VISUAL LANE TRACKING AND PREDICTION FOR AUTONOMOUS VEHICLES

VIVACQUA, RAFAEL P. D., MARTINS, FELIPE N.

NERA – Núcleo de Estudos em Robótica e Automação,
Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo – Campus Serra
Rod. ES-010, km 6,5, Manguinhos, Serra, ES - 29173-087
E-mails: rafsat@ifes.edu.br, felipemartins@ifes.edu.br

VASSALLO, RAQUEL F.

LAI - Laboratório de Automação Inteligente,
Depto. de Engenharia Elétrica, Universidade Federal do Espírito Santo – Campus Goiabeiras
Av. Fernando Ferrari, 514, Goiabeiras, Vitória, ES - 29075-910
E-mail: raquel@ele.ufes.br

Abstract— We propose a visual method to be used in autonomous vehicles to track the position of horizontal road lane marks. Our method is based on filtering with prediction. The prediction step estimates the expected lane mark position in each new image frame. It is also based on the kinematic model of the car and on the information generated by the embedded odometric sensors. Experimental results obtained using a properly prepared test vehicle demonstrated that the prediction step is responsible for a significantly tracking error reduction under certain conditions, like oscillations and lane change. Therefore, we believe that our method can improve system performance when applied to image-based controlled autonomous vehicles.

Keywords— Autonomous vehicles, Computer vision, Lane tracking.

Resumo— Propõe-se um método visual a ser utilizado em veículos autônomos para seguir a posição das faixas horizontais de estradas. O método baseia-se em filtragem com predição. A etapa de predição provê uma estimativa da posição esperada da faixa da estrada a cada novo quadro de imagem. Tal etapa baseia-se também no modelo cinemático do carro e na informação gerada pelos sensores odométricos incorporados. Os resultados experimentais obtidos utilizando um veículo de teste adequadamente preparado demonstraram que há redução significativa do erro de seguimento de faixas em certas condições, como oscilações e trocas de faixa. Portanto, acreditamos que o método aqui proposto pode melhorar o desempenho do sistema quando aplicado a veículos autônomos controlados com base em imagem.

Palavras-chave— Veículos autônomos, Visão computacional, Seguidor de pista.

1 Introduction

To make a car drive itself is certainly one of the greatest challenges of modern automation. On one side, auto-pilots for airplanes and ships are well known and widely used. On the other side, due to the much greater complexity, for cars and trucks this technology is still being developed. The goal is to make the car drive autonomously from address A to address B, as safely as (or safer than) an experienced human driver. Under these conditions, the user becomes just a passenger.

For a car, a fundamental requirement of an autonomous driving system is the ability to keep the vehicle within the road and drive in the center of its lane. This requires knowledge of the surrounding environment, which is obtained through sensors installed in the car, such as camera, LASER scanning, GPS, and inertial units, for example.

Projects like (NASHMAN, 1992), (SCHNEIDERMAN, 1994), (POMERLEAU, 1996), (BROGGI, 1999) and (LU, 2002) rely only on a gray-scale images generated by a camera for detecting the position of the road and its estimate curvature. In (CHRISMAN, 1998) and (SALES, 2010) color images were used to classify the regions belonging to the road. This idea was also applied in (DAHLKAMP, 2006) in order to extend the range of the system beyond the range of the LASER range-finder (LRF) sensors used on their autonomous the vehicle.

Stereo vision systems allow the calculation of distance of the objects on the scene. A binocular vision system (stereo vision) was used in (BROGGI, 2010) and (LIMA, 2010) for obstacle detection. But, this kind of system has the drawback of being of high computational cost.

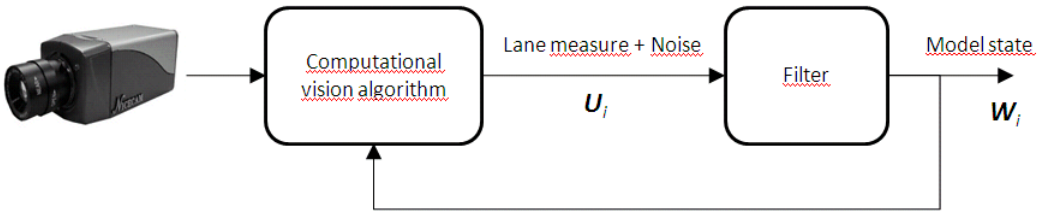

Figure 1: Visual information flow to generate the model state.

