Model-Based Car Tracking Integrated with a Road-Follower

Frank Dellaert Dean Pomerleau Chuck Thorpe
Computer Science Department and The Robotics Institute
Carnegie Mellon University, Pittsburgh PA 15213

Abstract

This paper discusses how we integrated our 3D car tracking approach with the lane following module RALPH on the Navlab autonomous vehicles, obtaining a hybrid vision system that tracks both the road and cars better than those two systems in isolation. The tracking system brings precise and crisp measurements of the car in the image, and performs image stabilization. However, because it does not know about the yaw or lateral offset of the ego-vehicle, its curvature estimate can be misguided. RALPH takes a more global image processing approach and can provide this missing information, as well as a good estimate of curvature, so that the combined curvature estimate is superior to both taken in isolation. The additional information provided by RALPH also improves tracking performance, and allows us to estimate properties of the tracked car that were previously unobservable, in particular its in-lane displacement. Better car tracking, and a better idea of where the road is, gives us a substantial foundation on which to base other capabilities needed to realize fully autonomous vehicles.

1 Introduction

In this paper we show how we integrated our 3D car tracking approach with the lane following module on the Navlab autonomous vehicles, obtaining a hybrid vision system that tracks both the road and cars better than those two systems in isolation. The Navlabs are the experimental platforms for the Automated Highway System (AHS) research being conducted at CMU[1]. Several other groups have similar research programs, in particular Dickmanns' group at the Bundeswehr University in Munich [2]. Much of the research done on the Navlabs has been concentrated on road-following. The first convincingly successful system, ALVINN, consisted of a neural network that learned to predict steering direction from subsampled video images [3]. Since then, ALVINN has been superseded by RALPH, which uses a more domain specific method to estimate the curvature of the road and the lateral offset of the vehicle, which can then be used to calculate a steering command [4]. Although RALPH has been quite successful for road-following, it does not provide any information about the position or behavior of other cars on the road. This capability, *situational awareness*, enables autonomous vehicles to plan a course of action, ranging from matching the speed of a car ahead, to planning more complicated behaviors involving lane changes and overtaking other cars. To provide situational awareness, we recently proposed a model-based vision approach to car tracking, in which we estimate the 3D position and motion of a car by tracking a 2D bounding box in the video stream [5]. Since only line segments are tracked, the image processing involved is relatively simple, and the system can run at frame rate.

By combining the strengths of the tracking algorithm with the strengths of the RALPH road following module we can obtain an overall system which is superior to both approaches taken individually. The tracking algorithm provides very crisp measurements of the relative position and speed of the car that is being tracked, but could do even better if given an idea of exactly where the road is, as then it can form better expectations of how a tracked car will behave. Likewise, the road-follower could benefit from information provided by the tracker, since the position of the tracked vehicle provides an important clue as to where the road is. Having this additional estimate of curvature will be especially helpful in situations where RALPH traditionally has problems, e.g., when the flat earth assumption is violated. A principled way of combining the information provided by both techniques separately is by using an extended Kalman filter [6], and that is the way we will approach it here. Since the tracking algorithm relies heavily on a Kalman filter already, integrating the RALPH measurements in this framework comes rather naturally.

In the remainder of this paper we will give a brief overview of RALPH (Section 2) and the image processing used for the tracking (Section 3). Then, in Section 4, we describe the Kalman filter used for inte-

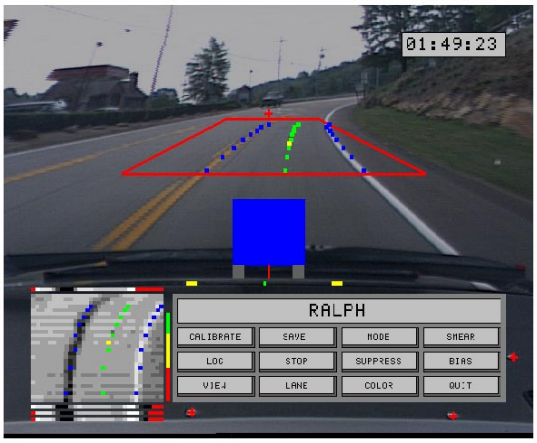

Figure 1: Screen shot of RALPH.

grating the two modules, i.e., we describe the model of how both cars relate to each other and the road, formulate the dynamics of this model and discuss the measurement equations that relate the two vision techniques to the model. Both qualitative and quantitative results are presented in Section 5 to establish the superiority of the resulting hybrid system to both modules in isolation. Finally, we conclude with a discussion and opportunities for future work in Section 6.

