*Int. J. Appl. Math. Comput. Sci.*, 2025, Vol. 35, No. 1, 179–189
DOI: 10.61822/amcs-2025-0013

amcs

# PROBABILISTIC LANE SEGMENTATION USING A LOW–DIMENSIONAL LINEAR PARAMETRIZATION

CARLOS ACUÑA [a,b], GUSTAVO ARECHAVALETA [a], MARIO CASTELÁN [a,*]

[a]Robotics and Advanced Manufacturing Group
Center for Research and Advanced Studies of the National Polytechnic Institute
1062 Industria Metalúrgica, Ramos Arizpe, 25900, Mexico
e-mail: {garechav,mario.castelan}@cinvestav.edu.mx

[b]School of Engineering and Sciences
Monterrey Institute of Technology
2501 Eugenio Garza Sada Sur, Monterrey, 64700, Mexico
e-mail: carlos.acunaocampo@tec.mx

Lane detection is an important module for active safety systems since it increases safety and reduces traffic accidents caused by driver inattention. Illumination changes or occlusions make lane detection a challenging task, especially if the detection is performed from a single image. Consequently, this paper presents a probabilistic approach based on the Kalman filter, which uses information from previous image frames to estimate the lane that could not be detected in the current image frame, considering uncertainty in the prediction as well as in the detection. To this end, a principal component analysis of the segmented curvature is introduced with the purpose of dimensionality reduction, moving from a large dimensional pixel representation to a considerably reduced space representation. Furthermore, the proposed approach is compared with a fully connected pretrained CNN model for lane detection, demonstrating that the proposed method has a lower computational cost in addition to a smoother transition between lane estimates.

**Keywords:** lane detection, Kalman filter, dimensionality reduction.

## 1. Introduction

Autonomous robotics is aimed at perceiving the physical world through computer-controlled mechanical devices in order to navigate accordingly (Thrun *et al.*, 2005). The reliability of autonomous navigation depends directly on the ability to perceive the environment. In autonomous vehicles, advanced driver assistance systems have been incorporated for alerting in dangerous situations or for taking an active part in driving. The main bottleneck in the development of such systems is the perception problem (Thorpe *et al.*, 1991), which has two elements: lane and obstacle detection.

Lane detection and tracking have been investigated from different viewpoints due to their complexity and the distinction of the problems presented by each. Lane detection involves determining the location of lane boundaries in a single image without solid prior knowledge about the position of the lane (Kreucher *et al.*, 1998). Among the important challenges associated with lane detection on structured roads is the presence of lane markings that are not always clearly visible due to their print quality (Yenikaya *et al.*, 2013) and changes in environmental conditions (McCall and Trivedi, 2004). Most of the algorithms proposed are based on edge detection (Phueakjeen *et al.*, 2011; Truong and Lee, 2008) and color segmentation (Sun *et al.*, 2006; Chiu and Lin, 2005). However, these approaches are limited to the information obtained from the current image frame. Therefore, approaches that allow lane tracking emerge, determining the location of lane boundaries in a sequence of consecutive images, using information on the location of the lane in previous images to restrict the probable location of the lane in the current image (Kluge and Lakshmanan, 1995). These approaches are based on Bayesian filters (Nieto *et al.*, 2012) which provide a framework for inferring the probability of a hypothesis

[*]Corresponding author

being true from observations.

In recent years, deep learning has significantly advanced lane detection by leveraging convolutional neural networks (CNNs) and other deep learning architectures. Methods such as segmentation-based methods (e.g., GCN (Chiang *et al.*, 2019), SCNN (Parashar *et al.*, 2017)), GAN-based methods (e.g., EL-GAN (Gao *et al.*, 2022)) or lane detection systems using deep learning frameworks such as YOLO (You Only Look Once) and its variants have shown promising results in achieving high accuracy in lane detection (Tang *et al.*, 2021). However, deep learning-based approaches also have their limitations. One of the main limitations is the computational complexity and the considerable amount of processing computing power required. This high computational demand can result in significant latency, making it difficult to implement these systems in real-time applications, especially in environments with limited hardware capabilities (Zakaria *et al.*, 2023). In addition, the need for large labeled datasets to train these deep learning models add another layer of complexity, as the collection and annotation of such data is an exhaustive process.

In both lane detection and lane tracking, there are basically two classes for lane representation: the feature-based and the model-based representations. The first parameterizes lanes by combining low-level features such as painted lines, edges, and lane segments. Consequently, this technique depends on the number of features extracted from the lane and has the disadvantage of not imposing restrictions on shape or length (Wang *et al.*, 2004). On the other hand, model-based techniques only use some parameters to represent the lanes, assuming that lane shapes can be parameterized using mathematical models, such as straight lines, parabolic curves, spirals or splines. Model-based approaches are more robust against noise and the lack of data than feature-based techniques (Barshan *et al.*, 2011).