Projects like (GUIZZO, 2011) and (autonomous LABS, 2012) use the LRF sensor in conjunction with cameras, GPS, IMU and map information to lead autonomous driving research to an unprecedented level. Missions of autonomous navigation in real urban environment have been completed successfully by both projects.

In Brazil this field of research is starting to develop and we highlight some: Driving 4 You, from UNIFEI (VEERMAS et al., 2010), CADU, from UFMG (LIMA, 2010), VERO, from CTI / Campinas (VERO, 2012), SENA, from EESC / USP (SENA, 2012), and CARINA (CARINA, 2010), developed at ICMC / USP. Another group in our university (UFES) is also developing an autonomous car project. They bought a car with the main hardware components already installed, and their main focus is on the software side. Their main goal is to understand how the human brain interprets the world using images (LCAD, 2013). On the other hand, the main objective of our project is to build an autonomous car with simple and low computational costs solutions. We are adapting a regular car by installing actuators and sensors in order to achieve this goal. Our car is already able to drive itself on real roads, under controlled conditions (VIVACQUA et al., 2012).

Our system consists on a notebook computer that is connected to a USB camera and an interface board. The USB camera is installed on the front windshield, pointing towards the road. The interface board connects the computer to the car encoders, and to the step-motor that is responsible for the driving wheel positioning.

The visual information generated by the computational vision system is subjected to noise or errors caused by occlusions, reflexes, shadows etc. To minimize the effect of these problems a filtering process is executed over the raw image to produce more reliable information. This information is the base for the so called model state, and is fed back into the vision algorithm to define the image region of interest (ROI) (see Fig. 1). In (VIVACQUA et al., 2012) this information is also used by the controller to keep the autonomous car driving in the center of the road.

Some methods for visual lane track have been proposed (SCHNEIDERMAN, 1994), (AUFRERE, 2000), (LU, 2002) and (CHOI, 2012). All of them use some kind of filtering process to update its model state (parameters that descript the lane marks), but none use prediction. To implement prediction, vehicle displacement information must be accessible and reliable. We have made the necessary adaptations in our autonomous car and in this paper we present our proposal for a visual method to track the position of horizontal road lane marks, which is based on filtering with prediction. The prediction step estimates the expected lane mark position in each new image frame in order to minimize tracking error. The following sections present an explanation of our system and discuss some experimental results.

2 Reference Systems

Before making the prediction step, it is necessary to change the system reference frame from the camera's frame to the car's reference frame, considering the camera in the origin (own reference – OR). The horizontal lane marks detected by the vision system that is being used in the autonomous car (VIVACQUA, 2012) are mapped through inverse perspective projection (Fig. 2) as if the image was seen from the sky, assuming that the road is perfectly plane (bird's eye view).

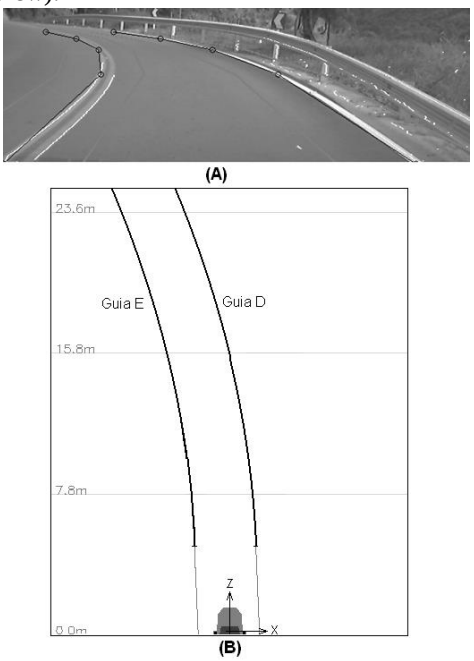

Figure 2. Reference system: (A) Camera frame; (B) Car frame.