2 The RALPH Vision System

The RALPH vision system helps automobile drivers steer, by processing images of the road ahead to determine the road's curvature and the vehicle's position relative to the lane center. RALPH uses this information to either steer the vehicle autonomously, or warn the driver if he/she is steering inappropriately.

In order to locate the road ahead, RALPH first resamples a trapezoid shaped area in the video image to eliminate the effect of perspective (see Figure 1). RALPH then uses a template-based matching technique to find parallel image features in this perspective free image. These features can be as distinct as lane markings, or as subtle as the diffuse oil spots down the center of the lane left by previous vehicles. RALPH rapidly adapts to varying road appearance and changing environmental conditions by altering the features it utilizes to find the road. This rapid adaptation is accomplished in under one second.

3 Image Processing for Tracking

Although many features could be used to track a car in a sequence of video images, we track only the 2D bounding box around the car (Figure 2). When looking at the image of a car, you can see that in general it has strong edges on all sides where the image changes from the car to the background. This contour of strong edges is not always regularly shaped, but it can be approximated by a rectangle for a wide range of aspects. In the remainder of this section we will briefly discuss the image processing technique we use to track this 2D bounding box in a video stream. A more detailed treatment and a discussion of related work can be found in [5], where we also make the coupling with the Kalman filter more explicit. The overall approach is closest in spirit to the work of Schmid [7], although the underlying image processing is quite different.

We rely on an energy minimization technique to find the bounding box around the tracked car in each frame. An initial estimate is obtained by projecting the imaginary 3D bounding box around the car, as estimated by the Kalman filter (see below), into image space. As we are looking for a rectangular contour with strong edges on all sides, we maximize an objective function F that measures the strength of the edges around a particular contour. With (t, l) and (b, r) the top-left and bottom-right coordinates of the bounding box, respectively, and $I_v(I_u)$ the horizontal (vertical) gradient image, we define F as the contour integral of the average gradient perpendicular to the contour:

$$\begin{aligned} F &= \frac{1}{r-l}\int_l^r I_u(t,v)\,dv + \frac{1}{r-l}\int_l^r I_u(b,v)\,dv \\ &+ \frac{1}{b-t}\int_t^b I_v(u,l)\,du + \frac{1}{b-t}\int_t^b I_v(u,r)\,du \end{aligned}$$

We can relate this objective function to a Bayesian likelihood function by using the Gibbs/Boltzmann distribution [8, 9]. Thus, if we define $\alpha E_d = -F$ as the energy term we are trying to minimize, and $\mathbf{x} = [t\, b\, l\, r]^T$ as the 4-dimensional vector encoding the position of the bounding box, then the likelihood of bounding box $\mathbf{x}$ given the image $\mathbf{z}$ can be expressed as (where Z_d is a normalization factor):

$$P(\mathbf{z}|\mathbf{x}) = \frac{1}{Z_d}\exp[-\alpha E_d(\mathbf{x},\mathbf{z})] \qquad (1)$$

The advantage of working with this probabilistic interpretation is that we can conveniently introduce the notion of a Bayesian prior, and combine it by means of Bayes law with the likelihood term above. The result is a maximum a posteriori (MAP) estimate for the position of the bounding box. For example, when working in image space, the prior could be a Gaussian density centered around the previous position of the bounding box [5, 9].