In order to tackle the above problems, we present a lane detector, based on the Kalman filter so as to incorporate information from previous frames. To achieve this task, a principal component analysis (PCA) (Tharrault *et al.*, 2008) generated from parameterized routes of segmented lanes is developed, obtaining example-based lane models, which are ideal for dimensionality reduction. As a consequence, instead of having the Kalman filter defined at the pixel space (considering one hundred pixel positions), we use a reduced PCA projected space of three parameters. This lower dimensional representation can be incorporated into covariance self-tuning strategies.

**1.1. Related work.** Lane detection is an important challenge to make the future of autonomous vehicles possible. In recent years, several lane detection methods based on image processing and filtering techniques have been proposed as reported in the literature (Borkar *et al.*, 2009; Li *et al.*, 2018). Such works have been focused on solving the problem using classical computer vision methods at the image level, i.e., color segmentation and gradient operators (Truong and Lee, 2008). For example, Chiu and Lin (2005) used the RGB space to find the lane marking in contrast to the color of the road through a threshold, which was defined in accordance with experiments performed at different times of the day. However, this method showed problems in the presence of shadows cast on the lane. Other approaches have explored the effect of using gradient operators over grayscale images based on Canny's algorithm, which performs edge detection with reduced processing time (Assidiq *et al.*, 2008). In addition, this work includes techniques for diminishing projected shadows, previously developed by Finlayson *et al.* (2002). Unfortunately, the above methods do not consider previous information; therefore they do not benefit from the continuity naturally exhibited by lane paths (Zou *et al.*, 2020; Rhouma *et al.*, 2022).

An alternative way to tackle the single frame problem in lane detection is through probabilistic approaches which were proposed to improve the robustness and stability of lane tracking (Liu *et al.*, 2010). For instance, Danescu and Nedevschi (2009) designed a cascade particle filter to work with a nonlinear system (Liu *et al.*, 2022) and reducing the computational cost of the conventional particle filter. However, the main problem of this filter is that the particles are degraded as the importance weight of each particle decreases. Another filter commonly used in lane tracking is the Kalman filter (Rhouma *et al.*, 2022), which allows modeling and reducing the variation of lane estimation through a correction stage, obtaining temporal coherence (Meuter *et al.*, 2009). Mammeri *et al.* (2014) used this filter to track the endpoints of a line obtained using the Hough transform, which was restricted to parameterize straight lines. This parameterization was required by the Kalman filter, as it is an optimal estimator for linear systems. In practical applications of this filter, the input parameters are generally unknown, and adjusting them is a difficult task (Åkesson *et al.*, 2008). If performed manually, this would represent a considerable burden on the user. Several works have developed self-adjustment strategies (Odelson *et al.*, 2006; Sumithra and Vadivel, 2021; Macias and Gomez, 2006), where it was shown that it is possible to control the behavior of the filter in the presence of outliers.

We aim at obtaining both a prediction and a linear model that are suitable for being used into a Kalman filter framework, which provides an efficient recursive solution to the least squares method. The Kalman filter is a linear, unbiased and optimal estimator of the state of a process at each moment of time. Based on information available at time $t-1$, an updating step is carried out through additional information available at time $t$. Considering the linear nature of the filter, we perform a PCA based on parameterized segmented lanes from a training obtained in a competition of small-sized vehicles (see Fig. 1). It is important to note that not all lanes are segmentable (see Fig. 2), mainly due to illumination problems.

**1.2. Deterministic lane detection.** The measurement model is based on the extraction of characteristics by implementing the Canny algorithm (Green, 2002) and subsequently parametrizing the curves through first-order polynomials for straight shapes, and third-order polynomials for curved shapes. With this, it is possible to gather routes for a database and then represent such routes considering a set of equally spaced points conforming them. To distinguish the left from the right lanes, a hierarchical clustering technique is used (Johnson, 1967). The measurement model process, used to construct the lane database, is depicted in Fig. 3.

**1.3. Dimensionality reduction.** One of the main objectives of this work is to build a linear model of segmented lanes, which allows for projection and reprojection of shapes from new examples. The principal components obtained from a dataset of segmented lanes with the reference frame relative to the vehicle frame must contain a high percentage of the variability observed along the examples. Our data were gathered during three autonomous vehicle competitions organized by the Mexican Robotics Federation, which were held in Mexico City (CDMX), Monterrey (MTY) and Guadalajara (GDL), from 2017 to 2019. Samples from each database were segmented by the deterministic lane detector defined in Section 1.2, and are shown in Fig. 4.