In the OR frame, the origin is the point exactly below the camera in the ground plane, the Z axis corresponds to the car displacement direction (while moving forward), and the X axis is the transversal direction. Figure 2 (B) shows two guide lines (Left and Right) corresponding to the detected information in the image of Figure 2 (A). The basic idea of the prediction step is that it is possible to predict the car position in the next image frame based on its kinematic model and on the lane marks positions on the actual frame.

3 Car kinematic model

The car kinematic model can be approximated by a bicycle model presented in Figure 3. In this Figure:
$\mathbf{P_r}$ is the center of the circumference that describes the car trajectory;
$\mathbf{P_c}$ is the camera position in the vehicle (OR origin);
R is the trajectory curvature radius described by P_c;
L_t is the distance between the car front and rear axes;
L_r is the distance between the OR origin and the rear axis;
α is the front wheel turning angle.

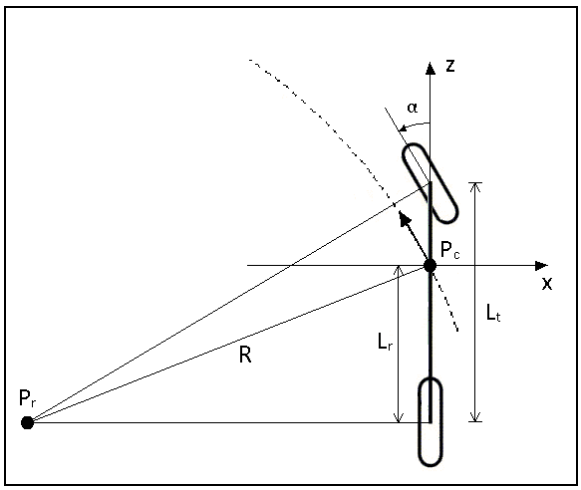

Figure 3: Car kinematic Model.

Using the car kinematic model (Eq. 1) we can calculate the vehicle trajectory in the world reference frame (WR) given its parameters and the front wheel turning angle (α).

$$\begin{aligned} \dot{x} &= v\cos(\theta) \\ \dot{z} &= v\sin(\theta) \\ \dot{\theta} &= \frac{v}{L_t}\tan(\alpha) \end{aligned}, \qquad (1)$$

where *x*, z and θ are the position and orientation of the vehicle in WR, and *v* is the car linear velocity. It is important to note that the origin of WR is fixed in the ground and not in the car. Thus, as the car moves, the point P_c describes the dotted trajectory, and the position of fixed objects in the world is kept constant.

4 Position Prediction

In the OR reference frame the point P_c coincides with the origin and the fixed objects in the world turn around P_r, but in reverse direction.Figure 4 presents the initial position of a fixed object in the world seen in the OR reference system. It is possible to estimate the future position based on the information of initial position and the trajectory defined by the kinematic model.
The rotation Center ($\boldsymbol{P_r}$), the curvature radius of the camera trajectory (R) and the angular displacement ($\Delta\theta$) are calculated using (2), (3) and (4), that make use of the car odometric information.

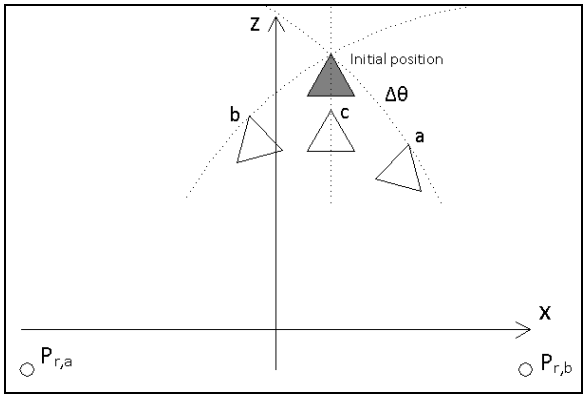

Figure 4: Prediction of an object position: (a) Right corner; (b) Left corner; (c) Straight.

The distance travelled in meters (obtained through an encoder localized in the front axis), and the front wheel turning angle (α) (obtained through another encoder installed in the driving wheel axis):

$$\boldsymbol{P_r} = (L_t \cot \propto ; -L_r), \qquad (2)$$

$$R = \sqrt{(L_t \cot \propto)^2 + L_r^2}, \qquad (3)$$

$$\Delta\theta = \frac{\Delta L}{R}, \qquad (4)$$

where ΔL is the travelled distance.

