, for each horizontal line y find max right-oriented gradient-magnitude value

$$x_{r-max}(y) = \max_x \{G(y, x) | D_x(x, y) < 0\}$$

and select the set

$$x_r(y) \in \{x | G(x, y) > 0.99 * x_{r-max}(y) \text{ and } D_x(x, y) < 0\}$$

as right edge candidates.

Finally, the road edge pixels votes for road shapes which are consistent with their positions. The similarity between the edge pixels and the road shape is measured as:

$$S = \sum_{y=0}^{hn} \left(\sum_{x \in x_l(y)} \frac{1}{1 + 0.1 d_l(x, y)^2} + \sum_{x \in x_r(y)} \frac{1}{1 + 0.1 d_r(x, y)^2} \right)$$

where $d_l(x, y)$ is the distance between a left road edge at (x, y), $x \in x_l(y)$, and the corresponding edge according to the road shape model. $d_r(x, y)$ is the corresponding measure for the right edge. Note that $0 \leq S \leq 1$ and $S = 1$ for a perfectly extracted road shape.

The benefits of the max-edge method is that the complexity is proportional to the road height. Therefore a five parameter model, including curvature, is allowed in the current implementation.

E. Fitness of model

Since the vehicle is designed to move autonomously it is of interest that it stops and reinitializes if it is uncertain of its current road model. We will in this section describe a measure of the fitness of a road model.

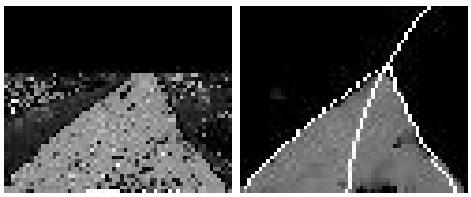

Fig. 3. Road probability map before and after thresholding

In figure 3, left most image, an example of a road probability map is shown. The road probabilities corresponding to the true road is on average 0.6. However, there is a clear road shape present. The fitness value in this implementation is based on the shape of the classification rather than the amplitudes of probabilities. The road probabilities $p_{rd}(x, y)$ are remapped according to the following mapping function, a soft decision function:

$$c(x, y) = \begin{cases} 0 & \text{if } p_{rd}(x, y) < 0.4 \\ 5 * (p_{rd}(x, y) - 0.4) & \text{if } 0.4 \leq p_{rd}(x, y) \leq 0.6 \\ 1 & \text{if } p_{rd}(x, y) > 0.6 \end{cases} \quad (5)$$

The remapped probability image is subsequently post processed with a median filter to reduce noise. The final classified image is matched against the extracted road shape model, illustrated in figure 3, right most image. The fitness value is defined as the mean square difference between the road shape model and the classified image, below the static horizon.

F. Voting versus max-edge complexity

In figure 4 the resulting road shape models using voting and max-edge are shown. The red-lines illustrates the road shape model up to the horizon (horizontal red line). The white-line illustrates the suggested steering direction; from the current position to the middle of the road half way up to the horizon or at the tangent point, from current position to inner curve-edge, in sharp curves, as presented by [17].

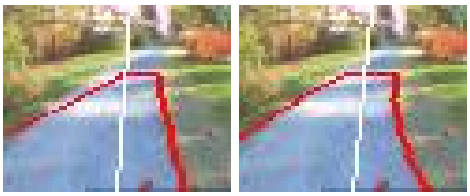

Fig. 4. Road shape extracted by voting (left) and max-edge (right)

The complexity of extracting a road shape using max-edge is proportional to the road height and proportional to the image size using voting. This is the reason why the voting only extracts straight roads using a down-sampled road fitness map. However, the max-edge method requires processing of derivatives before the extraction of shape parameters can start. It has been empirically found that the max-edge and voting method has the same execution time using the given implementations and image sizes. It should, however, be noted that a larger image size would make the voting algorithm slower compared to max-edge.