In our experiments, each lane is represented by a discrete finite sequence of $m = 102$ points $(x, y)$ as $\mathbf{t}_{n\times 1} = [x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_m]^T$, where $n = 2m$ is the number of elements of each vector $\mathbf{t}$. The training set data matrix, $[\mathbf{t}_1 \mid \mathbf{t}_2 \mid \cdots \mid \mathbf{t}_k]$, where $k$ is the number of training examples is then formed by using the long vectors $\mathbf{t}$ as columns. The differences from the average route $\bar{\mathbf{t}}$ are used to construct the centered training data matrix $\mathbf{T}_{n\times k} = [(\mathbf{t}_1 - \bar{\mathbf{t}}) \mid (\mathbf{t}_2 - \bar{\mathbf{t}}) \mid \cdots \mid (\mathbf{t}_k - \bar{\mathbf{t}})]$. Principal component analysis seeks a set of $n - 1$ orthogonal vectors which, in the least-squares sense, minimize the correlation between the columns of $\mathbf{T}$. The solution is found by calculating the eigenvectors of the covariance matrix $\Sigma_{n\times n} = \mathbf{T}\mathbf{T}^T$. As $\Sigma$ is symmetric, there always exist an orthogonal basis $U_{n\times n}$ and a diagonal matrix $\Lambda_{n\times n}$ that satisfies the relationship

$$\Sigma = U\Lambda U^T, \tag{1}$$

where $U_{n\times n}$ is the eigenvector matrix and the eigenvalues of $\Sigma$ are the diagonal elements of matrix $\Lambda$. Typically, a number of eigenvectors $l$ is chosen so that sufficient in-training variability is conserved. From now on, we will refer to matrix $\widehat{U}_{n\times l}$ as the model.

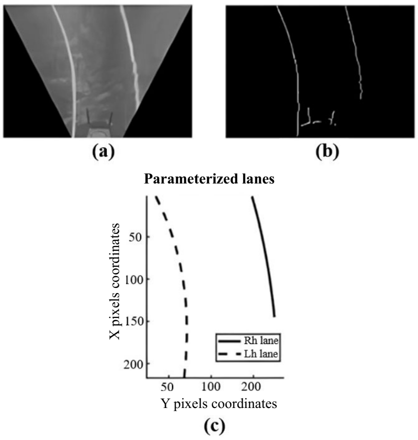


Fig. 1. Segmentable lane case: image frame after homography (a), image frame applying a Canny edge detector (b), segmentable lanes (c).

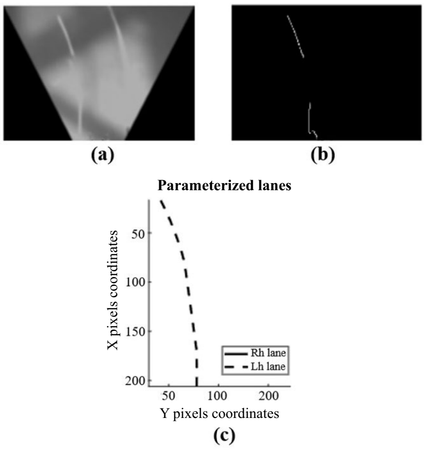


Fig. 2. Unsegmentable lane case: image frame after homography (a), image frame applying a Canny edge detector (b), lane parametrization in image frame with light exposure (c).

An out-of-training example $t_0$ can be fitted to the model by calculating the parameter vector $b = [b_1, b_2, \cdots, b_l]^T$ that minimizes the squared error $e$, expressed as

$$b = \widehat{U}^T(t_0 - \bar{t}). \tag{2}$$

The recovered approximation of the out-of-training lane $t_0$ is given by

$$t_0 \approx \bar{t} + \widehat{U}b. \tag{3}$$

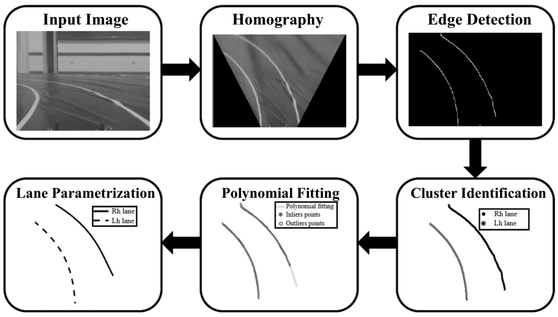
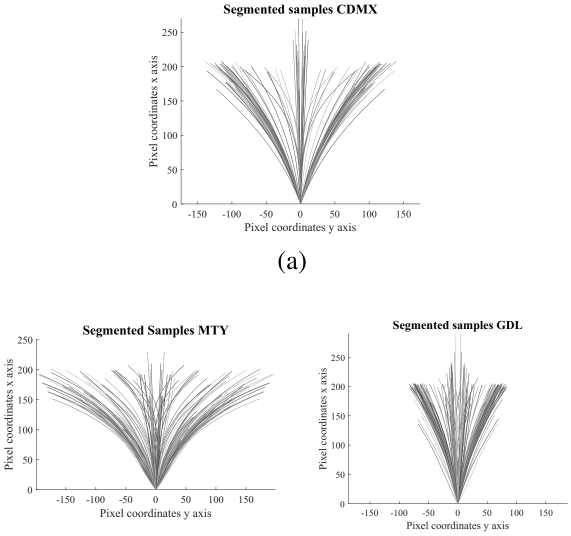


Fig. 3. Deterministic lane detection model. The different stages for accessing the measurement model are shown in the figure.