5 Rotation of a Parabola

To estimate the new position of any object, one just has to know the rotation center ($\boldsymbol{P_r}$), the rotating angle ($\Delta\theta$), and apply a 2D rotation operation. In the specific case of the autonomous vehicle described in (VIVACQUA, 2012), where the system was tested, the tracked objects are the two road lane marks that compose the road model. Each one of this marks is represented by a 2nd degree polynomial.

.

$$x(z) = a_2 z^2 + a_1 z + a_0, \qquad (5)$$

Given a point (*X*, *Z*) of the rotated parabola, we can find the corresponding point (*x*, *z*) in the original parabola through the rotation operation in the reverse direction, given by

$$x = (X - P_x)\cos\Delta\theta - (Z - P_z)\sin\Delta\theta + P_x, \quad (6)$$
$$z = (Z - P_z)\cos\Delta\theta + (X - P_x)\sin\Delta\theta + P_z, \quad (7)$$

where P_x and P_z are the coordinates of the rotation center. Substituting (6) and (7) in (5) we get to the general form of the rotated parabola:

$$X^2 + c_1Z^2 + c_2X + c_3Z + c_4XZ + c_5 = 0. \quad (8)$$

This form is incompatible with the original data structure used in the software and do not allow the explicit calculation of X given Z, which is computationally undesirable. To circumvent this problem, we propose a method to find a good approximation to the rotated parabola keeping the default form given in (5).

Consider the rotated parabola equation

$$x_r(z) = a_{2r}z^2 + a_{1r}z + a_{0r}. \quad (9)$$

The problem we need to solve is to find the coefficients a_{2r}, a_{1r} and a_{0r}. The general idea to solve this problem is: Find the central point ***M*** of the original parabola *x(z)* in the interval $[z_0..z_1]$; apply a rotation of $\Delta\theta$ around the point $\boldsymbol{P_r}$ (thus making *M* turns into $\boldsymbol{M_r}$); Consider that $\boldsymbol{M_r}$ belongs to the new parabola $x_r(z)$; and consider that the direction of the tangent in this point was subtracted by the rotation angle $\Delta\theta$.

The inclination of the tangent line in the point ***M*** is given by $x' = 2a_2z_M + a_1$. Using the property of tangent difference, we have: $x'_r = \frac{x' - \tan\Delta\theta}{1 + x'\tan\Delta\theta}$.

As the rotated parabola curvature is not altered, we have $a_{2r} = a_2$. The term a_{1r} can be calculated by $a_{1r} = x'_r - 2a_{2r}z_{Mr}$, and the term a_{0r} is calculated by $a_{0r} = x_r - a_{2r}z_{Mr}^2 - a_{1r}z_{Mr}$. By doing so, we have the rotated parabola parameters in the OR reference frame.

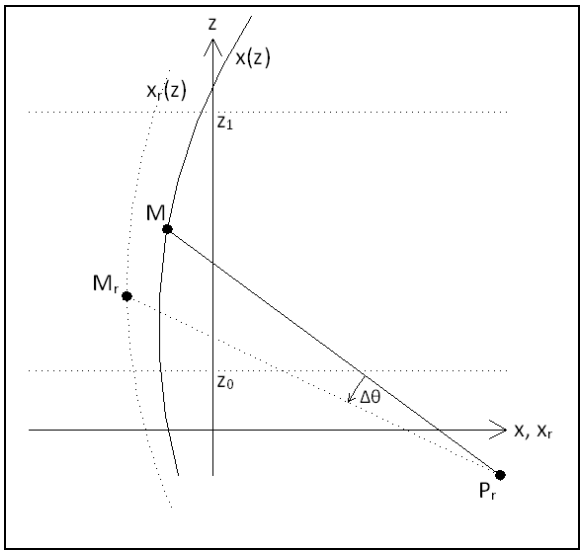

Figure 5: Approximation for the rotated parabola.