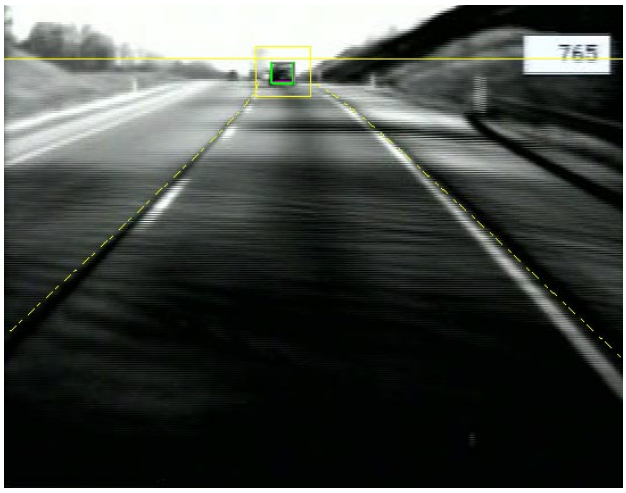

Figure 2: Scaling a hill at $t = 765$.

Initialization of tracking is done automatically using the Candidate Selection and Search (CANSS) algorithm, which we discuss in detail elsewhere [5, 10]. It uses a Hough Transform on the image gradient to advance candidate image rows and columns that might contain edges of a car, after which a combinatorial search takes place for the bounding box most probably generated by a car. This search minimizes the same energy measure E_d that we use for tracking, but combines it with a prior probability distribution over likely bounding boxes.

4 Integration: the Kalman Filter

We use a Kalman-Bucy filter to accomplish three simultaneous goals: (a) extract useful information from the tracking process, (b) improve tracking performance by having better expectations of how the tracked car will behave, and (c) conveniently integrate the additional measurements provided by RALPH. A Kalman filter implements the iterative application of Bayes law under Gaussian white noise assumptions for both dynamic and measurement noise, and the use of linear dynamics and measurements [6]. It also propagates the conditional densities involved forward in time between measurements. We use an extended (continuous) Kalman-Bucy filter as both our system dynamics and measurements are non-linear.

4.1 System Dynamics

The filter has 14 state variables, chosen because they represent a quantity of interest, are needed to model the dynamics, or both. They can be partitioned in a natural way, which is how we will discuss them.

For the ego-vehicle B we model the forward velocity V_b, and furthermore the yaw ψ_b and lateral offset d_b, both with respect to the lane center. Thus, $\mathbf{x_b} = [V_b\, d_b\, \psi_b]^T$, where the state variables are related via the following vector differential equation:

$$\dot{\mathbf{x_b}} = \begin{bmatrix} \dot{V_b} \\ \dot{d_b} \\ \dot{\psi_b} \end{bmatrix} = \begin{bmatrix} 0 \\ -V_b sin(\psi_b) \\ -\psi_b / T_{\psi_b} \end{bmatrix} + \begin{bmatrix} w_{V_b} \\ 0 \\ w_{\psi_b} \end{bmatrix}$$

Here w_{V_b} and w_{ψ_b} are Gaussian white noise terms that constrain how much velocity and yaw can change over time. V_b is modeled as a random walk, and we use the dynamic noise term to constrain the acceleration within reasonable bounds. The yaw ψ_b is modeled as a zero-mean time-correlated noise, as we know that the mean yaw with respect to the road must be zero.

The lane is modeled as having constant curvature κ and width W_r. The time derivative of the curvature is modeled as a random walk, i.e., $\mathbf{x_r} = [\kappa\, \dot{\kappa}\, W_r]^T$ and

$$\dot{\mathbf{x_r}} = \begin{bmatrix} \dot{\kappa} \\ \ddot{\kappa} \\ \dot{W_r} \end{bmatrix} = \begin{bmatrix} \dot{\kappa} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ w_{\ddot{\kappa}} \\ w_{W_r} \end{bmatrix}$$

