Real-time Road Tracking using Templates Matching

Miguel Angel Sotelo* & Alexander Zelinsky ** & Francisco Javier Rodríguez* & Luis Miguel Bergasa*
* Department of Electronics, University of Alcalá,
Alcalá, Campus Universitario s/n 28805, Madrid, Spain.
e-mail:sotelo,fjrs,bergasa@depeca.alcala.es
http://www.depeca.alcala.es/personal/~masv
** Robotic Systems Lab.,
Department of Systems Engineering, ANU.
Canberra, ACT 0200, Australia.
http://wwwsyseng.anu.edu.au/rsl/

Keywords:real-time tracking, templates matching, least-squares estimation.

Abstract

This paper describes a vision-based system for autonomous navigation in outdoor environments. Road tracking tasks are performed in the context of templates matching. In order to achieve a high performance several templates of road edges are precomputed and correlated with the incoming image in real time. The resulting points found by the algorithm are filtered using a least-squares approach.

1 Introduction

One of the most challenging issues in the field of autonomous navigation is road tracking in outdoor environments. For many years, researchers building mobile robots have concentrated on applications involving hazardous environments. It has been clearly stated in the last years that one of the most hazardous environments is the automobile expressway. The problem is to control speed and steering using a single video camera so that the vehicle remains on the road. There are two main requirements for road following for autonomous vehicles. First, the road following behaviour must be able to operate on very different types of roads. Second, road following must be fast enough to drive the vehicle at normal speeds (e.g., 80 km/h on highways). That's the main rationale for the real-time approach using templates matching. This requirement puts severe constraints on the type of algorithms used for road following.

2 Previous work

Previous work apply vision-based tecniques for detecting certain characteristics in the image, like for instance, lanemarks (Dickmanns et al., 1994). Others are based on colour (Turk et al., 1988) or texture (Thorpe, 1990) features. An alternative approach considered in the NAVLAB project in the Carnegie Mellon University combines vision and learning tecniques (neurally inspired) to compute the characteristics that properly describe the path along the road (Pomerleau, 1993). Included in the Prometheus III project (EUREKA programme) of the European Community (Dickmanns et al., 1994), a Mercedes 500 SEI car (VaMoRs-P) was equipped with a complex sensor system (4 colour cameras, three inertial sensors,etc) and a sophisticated processing system (60 transputers and several PC 486's) with the aim of driving the vehicle along motorways at high speed. Within the PATH programme in the American State of California, with the support of the Institute of Transportation Studies of the University of California in Berkeley, extensive work has been carried out since 1986 on autonomous vehicles steering. Magnetics sensors buried under the road are used to facilitate lateral control of the vehicle along with a radar for obstacles avoidance.

3 Vision System

The MEP tracking colour vision system manufactured by FUJITSU is designed for real-time tracking of multiple objects in the frames of a NTSC video stream running on VxWorks. It is able to track over 100 templates at video frame rate (30 Hz for NTSC). The formula for the distortion D indicating how well a template fitted the video image is shown in equation 1 where *Size* is the template size (8 or 16), RGB is 3 and stands for the three colour components, $g_{ti}(x,y)$ is the grey value of the pixel in the Red, Green or Blue component in the last frame, m_x

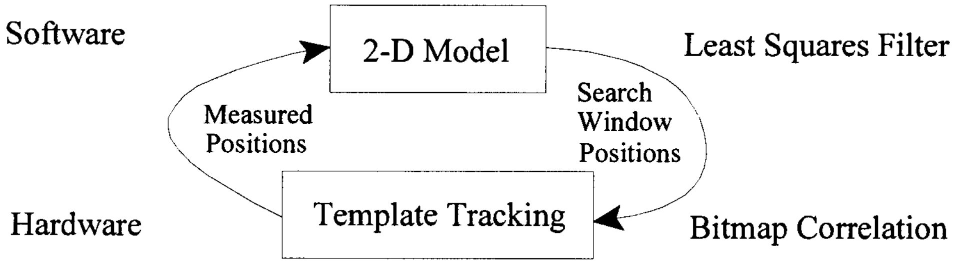

Figure 1. Two layered road tracking system.

and m_y are the magnifications of the template in X and Y direction and o_x and o_y are the offsets in the frame.

$$D=\sum_{x=0}^{Size}\sum_{y=0}^{Size}\sum_{i=0}^{RGB}|g_{ti}(x_t,y_t)-g_{fi}(x_f,y_f)| \tag{1}$$

where $x_t = x \cdot m_x \qquad y_t = y \cdot m_y$

$x_f = x \cdot m_x + o_x \qquad y_f = y \cdot m_y + o_y$

To track the template of an object it is necessary to calculate the distortion not only at one point in the image but at a number of points within the search area. To track the motion of an object the tracking module finds the position in the search window where the template matches with the lowest distortion. By moving the search window along according to the tracking results any object can be easily tracked. This method works perfectly for templates representing objects that do not significantly change their appearance or shade.