6 Filtering

We are considering that the lane mark position cannot change much in consecutive frames, and a great amount of noise lead to fast image variation (such as reflection or occlusion). For this reason, we chose to use a first order low pass filter, which time constant was defined experimentally taking into account its influence on the system behavior: if the val-

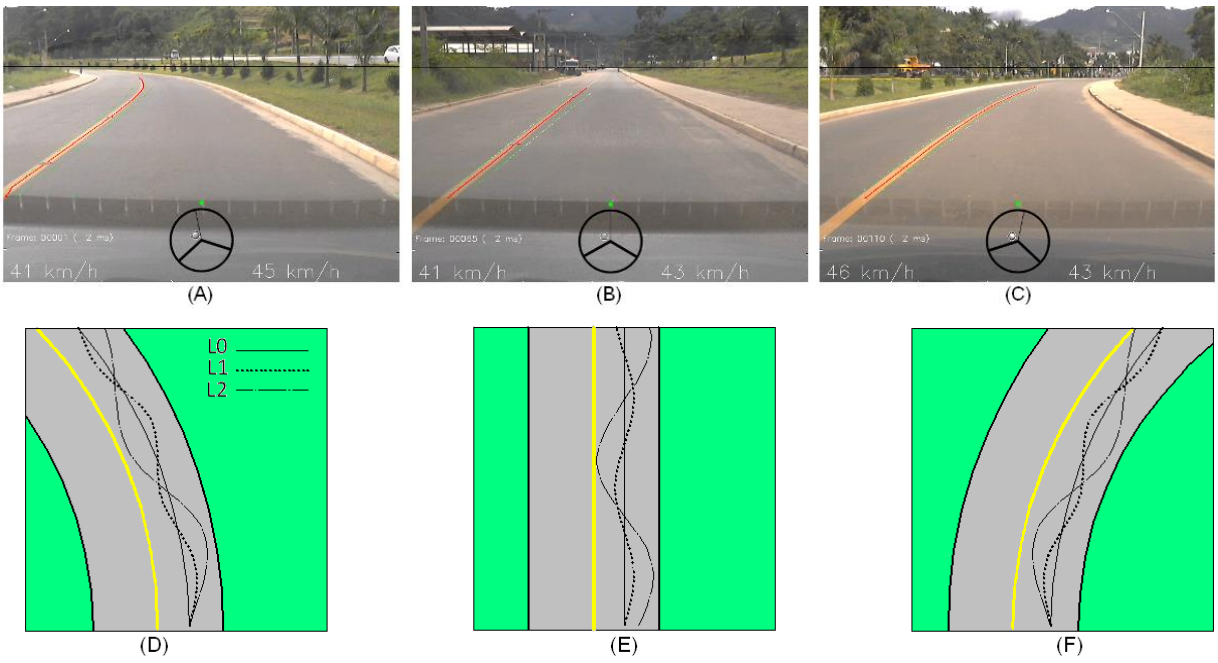

Figure 6: Experiment conditions: (A) Left corner; (B) Straight; (C) Right corner; (D, E, F) Corresponding paths.

ue is too low the filtering would have little effect in attenuate the sporadic erroneous measures. On the other hand, if it is too high, this would delay the response, and would cause difficulties to the model in following the slow changes in the road shape that occurs before and after corners, for example. After testing with videos captured with the vehicle camera at the speed of 80km/h, we found that the value of $\tau = 3$ produced a good compromise between robustness and response velocity.

Equation (10) shows vector $\boldsymbol{W}_i$ that represents the state model. Equation (11) is the expression for the low pass filter that updates the model state from the measures generated by the vision system ($\boldsymbol{U}_i$).

$$\boldsymbol{W}_i = \{a_2, a_1, a_0\}^T, \quad (10)$$

$$\boldsymbol{W}_{i+1} = \boldsymbol{W}_i + \frac{1}{\tau}(\boldsymbol{U}_i - \boldsymbol{W}_i), \quad (11)$$

7 Experimental Results

To measure the performance of the prediction system, we performed some experiments using a test vehicle properly adapted to acquire the road images synchronously with the odometric measurements. This vehicle was driven in three curvature conditions as Figure 6: (A) In a left corner; (B) In a straight line; and (C) in a right corner. In each curvature condition, the vehicle was conducted in three different modes regarding the oscillation level of the trajectory (Figures 6 (D) to 6 (F)): Without oscillation (L0), where the vehicle travels the road parallel to the guide line; with low level of oscillation (L1) where the vehicle crosses cyclically the central line; and with high level ofoscillation (L2) where the vehicle also crosses the central line but with a more accentuated deviation.