(a)

(b)

(c)

Fig. 4. Pixel-level databases of segmented samples: CDMX database, 82 curve and 38 straight lanes were segmented (a), MTY database, 76 curve and 40 straight lanes were segmented (b), GDL database, 154 curve and 48 straight lanes were segmented (c).

## 2. Low-dimensional parametrization

According to Fig. 5, it was observed that three eigenvectors were sufficient to capture at least $98\%$ of the model's variability, for which the matrix $\widehat{U}$ was set to dimensions $204 \times 3$. As a consequence, using the model it is possible to fit an out-of-training trajectory using only three eigenvectors.

Figure 6 depicts the behavior of training–testing pairs between the different datasets used in our experiments. The box plots illustrate the interquartile ranges and highlight the median values, indicating that MTY–GDL not only has a lower standard deviation but also exhibits less variability and fewer outliers. Box plot whiskers were determined by calculating $Q_1 - 1.5 \times$ IQR and $Q_3 + 1.5 \times$ IQR, where $Q_1$ represents the first quartile, $Q_3$ is the third quartile, and IQR represents the interquartile range (i.e., IQR $= Q_3 - Q_1$). Data points outside these whiskers were considered outliers. This observation reveals that the MTY data represent the most robust model when entering out-of-training lanes, as evidenced by its smaller dispersion error. Specifically, the standard deviation for MTY–GDL is 0.24, compared with 0.81 for GDL–CDMX and 0.63 for CDMX–GDL. Recall that, in order to achieve dimensionality reduction, first an image sized $480 \times 640$ undergoes lane segmentation; then, a $204 \times 1$ vector is formed through the acquired $(x, y)$ coordinates of the segmented lanes; and finally, the model transforms the segmented shape vector into a highly reduced $3 \times 1$ vector.

In order to know the contribution of each eigenvector, these were reprojected by Eqn. (3) varying the coefficient of $b$ within the range from $-12$ to 12, which represents 66 percent of the $b$ data distribution. The shape and length contributions of the resulting reprojected lanes are shown in Figure 7.

Figure 7(a) shows that the contribution of the first eigenvector corresponds to left-to-right variations of the lanes, which exhibit a rather constant curvature. Thus, this eigenvector will have a greater participation in the reprojection of lanes that belong to curves in the road. The contribution of the second eigenvector corresponds to an elongation with respect to the $y$ coordinate axis, as shown in Fig. 7(b). Alternatively, Fig. 7(c) presents a shape contribution related to an inflection point, characterizing another kind of curvature observed in the lanes. The prior knowledge of the contribution of eigenvectors allows designing a strategy for covariance self-tuning within the proposed Kalman filter described in the next section.

## 3. Probabilistic segmentation based on the Kalman filter

The Kalman filter for probabilistic segmentation is formulated from the dimensionality reduction analysis presented in Section 1.3, which shows a model generated by PCA that allows for the reprojection of segmented routes in terms of three parameters. Since this model is linear, the Kalman filter provides an efficient recursive solution of the least squares method which consists of two main stages: prediction and update, the state representation is shown in

$$x_t = Ax_{t-1} + Bu_t + \epsilon_t, \tag{4}$$

$$z_t = Cx_t + \delta_t, \tag{5}$$

respectively, where $A$ is the $n \times n$ state matrix, $n$ is the dimension of the state vector $x_t$, $B$ is the $n \times m$ control matrix, and $m$ is the dimension of the control vector $u_t$. The random variable $\epsilon_t$ is a Gaussian random vector of the same dimension as the state vector, with zero mean and covariance matrix $R_t$, $C_t$ is a $k \times n$ matrix, where $k$ shares the same dimension as the measurement vector $z_t$. The vector $\delta_t$ describes the measurement noise and its distribution is multivariate Gaussian with zero mean and covariance matrix $Q_t$.