V. Initialization

The GMM of the road and the background are computed using the road shape and the road shape is computed using the GMM. We will in this section discuss how this chicken-egg problem is resolved by an initialization process that utilizes random search.

During the initialization the road width and the horizon are set to a predefined value and k_0 to k_2 are varied. The road shape is initialized to a "common" state, a straight road in front of the vehicle ($k_0 = k_1 = k_2 = 0$), and the GMM are computed according to the this road shape. If the fitness of the model is lower than 0.8 the process continues with a different road shape, where k_0–k_2 are changed using a Gaussian random variable:

$$k_n = k_n + N(0, v); \tag{6}$$

where $n \in \{0, 1, 2\}$ and $N(0, v)$ is a normal distribution with zero mean and v variance.

This process will explore variations similar to the "common" road shape and fewer "unusual" shapes. However, in theory all shapes will eventually be explored and in practice the process normally initializes within a few iterations.

VI. Reference model

In order to evaluate the proposed system, we have implemented SCARF [14] as a reference model.

The proposed road-following system, using the voting method, is similar to SCARF. The major differences are that SCARF uses the RGB feature space and does not compensate for shadows and specularities. However, the road probability map and voting mechanism, see section IV), are the same. Furthermore, SCARF does not compute any fitness value. The proposed model, without illumination-compensation and with only the RGB color space, is similar to SCARF. The introduction of the fitness value has only the role of verifying if the road model is accurate.

VII. Evaluation

The proposed road-following system is evaluated on a set of recorded image sequences. The sequences has been recorded by the same vehicle under different seasonal and weather conditions, posing a number of different challenges. Samples from the sequences are shown in figure 5.

In total six sequences are used in the evaluation, the sequences were selected based on their different properties: The wet1 and wet2 sequences were recorded after rainfall and the wet asphalt creates challenging patterns. The snow sequence was recorded during the winter, the snow can be challenging to due to the illumination compensation which might not be able to differentiate snow and sun specularities. The sun1 and sun2 sequences were recorded during bright sun light causing challenging shadow effects. The dawn sequence was recorded when the sun was low on the horizon, behind the trees.

Five of the sequences are between 60 and 90 seconds long, the winter scenario is only 10 seconds long. The robot was moving at 1 m/s and was sampling 2 images per second on average.

VIII. Comparing voting, max-edge, and SCARF

The voting, max-edge, and SCARF method was applied to the six sequences. If the fitness was below 0.8 two iterations in row the system reinitialized. In table 6 and 7 the number of initializations and the mean fitness value across the six sequences are reported.

We observe that the SCARF method clearly has problems with the sequences. In figure 8 typical behaviors of the three models are shown. SCARF uses the RGB color space and is therefore sensitive to illumination variations and false volatile patterns in the road are easily confused for the road. The voting and max-edge methods utilize a number of competing feature maps and it is therefore more robust. The voting method has its limitation in curves and when the road edge is fuzzy it can be confused. However, these errors are normally small. In figure 9 a scene is shown where the max-edge fails and voting succeeds. Strong edge responses are found distant from the

Fig. 5. Samples from the six evaluation sequences. Upper left to lower right: wet1, wet2, snow, sun1, sun2, dawn

init	wet1	wet2	snow	sun1	sun2	dawn
voting	0	0	0	0	0	0
max-edge	0	1	1	0	0	0
SCARF	4	15	2	10	6	4

Fig. 6. Number of initializations

fit	wet1	wet2	snow	sun1	sun2	dawn
voting	0.92	0.92	0.87	0.94	0.92	0.95
max-edge	0.93	0.93	0.86	0.94	0.93	0.95
SCARF	0.86	0.84	0.83	0.85	0.89	0.88

Fig. 7. Mean fitness

road edge, if the true edge is not clearly visible such noise can corrupt the model. The voting method succeeds since it fuses more data, the noise is normally much smaller than the amount of true data. In figure 10 a scene is show where both the max-edge and voting method fails. In the chromaticity feature space the illumination compensation occasionally adds some of the snowy surrounding to the road model, it is likely that the non-snow patches in the surrounding confuses the two methods. In this example the max-edge detects the fault and stops to reinitialize, but the voting does not. However, both recovers from the error.