The oscillation was introduced in the trajectory to force the vehicle to move not parallel to the road line so that the prediction could demonstrate its capacity in keeping the model state close to the instantaneous measures.

The tests were conducted in approximate speed of 40 km/h in a 100m length path and the data (images and odometry) was acquired each 40ms. Figure 7 shows the registered driving wheel position for each of the trajectories of Figure 6.

The collected data was analyzed in the laboratory to evaluate the system performance comparatively (with and without use of prediction). The performance index used was the absolute horizontal average error (AHAE) calculated as a difference of the curve defined by the model and the one defined by the instantaneous measure.

$$AHAE = \frac{1}{n}\sum_{i=1}^{n}\left|x_{measur\_i} - x_{model\_i}\right|. \quad (12)$$

8 Discussion

Figure 8 presents a frame sequence taken from the video obtained in the test that corresponds to the case shown in Figure 6(D), for the L2 trajectory. This sequence corresponds to the stretch where the vehicle is moving away from the central line what can be perceived by the successive displacement of the yellow mark to the left.

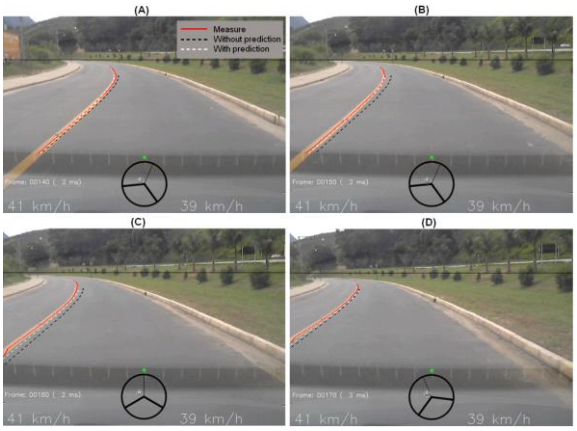

Figure 8: Frame sequence showing the tracking process with and without prediction.

Figure 8 clearly shows that the model that uses filtering and prediction (white dotted line) keeps closer to the measure (red continuous line) than the model that uses just filtering (black dotted line).This illustrates the benefit of our method.

Another way of perceiving the closer proximity between the model state and the visual measures is

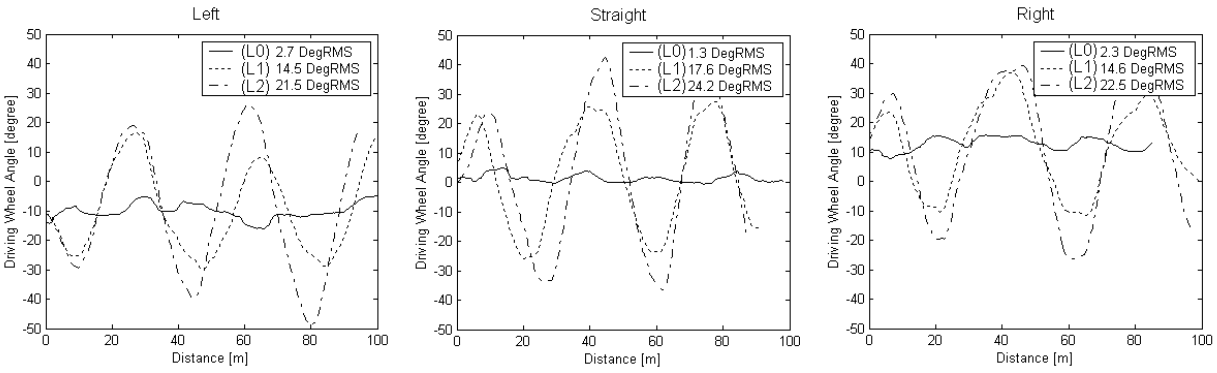

Figure 7: Driving wheel position data obtained in the tests.