**3.1. Measurement model.** According to the measurement model equation of the Kalman filter in the state-space form, Eqn. (5), and Eqn. (2), corresponding to the reprojection of the segmented route, the measurement model of the probabilistic segmentation method is defined by

$$z_t = \widehat{U}^T x_{t,d} + \delta_t, \tag{6}$$

where $C = \widehat{U}$ is the PCA model of dimension $204 \times 3$ and $x_{t,d}$ is the segmented lane given by the deterministic lane detector with dimension $204 \times 1$.

**3.2. Prediction model.** At the image level, there is no control over the behavior of lanes. For this reason, the prediction will depend directly on the added Gaussian noise, with $u_t = 0$ and $A_t = \widehat{U}^T$ for the PCA model, transforming Eqn. (4) (corresponding to the prediction) into the state space form as

$$x_t = \widehat{U}^T x_{t-1} + \epsilon_t, \tag{7}$$

where $x_{t-1}$ is the prior state corresponding to a lane of dimension $204 \times 1$, reprojected by the matrix of the PCA model $\widehat{U}$ $204 \times 3$, and $\epsilon_t$ is the Gaussian noise with zero mean and covariance matrix $R_t$ of size $3 \times 3$.

**3.3. Formulation of the proposed Kalman filter algorithm.** It is important to note that not all lanes are segmentable due to lighting conditions or occlusions. For this reason the filter proposed in this work performs a strategy to correct such situations. This strategy was inspired by the work of Macias and Gomez (2006), in which a method was presented to estimate the noise covariance from the process data, performing an update of the covariance during the transitory periods and thus allowing rapid convergence of the Kalman filter in the presence of sudden changes in the observation. The sudden change in lane segmentation arises from not detecting a route in the observation model. Based on the above, our strategy maintains constant covariance matrices while the measurement model detects a route and updates the covariances by not detecting a route. Before showing the proposed algorithm, Section 3.3.1 details the main idea about the covariance update strategy.

**3.3.1. Updating strategy for covariance matrices.** According to the analysis carried out in Section 2, it was observed that the first eigenvector corresponds to a

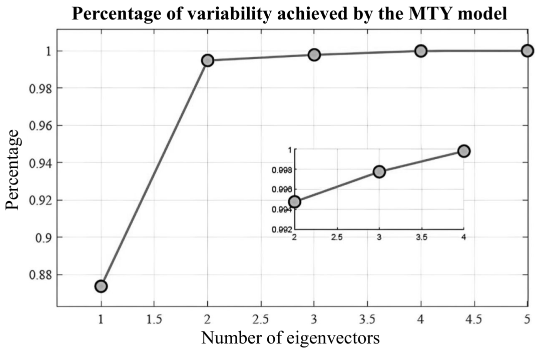


Fig. 5. Percentage of variability achieved by the model. The figure shows that with only three eigenvectors it is possible to achieve more than 98% variability.

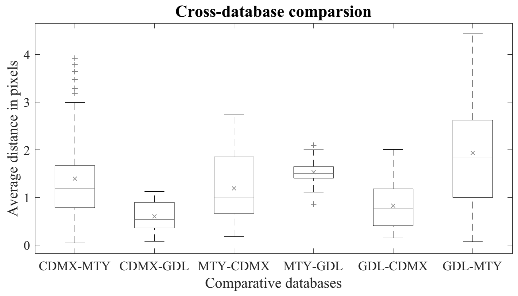
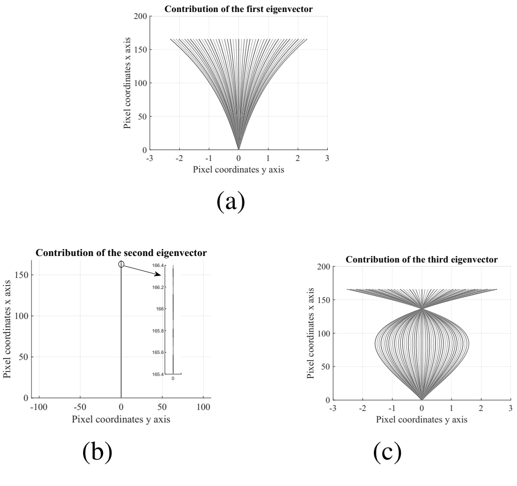


Fig. 6. Selecting the best database for training. The box plots show the average distance in pixels between reconstructed examples and ground truth. Each box plot relates to a training–testing pair, i.e., the pair CDMX–MTY indicates that the model was trained using the lanes segmented from Mexico City's competition and tested using the lanes segmented from Monterrey's competition.