The position of the tracked vehicle C is given by its distance Y_c *along the road*, with lateral offset from the lane center d_c. In contrast to most work in car tracking, *we do not make a flat earth assumption*, but estimate the vertical coordinate Z_c of the tracked vehicle as well. We do require that the vehicle is in the same lane as the tracker, which we argue in detail elsewhere [5]. Lastly, we also model the width, length, and height of the vehicle as unknown constants, leading to the 7-variable state $\mathbf{x_c} = [Y_c\, V_c\, d_c\, Z_c\, W_c\, L_c\, H_c]^T$. V_c and Z_c are modeled as random walks, d_c as a time-correlated noise, and the distance Y_c changes in function of both V_b and V_c:

$$\dot{\mathbf{x_c}} = \begin{bmatrix} \dot{Y_c} \\ \dot{V_c} \\ \dot{d_c} \\ \dot{Z_c} \\ \dot{W_c} \\ \dot{L_c} \\ \dot{H_c} \end{bmatrix} = \begin{bmatrix} V_c - V_b \\ 0 \\ -d_c / T_{d_c} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ w_{V_c} \\ w_{d_c} \\ w_{Z_c} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

By including them as state variables, W_c, L_c, and H_c are iteratively estimated, based on the initial measurement output by the CANNS algorithm. The length L_c can only be observed when the tracked vehicle goes through a curve, but after it has been properly estimated it yields important information about the yaw of the tracked vehicle.

Finally, since the image measurement equation is very sensitive to changes in camera pitch, we estimate that on-line as well, in effect achieving image stabilization. We do that by means of the state variable α_n, a zero-mean time correlated noise which is added to the camera calibration baseline pitch. Thus:

$$\dot{\alpha_n} = -\alpha_n / T_{an} + w_{\alpha_n}$$

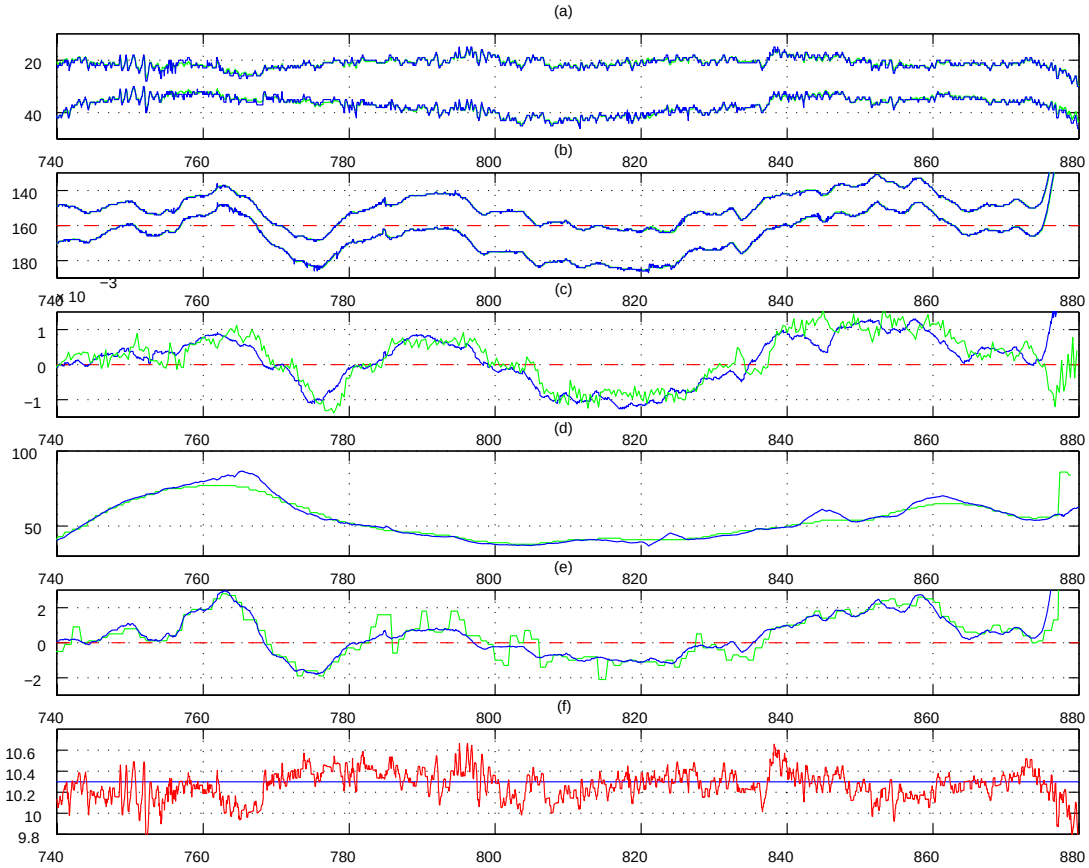

Figure 3: Combined system output. (a) pixel coordinates of top, bottom, and (b) left, right edges of tracked car; (c) curvature κ compared with the gyro (in gray); (d) distance Y_c to and (e) lateral position X_c (not d_c!) compared with radar (in gray); (f) pitch bounce estimate α_n

4.2 RALPH and Image Measurements

We integrate 3 quantities estimated by RALPH: road curvature κ, ego-vehicle lateral offset d_b, and lane width W_r. The measurement equation is trivial, as these are state variables of the filter. They are passed to the filter at a rate of 4 Hz, in addition to a measurement of the vehicle speed V_b obtained from GPS.

The image measurement equation simply consists of projecting the imaginary 3D bounding box around the tracked vehicle, as defined by the three state variables $[W_c\ L_c\ H_c]^T$, into the image. The 3D position and attitude of the car can be obtained directly from the filter, by combining the curvature estimate κ with the arclength Y_c, and the lateral offsets d_b and d_c. Taking into account the camera calibration and the current estimate of camera pitch, both the projection *and* its Jacobian are evaluated numerically using an OpenGL 3D graphics accelerator card. After the projection, it is a simple matter to determine the predicted 2D bounding box $[t\ d\ l\ r]^T$, which is used as an initial estimate for the image processing discussed above.

5 Results

Tracking results suggest excellent performance for the resulting integrated system, as will be illustrated here in both a qualitative and quantitative manner. In Figure 3 we present the output of the tracker on a long sequence of video recorded on the Navlab 8, an Oldsmobile Silhouette minivan, while it was being driven manually on I-79N near Pittsburgh. The images were taken early in the morning with the sun still low on the horizon, throwing long shadows of trees across the road. These shadows created strong distracting edges on the road surface. In addition, the terrain at that location was quite hilly, often violating the flat earth assumption for extended periods of time, in particular on hill crests.

The sequence shown in Figure 3 illustrates the strengths of our approach to car tracking, that have previously been discussed in [5]. Most notably, despite the challenging nature of the sequence, the system kept track at all times for the duration of the sequence. Since the processing was done at frame rate, and the sequence lasted 140 seconds, this represents