Fig. 8. Typical behavior of the three methods: Voting, max-edge, and SCARF

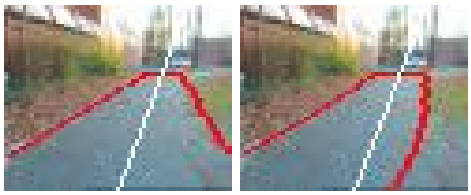

Fig. 9. A scene where the max-edge fails and voting succeeds

Fig. 10. A scene where both the max-edge and voting method fails

IX. Comparing feature spaces

In this section we evaluate the voting method with only one of its three feature maps at a time. That is, the redundant distributed processing of visual features is omitted. Note that this is equal to SCARF apart from the feature space and that chromaticity and UV will compensate for illumination effects.

In table 11 and 12 the number of initializations and the mean fitness value across the six sequences are reported.

init	wet1	wet2	snow	sun1	sun2	dawn
rg	0	1	2	3	3	0
UV	0	0	0	2	32	0
int	7	15	26	8	2	19

Fig. 11. Number of initializations for the set of different visual features

fit	wet1	wet2	snow	sun1	sun2	dawn
rg	0.93	0.89	0.90	0.82	0.91	0.94
UV	0.93	0.92	0.92	0.83	0.77	0.94
int	0.86	0.86	0.82	0.70	0.91	0.85

Fig. 12. Mean fitness for the set of different visual features

We observe than no single feature map can cope with the full set of sequences, although some can handle specific challenges well. For example UV can handle the wet asphalt and snow well but has large problems with “sun1”, which has a strong green bias. Context recognition for adaptive feature fusion might be a strategy to achieve, but that is clearly beyond the present work.

X. Other road segmentation examples

On the DARPA Grand Challenge web site there is an image gallery [18]. We evaluated the presented methods on six selected images of challenging unmarked roads, see figure 13 and 14.

There is no ground-truth for these images. We can observe that the two methods chose different, but equally likely, models of the third image (top-right). In the fourth image (bottom-left) both methods chose the stripe between wheel-patterns; note that the white line, the direction indicator, points to the

Fig. 13. Snap-shots from DARPA Grand Challenge using voting

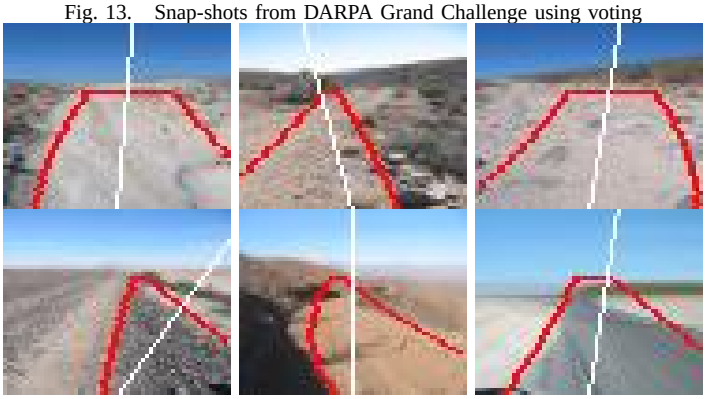

Fig. 14. Snap-shots from DARPA Grand Challenge using max-edge

right since the vehicle is near the left edge according to the road model. However, it can be concluded that both methods generate likely road shape models from this set of challenging unmarked roads.

XI. Summary

A road-following system for unmarked roads has been presented. The system is similar to [14] and [15], which represent two different methods for road shape extraction, voting and max-edge. The system is focused on generating an accurate model of the road using redundancy of image features. The two methods for extracting road shape have different properties: voting fuses much data using a simple road shape model and max-edge fuses less data using a richer road shape model.