4 Feature tracking

Pose estimation is strongly interconnected with robust feature tracking. Obviously pose estimation depends completely on the ability of the system to track certain road features reliably. But the feature tracking itself is error-prone without verification and using forward estimation based on higher level information. Features can become temporarily occluded or distorted and they may be ambiguous making the position recovery more difficult after a tracking failure. Even the decision of whether a feature is tracked correctly or not can not be made reliably without making use of higher-level information. Using correlation data only is insufficient. Thus, the information flow from top to bottom is crucial for the robustness of a road lanemark tracking system. The approach used in this work is a two layered system illustrated in figure 1. On the lowest level the vision system performs bitmap correlation in hardware. The results are the measured feature positions that may contain tracking errors. The measured positions are

forwarded to the 2-D model based on a least squares approach, which takes geometric constraints into account in the image plane and the correlation distortion to generate an estimate of the road edge equation. This layer is implemented as a Weighted Recursive Least-Squares Filter with Exponential Decay (Schneiderman. H, 1994). The estimated positions of features determine the location within the next image frame of the hardware search windows. All two layers run at 30Hz. The system achieves fast and robust performance by combining fast template tracking and real-time adaption of geometric constraints by 2-D pose estimation.

5 2-D Model of the road for lane marker tracking

The tracking algorithm requires that lane markings be present in the image. There exist two sucessive stages of computation.

1. Edge Extraction and Data Association. Extracting edge point position and determining likely groupings of edge points for the lane of travel.

2. Model update. Updating the lane marker models.

These sequence of operations is repeated for each new image. Lane markers representations are maintained in 2D with respect to the image plane throughout all computations in this algorithm. Only the right lane markings are modelled. These markings correspond to the white or yellow lines painted on the road.

5.1 Representation of lane markers

The lane marker boundary is modelled by a second order polynomial in the image plane.

$$j=a_1 i^2+a_2 i+a_3 \qquad (2)$$

where j is the column and i is the row, a_1, a_2 and a_3 determine the position and shape of the edge model. A second order model was chosen to provide adequate representation of shape within the real time constraints required for autonomous navigation.

5.2 Initial Conditions

The algorithm requeres initially an aproximate model of the lane markings before tracking can start. In order to solve this problem an operator should provide the system with the initial model according to the position of the real lane markings in the road image.

5.3 Edge Extraction and Data Association

In this step edge extraction is performed on each image. A sufficiently representative set of 16 different 16x16 templates of road edge are off-line stored on memory. They are used to search for the possible edge points in the current image. Each template is correlated with a reduced surrounding environment of the last lane markings. The resulting point where the distortion is minimum is annotated as a raw edge point, for each template.

A data association algorihm is used to determine which of these raw edge points are likely to be associated with the lane marker, discarding the rest of points. The algorithm compares each edge pixel to the current model of the lane marker. An edge pixel must satisfy two criteria to be associated with a lane marker. The first criterion is two-dimensional spatial proximity of the edge point to the model. The second criterion is low distortion value after the correlation is performed. For all raw edge points falling within the window of interest around the previous lane model, the two data association criteria are applied on a point by point basis.

5.4 Model updating

Several principles are followed to obtain robust lane marker updates.

1) More Data Improve the Estimate: In general, an estimate can be improved by using more data. It can be shown that if a measurement consists of a sum of a stationary signal and unbiased noise, the estimate of the signal will improve, i.e, the variance in the estimate will decrease, as more measurements are averaged. To achieve higher robustness all the visible portions of the lane marker in the image must be used to compute edge points.

2) Uncertainty in Road Change: The lane markers are not strictly stationary signals across successive images. They change relatively slowly assuming a nominal vehicle speed. A trade-off must be achieved between robustness of the estimate, by using data over a large temporal span, and responsiveness to actual changes in the lane marker. This compromise is solved by the relative weighting of new data with respect to older data in the estimate, governed by an exponential decay factor, where the weight contributed by the exponential decay, λ, for each edge point is λ^{t-to}, where $0.0<\lambda<1.0$, t is the current time and t_0 is the time the image was sampled.