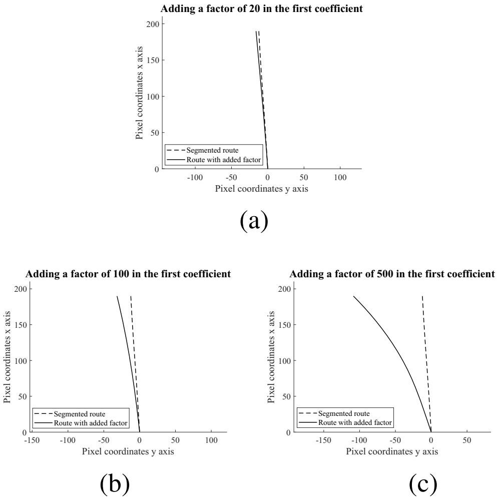
(a)

(b) (c)

Fig. 7. Eigenvector contribution from the MTY model: synthetic variations along the first eigenvector show contributions related with left-right curvature changes in the segmented lanes (a), the second most important variability, related to the length of the segmented lanes (b), the third eigenvector depicting a greater elongation along curvatures (c).

contribution with respect to left-to-right variations of the lanes, as well as in the curvature of the projected route. This contribution is significant from a factor in the order of hundreds as shown in Fig. 8, where it is possible to observe that by varying positive values, the route begins to acquire a greater curvature, as well as a displacement from the endpoint to the left; for negative values the behavior is mirroring. Figures 8(b) and (c) show a significant change in curvature when a factor of an order of hundreds is added, presenting an average distance difference of $8.11$ and $40.56$ pixels, respectively, while in Fig. 8(a) this distance is $1.62$ pixels with a factor of 20. With this prior knowledge it is possible to anticipate the prediction behavior when a route $\mathbf{x}_{t,d}$ is not detected, identifying whether the last state corresponds to a line or a curve.

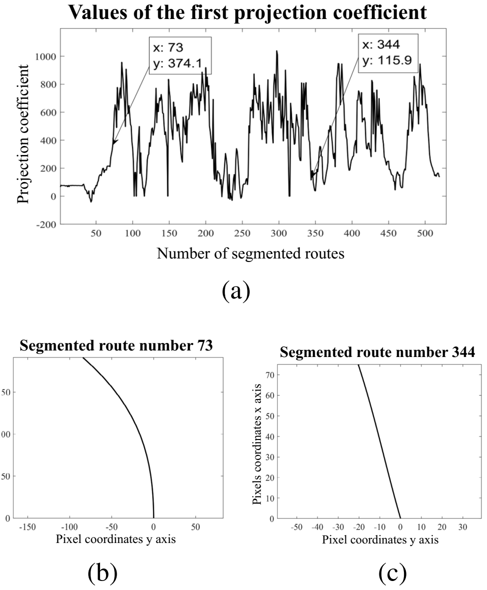
To obtain a threshold of the first coefficient that determines whether the state corresponds to a straight line or a curve, $520$ routes were segmented from the MTY database, from which the projection coefficients were obtained (see Fig. 9(a)). It was observed that for values greater than $250$ of the first coefficient the route corresponds to a curve (see Fig. 9(b)) and for values less than $250$ the route corresponds to a straight line (see Fig. 9(c)). With the threshold defining the state of the route and the knowledge about the behavior of the route to the variation of the coefficients, the strategy of updating the covariance matrices in the Kalman filter is developed.



(a)

(b) (c)

Fig. 8. Positive variation in the first coefficient: adding a factor of 20 (a), adding a factor of 100 (b), adding a factor of 500 (c).



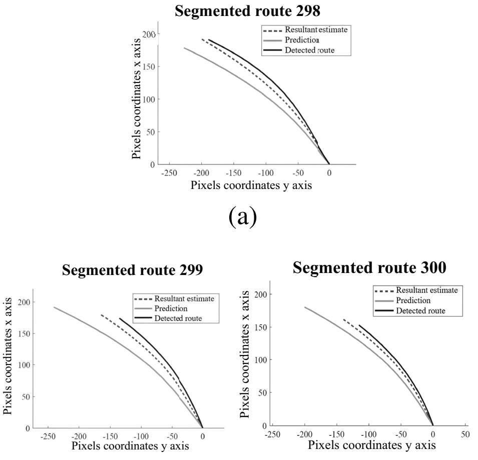
(a)

(b) (c)

Fig. 9. Definition of the threshold to determine whether the state of the segmented route corresponds to a curve or a straight line: first coefficient with respect to the segmented route (a), Route 73 corresponds to a curve (b), Route 344 corresponds to a straight line (c).