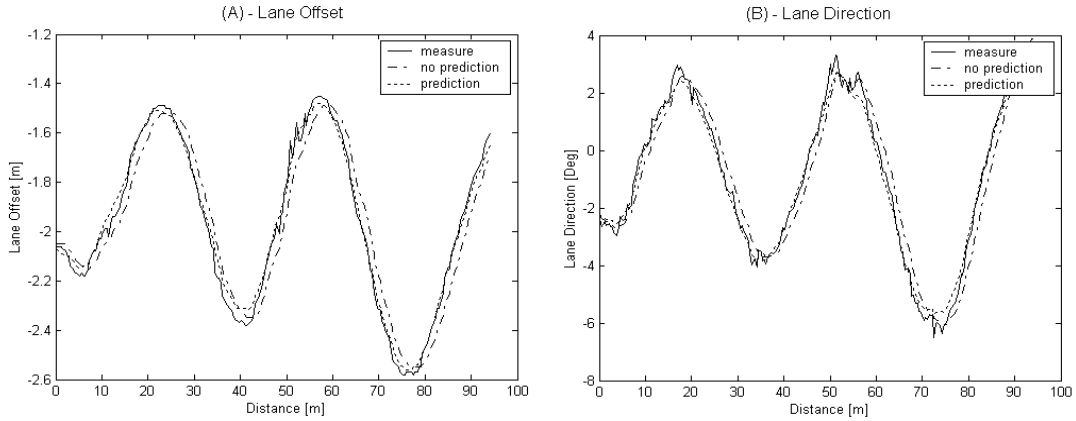

Figure 9: Only-filtered and filtered-and-predicted model parameters behavior.

through the graphs in figure 9, that present the parameters a_0(Lane Offset) and a_1(Lane Direction) of the polynomials of the filtered model and of the filtered model with prediction. The curvature parameter (a_2) was not presented because it is practically constant in the considered stretch.

In Figure 9 we can notice the ability of the low pass filtering in removing high frequency noise, which resulted in a delayed smoother curve (no prediction case). With the use of prediction it was possible to reduce the undesired delay (prediction). This reduction occurs because the prediction compensates the effect of the car movement over the captured images.

Figure 10 presents the normalized accumulated error (NAE) of the index AHAE-for the test condition with higher oscillation (L2), where the accumulated error with no prediction is considered to be 1. It is calculated with (13), where n is the total number of frames used in the experiment.

$$NAE_j = \frac{\sum_{i=1}^{j} AHAE_{pred,i}}{\sum_{i=1}^{n} AHAE_{nopred,i}}. \qquad (13)$$

In all test cases the accumulated error with prediction rise in a smaller rate (lower than 0,51) than the accumulated error without prediction.

The final value of NAE obtained in all tested conditions is summarized in Figure 11. We can notice that when the car is driven in trajectories with higher oscillation levels, the value of NAE reduces. This occurs because when there is oscillation, the lane marks displaces laterally within the camera visual field and the use of prediction compensates for this effect keeping the model closer to the real measures, while maintaining the robustness given by the filtering process. We believe that the prediction step will enhance the performance of the autonomous vehicle while driving under similar conditions.

References

Aufrere, R; Chapuis, R; Chausse, F. (2000) A fast and robust vision based road following algorithm. In IEEE Inteligent Vehicles Symposium.

Autonomos LABS. MadeInGermany drives through Berlin's traffic in the city center. Acesso em: fev/2012. Disponível em: http://www.autonomos.inf.fu-berlin.de/

Broggi, A; Bertozzi, M; Bianco, A. F. C. G; Piazzi, A (1999). The Argo Autonomous Vehicle's Vision and Control Systems. International Journal of Intelligent Control and Systems, Vol. 3, No. 4, pp. 409-441.

Broggi, A; Cappalunga, A; Caraffi, C; Cattani, F; Ghidoni, S; Grisleri, P; Porta, P. P; Posterli, M; Zani, P. (2010) TerraMax Vision at the Urban Challenge 2007. IEEE Transactions on Intelligent Transportation Systems, Vol. 11, No. 1, March 2010.