3) Uncertainty in Lane Marker Visibility. In each image, lane marker visibility is measured by the number of edge points matched to the lane marker model. This measure of visibility can also be used as an additional weight. So, for instance, when a small gap of lane markers is present in the image the data association algorithm finds few edge points and these images will therefore carry relatively less weight on the estimates and do not greatly perturb the estimated lane marker model.

The following cost function is minimized for updating the model:

$$J_R=\sum_{p=0}^{t}\left(\lambda^{t-p}\sum_{a=1}^{N_p}\left[j_{p,a}-(a_3+a_2 i_{p,a}+a_1 i_{p,a}^2)\right]^2\right) \qquad (3)$$

where t corresponds to the current image and N_p is the number of edge points encountered in the image p. The influence of each term in equation 3 is given by the number of edge points obtained for that image. On the other hand, the cost function also accounts for previous images to achieve a robust estimate. An exponentially decreasing parameter λ, varying between 0 and 1, is incorporated to weight the influence of past images on the final model. Small values of λ minimize the influence of previous images, decreasing the robustness of the model and making it more sensitive to changes. A dynamic adjustment of λ is essential to avoid the problem of saturation on straight stretches. λ is governed by the following expression:

$$\lambda = a_1 \alpha + \lambda_{min} \tag{4}$$

On straight stretches a_1 is practically 0 providing the algorithm with the ability to detect curves. The model is recursively updated in the state space, in three consecutive calculations.

a) The edge points estimated by the model are computed:

$$\hat{z}(n) = H(n)x(n-1) \tag{5}$$

b) Estimated covariance updating:

$$P(n) = \frac{1}{\lambda}[P(n-1) - K(n)H(n)P(n-1)] \tag{6}$$

where

$$K(n) = P(n-1)H'(n)(\lambda I + H(n)P(n-1)H'(n))^{-1} \tag{7}$$

c) State updating:

$$x(n) = x(n-1) + K(n)[z(n) - \hat{z}(n)] \tag{8}$$

where

$$z(n) = \begin{bmatrix} j_{n,1} \\ j_{n,2} \\ \cdots \\ j_{n,N_n} \end{bmatrix} \quad H(n) = \begin{bmatrix} 1 & i_{n,1} & i_{n,1}^2 \\ 1 & i_{n,2} & i_{n,2}^2 \\ \ldots & \ldots & \ldots \\ 1 & i_{n,N_n} & i_{n,N_n}^2 \end{bmatrix} \quad x(n-1) = \begin{bmatrix} a_3 \\ a_2 \\ a_1 \end{bmatrix} \tag{9}$$

x(n-1) represents the estimated state at instant *n-1*. P(n-1) represents an estimate of the states covariance. This filter is known as Weighted Recursive Least-Squares with exponential decay.

6 Experimental results

The road tracker was tested with recorded video sequences to evaluate the performance and robustness of the system. Several practical tests were performed with 10, 15 and 20 different templates (16x16 pixels) of road lanemarks obtained along a 3 km road strech. They were precomputed and stored in memory before starting the real-time tracker. Eventually it was empirically demonstrated that 15 different templates were sufficient to achieve good tracking results without losing real time performance. Figure 2 shows a sequence of images computed by the system, in which it can be clearly seen that the road edge is being tracked by the algorithm in a curved section. After the state model is updated the new search windows are located around the current edge points encountered by the least squares filter. Parameter λ_{min} was empirically set to 0.2 while α takes the value $0.8/a_{1,max}$, where $a_{1,max}$ is the maximum value found for parameter a_1 after several practical trials.

7 Conclusions and future work

We have designed a real-time templates-based road tracker in the context of state space filtering. Two aspects have been emphasised in this work. One is the real-time capability of the vision system to track the road edge and the other is the robustness of the tracking to tolerate occlusions of the lanemark for short intervals of time, because of its filtering nature.

This road tracking technique is intended to be tested in future on the real carlike robot shown in figure 3. The vehicle will carry out missions in a real environment achieving a global navigation behaviour by performing local navigation tasks (connected to a global planner). One of those tasks will be tracking a road based on computer vision.