**3.3.2. Proposed segmentation algorithm based on the Kalman filter.** Algorithm 1 shows, based on the previous section, the strategy for updating the covariances in the absence of a detected route $x_{t,d}$. In this covariance update algorithm, the function $\mathrm{diag}(\mathbf{vec})$ returns a diagonal matrix whose main diagonal elements are the vector $\mathbf{vec}$. The algorithm indicates in Line 3 that if a detected route $x_{t,d}$ is not obtained, the uncertainty in the measurement model increases by a factor of $12$. Conversely, in the prediction model, a factor of $0.5$ reduces the uncertainty. These factors increase linearly for each instant when the route $x_{t,d}$ is absent, as the counter $c$ increments by $\Delta c$. It is important to note that positive scaling factors are used for this covariance matrix update, which ensures that the resulting matrices $Q_t$ and $R_t$ are always symmetric and positive definite. This ensures that the resulting estimate will maintain a normal distribution. In Line 7 the difference between the segmented route and the previous estimate in the reduced dimension space is calculated, which allows from Lines 8 to 16 covariance matrices to be defined according to the value of the difference of the coefficients. In case the last detected route is a curve, a factor is added to obtain a variation in the first coefficient of the estimate. This determines the transition of estimates in curve sections where it is not possible to segment a lane. Given Eqns. (6) and (7) corresponding to the measurement and prediction model, respectively, Algorithm 2 based on the Kalman filter designed in the reduced space of three dimensions is proposed.

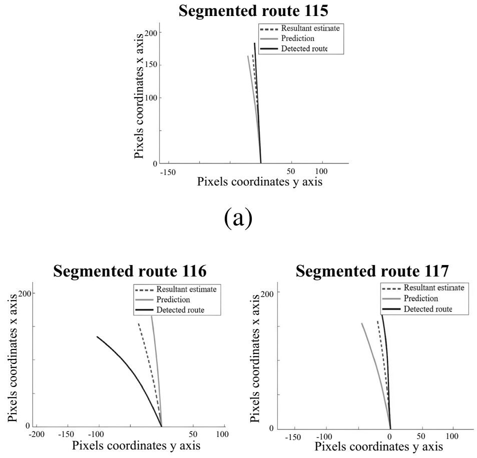
In Line 3, the filter is started by considering a hypothesis of the previous state $x_{t-1}$. In this process, the covariance matrices are held constant upon successful segmentation of $x_{t,d}$. However, when lane detection is not available, the covariances are updated according to Algorithm 1. Line 9 calculates the Kalman gain $K_t$, which regulates the contribution of the difference between the



(a)

(b) (c)

Fig. 10. Probabilistic segmentation method estimation: segmented Route 298 (a), segmented Route 299 (b), segmented Route 300 (c).



(a)

(b) (c)

Fig. 11. Regulation of the estimation in the presence of an atypical observation: segmented Route 115 (a), segmented Route 116 (b), segmented Route 117 (c).

detected route $x_{t,d}$ and the previous estimate $\mathbf{x}_t$, defined in Line 10. Finally, Line 11 updates the uncertainty matrix $\Sigma_t$. The results obtained by the proposed probabilistic segmentation method are presented in the next section.

## 4. Experimental results

In this section we validate the formulation of the proposed probabilistic segmentation method, testing its performance in practice by means of two experiments: the first one corresponds to the segmentation capacity when detecting the route $\mathbf{x}_{t,d}$; the second shows the predictive capability of the method when there is no information in the visual sensor.

**Algorithm 1.** Cov_update($\bar{x}_t, x_{t,d}, R_t, Q_t$).

```
1:  Function
2:  $c = 1; \Delta_c = 0.2; C_s = 100;$
3:  while $x_{t,d} == \{\}$ do
4:      $F\_Q_t = c\text{diag}([12, 12]), F\_R_t = c\text{diag}([0.5, 0.5])$

5:      if $x_{t,d}[1] > 250$ then
6:          $M\_R_t = \text{diag}([mrt, 1, 1])$
7:      end if
8:      $\Delta_x = x_{t,d} - \bar{x}_t$
9:      if $\Delta[1] > C_s$ then
10:         $R_t = F\_R_t\text{diag}([R_t[1,1], 0.1, 0.1])M\_R_t$
11:         $Q_t = F\_Q_t\text{diag}([Q_t[1,1], 0.1, 0.1])$
12:     end if
13:     if $\Delta[2] > C_s$ then
14:         $R_t = F\_R_t\text{diag}([0.1, R_t[2,2], 0.1])M\_R_t$
15:         $Q_t = F\_Q_t\text{diag}([0.1, Q_t[2,2], 0.1])$
16:     end if
17:     if $\Delta[1] > \Delta[2] > C_s$ then
18:         $R_t = F\_R_t\text{diag}([R_t[1,1], R_t[2,2], 0.1])M\_R_t$
19:         $Q_t = F\_Q_t\text{diag}([Q_t[1,1], Q_t[2,2], 0.1])$
20:     end if
21:     $c = c + \Delta_c$
22: end while
23: return $Q_t, R_t$
```