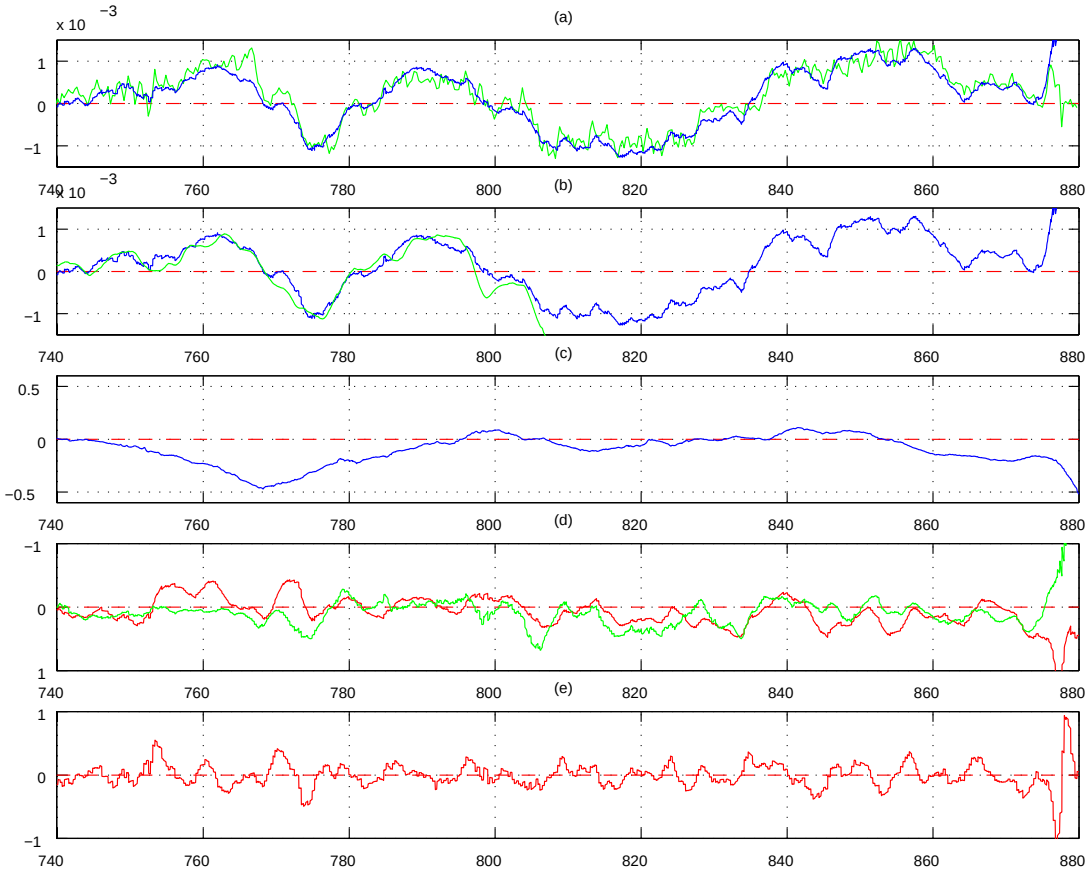

Figure 4: Benefits of integration. (a) curvature estimate κ compared with RALPH (in gray) (b) and with the tracker alone (in gray); (c) estimated target z coordinate Z_c; (d) lateral offset estimates d_b and d_c (in gray); (e) yaw estimate ψ_b

a continuous track over 4200 individual frames, while the distance covered was more than 4 km. In addition, the state estimates obtained by the tracker correspond closely to ground truth, as can be seen from Figure 3 (c) to (e), where the estimates of road curvature and relative target position are compared with recorded measurements from a yaw rate gyro and a Delco millimeter wave radar. The on-line pitch estimation, shown in panel (f), provides excellent image stabilization such that we do not lose track even when the Navlab drives over large bumps, e.g., the particularly challenging segment around $t = 750$. The extreme values at the end of the sequence occur because the Navlab exits the highway at that point.

The combination of RALPH and the tracker performs markedly better than RALPH in isolation, particularly where RALPH traditionally has problems, e.g., when the flat earth assumption is violated. This occurs for example at t=765, shown in Figure 2. Our tracking filter does not make a flat earth assumption: in Figure 4 you can see that the Z_c coordinate of the target is indeed negative at $t = 765$, while at the same time the grossly overestimated curvature output by RALPH is corrected by the tracker. Note that, as both the tracker and RALPH look ahead when estimating curvature, they will lead the gyro measurement when entering or leaving a curve. Finally, note that the filtered curvature estimate κ is a lot smoother than the raw RALPH measurement (panel a).

The integrated system also outperforms the tracker taken in isolation. To show this, we ran the tracker with the same settings but without the RALPH measurements, and compare its κ estimate with the integrated system in Figure 4 (b). Note that the comparison will not be not entirely fair, since in fact the tracker can be optimized to cope with the absence of RALPH's measurements. Nevertheless, the isolated tracker loses track about halfway through the sequence, which shows that RALPH helps the system keep track. More significant, however, is the large effect of ego-vehicle yaw ψ_b on the position of the tracked car in the image. Because RALPH has an accurate measurement of the lateral offset d_b, the integrated filter is now able to deduce ψ_b. For example, at $t = 770$ the estimated curvature is now zero, whereas the tracker in isolation mistakenly believed there was a

curve to the right, because of the large yaw to the left (see panel e). Also important is that, for the isolated tracker to work at all, we needed to force both the ego-vehicle and target offsets d_b and d_c to zero, as they are now unobservable. This points out another advantage of bringing RALPH into the picture: cars significantly swerve in their lane, and this can corrupt the curvature estimate when assumed otherwise. But accurate estimates of in-lane displacements are also important to recognize lane changes and sudden maneuvers.