An evaluation on a set of image sequences demonstrated the benefits of redundancy of image features. From the result it is clear that no single visual feature is applicable for the variations of visual environment. The properties of the two road shape methods was also demonstrated. The voting was more robust than max-edge, indicating that the amount of data fused is more important than road shape accuracy. However, a richer road shape model have other benefits such as better prediction abilities and steering guidance. The evaluation did not include such properties and the difference between the two methods was small. It is therefore left for further investigation what qualitative effects the enhanced prediction and steering information has to the system.

Acknowledgment

This research has been sponsored by the Swedish materials defense administration (FMV) under the “UGV demonstrator” contract. The support is gratefully acknowledged.

References

[1] E. D. Dickmanns, “The development of machine vision for road vehicles in the last decade,” *Paper presented at the IEEE Intelligent Vehicle Symposium, 18-20 June, Versailles, France*, 2002.
[2] Http://www-2.cs.cmu.edu/afs/cs/user/tjochem/www/nhaa/nhaa_home_page.html.
[3] M. Maurer, R. Behringer, S. Furst, F. Thomanek, , and E. D. Dickmanns, “A compact vision system for road vehicle guidance,” in *Proceedings of the International Conference on Pattern Recognition (ICPR ’96) Volume III-Volume 7276*. IEEE Computer Society, 1996, pp. 313–317.
[4] M. Bertozzi, A. Broggi, and A. Fascioli, “Vision-based intelligent vehicles: State of the art and perspectives,” *Robotics and Autonomous Systems*, no. 32, pp. 1–16, 2000.
[5] E. D. Dickmanns and T. Christians, “Relative 3d-state estimation for autonomous visual guidance of road vehicles,” *Intelligent Autonomous Systems 2, An International Conference*, pp. 11–14, 1989.
[6] E. D. Dickmanns, “Dynamic vision for intelligent motion control,” *Proc. IEEE-Workshop Intell. Motion Contr*, 1990.
[7] E. D. Dickmanns and B. D. Mysliwetz, “Recursive 3-d road and relative ego-state recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 199–213, 1992.
[8] R. Gregor, M. Lützeler, M. Pellkofer, K.-H. Siedersberger, and E. Dickmanns, “Ems-vision: A perceptual system for autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 48–59, 2002.
[9] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, “Toward autonomous driving: The cmu navlab. part i: Perception,” *IEEE Expert*, vol. 6, no. 4, pp. 31 – 42, August 1991.
[10] D. Pomerleau and T. Jochem, “Rapidly adapting machine vision for automated vehicle steering,” *IEEE Expert: Special Issue on Intelligent System and their Applications*, vol. 11, no. 2, pp. 19–27, April 1996.
[11] A. Broggi, M. Bertozzi, and A. Fascioli, “Architectural issues on vision-based automatic vehicle guidance: the experience of the argo project,” *Real-Time Imaging Journal*, no. 6, pp. 313–324, 2000.
[12] A. Apostoloff and A. Zelinsky, “Vision in and out of vehicles: Integrated driver and road scene monitoring,” *The International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 513–538, 2004.
[13] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, “Vision and navigation for the carnegie-mellon navlab,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 362 – 373, May 1988.
[14] J. Crisman and C. Thorpe, “Scarf: A color vision system that tracks roads and intersections,” *IEEE Trans. on Robotics and Automation*, vol. 9, no. 1, pp. 49 – 58, February 1993.
[15] M. Sotelo, F. Rodriguez, L. Magdalena, L. Bergasa, and L. Boquete, “A color vision-based lane tracking system for autonomous driving on unmarked roads,” *Autonomous Robots*, vol. 16, no. 1, pp. 95–116, January 2004.
[16] D. Forsyth and P. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.
[17] M. Land and D. Lee, “Where we look when we steer,” *Nature*, no. 369, pp. 742–744, 1994.
[18] Http://www.darpa.mil/grandchallenge04/media_images.htm.