Carina (2010). Projeto CARINA. Acesso em jul/2012. Disponível em: http://www.lrm.icmc.usp.br/carina/

Chrisman, J. D. and Thorpe, C. E. (1998). Color Vision for Road Following. Proceedings of SPIE conference on

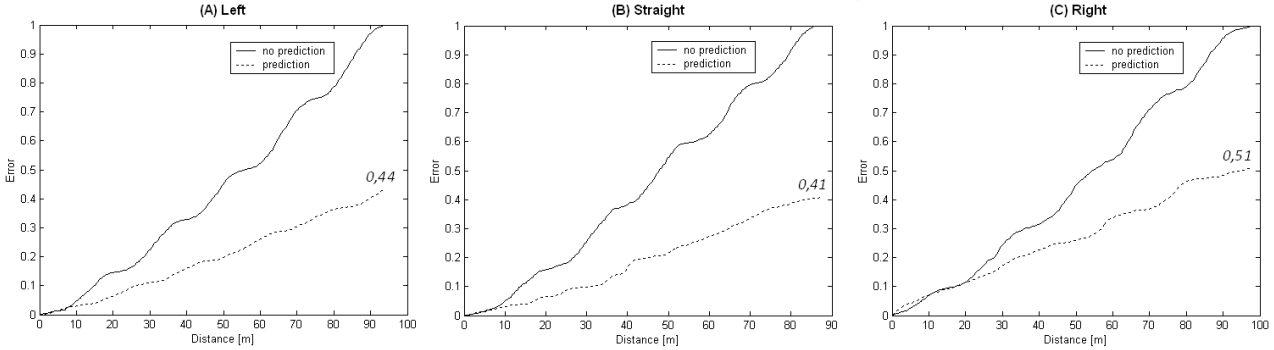

Figure 10: Accumulated error in oscillation level 2.

Mobile Robots.
Choi, H.-C.;Park, J.-M.; Choi, W.-S; Oh S.-Y. (2012) Vision-Based fusion of robust lane tracking and forward vehicle detection in a real driving environment. In International Journal of Automotive Technology, Vol. 13, No. 4, pp. 653−669.
Dahlkamp, H; Kaehler, A; Stavens, D; Thrun, S. (2006). Self-supervised Monocular Road Detection in Desert Terrain.
Guizzo, E. (2011). How Google's Self-Driving Car Works. Acesso em: nov/2011. Disponível em: http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works
LCAD – Laboratório de Computação de Alto Desempenho. Acesso em 31/maio/2013. Disponível em: http://lcadufes.wordpress.com/
Lima, D. A. (2010). Navegação Segura de um Carro Autônomo Utilizando Campos Vetoriais e o Método da Janela Dinâmica. Dissertação de Mestrado No. 643, UFMG, PPGEE.
Lu, J; Yang, M; Wang, H and Zhang, B. (2002). Vision-based Real-time Road Detection in Urban Traffic. Proceedings of SPIE, Vol. 4666.
Nashman, M. and Schneiderman, H (1992). Real-Time Visual Processing for Autonomous Driving.
Pomerleau, D. and Jochem, T. (1996). Rapidly Adapting Machine Vision for Automated Vehicle Steering. Journal IEEE Expert: Intelligent Systems and Their Applications, Vol 11, Issue 2.
Sales, D; Shinzato, P; Pessin, G; Wolf, D; Osório, F. (2010). Vision-Based Autonomous Navigation System Using ANN and FSM Control. Latin American Robotics Symposium and Intelligent Robotics Meeting
Schneiderman, H. and Nashman, M. (1994). A Discriminating Feature Tracker for Vision-Based Autonomous Driving. IEEE Transactions on Robotics and Automation, Vol. 10, No. 6, pp. 769-775.
Sena (2012). Projeto sena. Acesso em: jan/2012. Disponível em: http://www.eesc.usp.br/sena/url/pt/index.php
Veermas, L. L. G., Honório, L. M., and Vidigal, M. C. A. C. F. (2010). Uma metodologia para aprendizado supervisionado aplicada em veículos inteligentes. In Congresso Brasileiro de Automática - CBA, pages 1–6.
Vero (2012). Projeto vero - veículo robótico. Acesso em: jul/2012. Disponível em: http://www.cti.gov.br/index.php/drvc-projetos/vero-veiculo-robotico.html
Vivacqua, R.P.D.; Martins, F.N., Vassallo, R.F. (2012) Direção autônoma de um veículo baseada em visão computacional. In Congresso Brasileiro de Automática - CBA.