6 Conclusions and Future Work

We have shown how we integrated our tracking system with the RALPH road following module. The tracking system brings precise and crisp measurements of the car in the image, and performs image stabilization. However, because it does not know about the yaw or lateral offset of the ego-vehicle, its curvature estimate can be misguided. RALPH provides the missing information, as well as a good estimate of curvature, so that the combined curvature estimate is superior to both taken in isolation. RALPH also benefits from the capability of the tracker to cope with violations of the flat earth assumption, and is corrected appropriately when this is the case. The integration of these two systems substantially increased the overall system performance.

In the same way as we integrated RALPH, we could add in more measurements available from different sources. RALPH has already been integrated successfully with road map data in as yet unpublished work. We can very easily add this to our system. Other work in progress concentrates on tracking stationary points on and alongside the road, to obtain an accurate estimate of ego-motion. Lastly, although RALPH has been treated here as a black box, much of its internal processing could be integrated with the Kalman filter in a more direct fashion, such that we gain access to the probabilistic reasoning implicit in the filter.

There are also improvements to the tracker itself that we are considering. In particular, we would like to use machine learning to optimize which features we should be tracking, rather than hand tune the parameters or use ad-hoc techniques. For the bounding box feature this can be done readily by means of the likelihood energy term E_d, whose relationship to the image could be learned from training data. Multiple car tracking with occlusion reasoning, automatic error recovery and robust initialization are other opportunities for future work.

Better car tracking, and a better idea of where the road is, allows us to do more. Accurate estimates of ego-motion provide us with better information for controlling the vehicle. We should now be able to better recognize discrete events for what they are, e.g. lane changes or obstacle avoidance maneuvers of the car ahead. Using the accurate measurements of relative position and speed of other vehicles given by the system, we can predict ahead and construct a tactical plan to maneuver through traffic. In conclusion, the integration of two distinctly different vision systems into one hybrid system gives us a substantial foundation on which to base other capabilities needed to realize fully autonomous vehicles.

Acknowledgements

This work was supported in part by USDOT under Cooperative Agreement Number DTFH61-94-X-00001 as part of the National Automated Highway System Consortium, and by the National Highway Traffic Safety Administration (NHTSA) under contract DTNH22-93-C-07023.

References

[1] C. Thorpe, "Mixed traffic and automated highways," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '97)*, vol. 2, (Grenoble, France), September 1997.

[2] E. Dickmanns, "Vehicles capable of dynamic vision," in *IJCAI-97, (Nagoya, Japan)*, 1997.

[3] D. Pomerleau, *Neural Network Perception for Mobile Robot Guidance*, Kluwer Academic Publishing, Boston, MA, 1994.

[4] D. Pomerleau and T. Jochem, "Rapidly adapting machine vision for automated vehicle steering," *IEEE Expert* **11**, April 1996.

[5] F. Dellaert and C. Thorpe, "Robust car tracking using Kalman filtering and Bayesian templates," in *Proceedings of SPIE: Intelligent Transportation Systems*, vol. 3207, October 1997.

[6] P. Maybeck, *Stochastic Models, Estimation and Control*, vol. 1, Academic Press, New York, 1979.

[7] M. Schmid, "An approach to model-based 3-d recognition of vehicles in real time by machine vision," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '94)*, vol. 3, (Munich, Germany), September 1994.

[8] D. Terzopoulos and R. Szeliski, "Tracking with Kalman snakes," in *Active Vision*, A.Blake and A.Yuille, eds., pp. 3–20, MIT Press, Cambridge, MA, 1992.

[9] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *International Journal of Computer Vision* **8**(2), pp. 99–111, 1992.

[10] F. Dellaert, "CANSS: A candidate selection and search algorithm to initialize car tracking," Tech. Rep. CMU-RI-TR-97-34, Robotics Institute, Carnegie Mellon University, 1997.