8 Acknowledgments

This work has been funded by The Australian National University and the Spanish C.I.C.Y.T. as well as by the generous support of Wind River Systems, supplier of VxWorks.

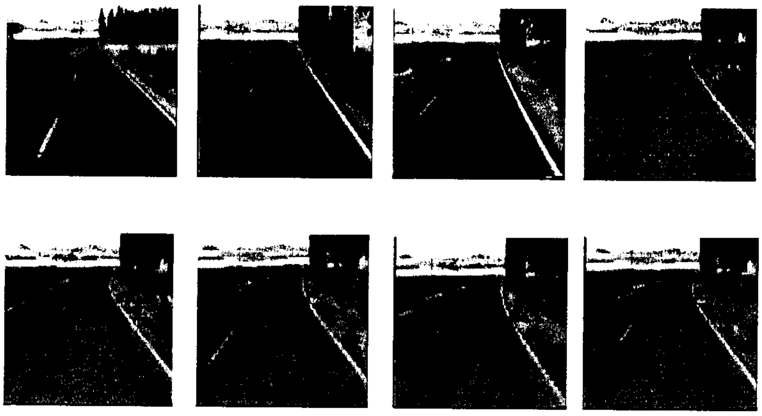

Figure 2. Sequence of road images.

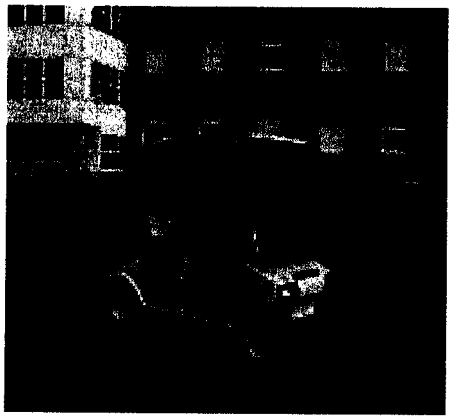

Figure 3. Carlike Robot for future work.

9 References

Crisman, J. D. and Thorpe, C. E. 1988. Color Vision for Road Following. In Proc. of Mobile Robots III.

Dickmanns, E. D., Behringer, R., Dickmanns, D., Hildebrant, T., Mauer, M., Thomanek, F., and Shielhlen, J. 1994. The seeing passenger car 'VaMoRs-P'. In Proc. Of Int. Symp. On Intelligent Vehicles, Paris.

Hebert, H., Thorpe, C and Stentz, A. 1997. Intelligent Unmanned Ground Vehicles. Autonomous Navigation Research at Carnegie Mellon. Kluwer Academic Publishers.

Heinzmann, J., and Zelinsky, A. 1997. A Visual Interface for Human-Robot Interaction. Proceedings of the International Conference on Field and Service Robotics. Canberra, Australia. 8-10 December 1997.

Jochem, T, Pomerleau, D and Thorpe, C. 1995. Vision guided lane transition. Proceedings of the 1995 IEEE Symposium on Intellignet Vehicles, September 25-26, Detroit, Michigan, USA. 1995.

Koller, D., Luong, T., and Jitendra, M. 1995. An integrated stereo-based approach to automatic vehicle guidance. In Proc. Int. Conference on Computer Vision. Boston.

Pomerleau, D. 1993. Neural Network Perception for Mobile Robot Guidance, Kluwer Academic Publishers: Boston.

Rodríguez, F. J., Mazo, M., and Sotelo, M. A. 1998. Automation of an Industrial Fork Lift Truck, Guided by Artificial Vision in Open Environments. Autonomous Robots 5, 215-231.

Sarkar, N., Yun, X., and Kumar, V. 1994. Control of mechanical systems with rolling constraints: Application to dynamic mobile robots. Int. Journal of Robotics Research, 13(1).

Schneiderman, H. and Nashman, M. 1994. A discriminating feature tracker for vision based autonomous driving. IEEE Trans. On Robotics and Automation. 10(6):769-775.

Stentz, A. and Hebert, M. 1995. A complete navigation system for goal acquisition in unknown environments. Autonomous Robots, 2, 127-145.

Thorpe, C. 1990. Vision and Navigation: The Carnegie Mellon Navlab, Kluwer Academic Publishers: Boston.

Turk, A.M., Morgenthaler, D., Gremban, D., and Marra, M. 1988. VITS- A vision system for autonomous land vehicle navigation. IEEE Trans. On Pattern Analysis and Machine Intelligence, 10(3): 342-361.