### 4.1. Experiment 1: Segmentation with visual sensor information.

For this experiment, 520 estimations were performed using the MTY data at 15 frames per second. According to the calculation of the covariance matrices, $R_t$ presents a higher uncertainty, having a $36.72\%$ higher variance in the first value of the diagonal, $36.72\%$ in the second value and $13.01\%$ higher in the third value with respect to the covariance matrix $Q_t$. These matrices determine uncertainty, indicating the degree of reliability of the model. Therefore, by having a detected route $x_{t,d}$ the measurement model has a lower uncertainty, and the resulting estimate $x_t$ will have a higher similarity with respect to the detected route compared to the prediction. The above is shown in Fig. 10, in which three consecutive estimates are presented. Comparing numerically, the prediction in Fig. 10(a) presents a difference in the average distance with respect to the detected route of $18.74$ pixels, as opposed to the resulting estimate which has a difference of $7.14$ pixels with respect to the detected route. Since the data sets were obtained at 15 frames per second, it is expected that the change in the segmented routes and the result of the probabilistic segmentation method shows a smooth transition, even in the presence of an outlier observation. This is due to the proposed covariance update strategy that regulates such transition that is shown in Fig. 11.

Note that in Fig. 11(b) the resulting estimate has a

**Algorithm 2.** Proposed Kalman filter—segmentation $(x_{t-}, \bar{\Sigma}_t, x_{t,d}, Q_t, R_t)$.

```
Function
if $t = 1$ then
    $\bar{x}_t = \widehat{U}^T(x_{t-1} - \bar{t})$
else
    $\bar{x}_t = x_t$
end if
if $x_{t,d} == \{\}$ then
    $[Q_t, R_t] = Cov\_update(\bar{x}_t, x_{t,d}, R_t, Q_t)$
end if
$\bar{\Sigma}_t = \bar{\Sigma}_t + R_t$
$K_t = \bar{\Sigma}_t(\bar{\Sigma}_t + Q_t)^{-1}$
$x_t = \bar{x}_t + K_t(x_{t,d} - \bar{x}_t)$
$\Sigma_t = (I - K_t)\bar{\Sigma}_t$
return $x_t, \Sigma_t$
```

higher reliability with respect to the prediction, because the route detected in this frame has a large variation with respect to the previous information. Analyzing the accuracy of the route prediction with the detected route revealed an average variation of 11.2 pixels, with a standard deviation of 2.4 pixels.

### 4.2. Experiment 2: Segmentation without information in the visual sensor.

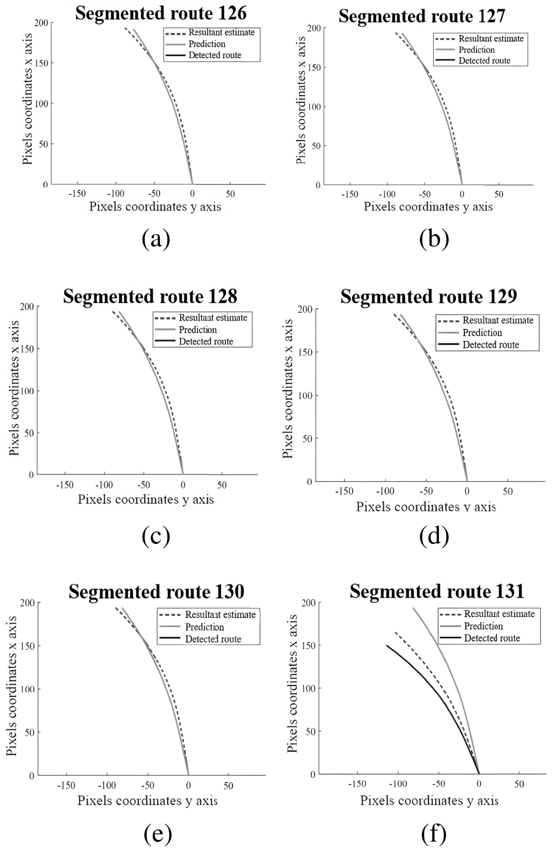
The purpose of this experiment is to demonstrate the predictive capacity of the probabilistic segmentation method when no information is available in the visual sensor. For this purpose, in this experiment, tests were performed in sections with different occlusion periods. Figure 12 shows the variation in the estimation of routes without observation, in which there is a significant difference in the reduced dimension space of 43.1 pixels with respect to the prediction, demonstrating the predictive ability of the proposed method, once a detection is obtained (cf. Fig. 12(f)), the regulation of the proposed model is observed.

With these experiments, it has been shown that the proposed method based on the Kalman filter results in a smooth transition in the estimations. This behavior is favorable since there is no atypical change in the lanes. On the other hand, at the deterministic level, the observation model presents significant changes from one lane to another.

### 4.3. Performance of the proposed model estimates.

Figure 13 shows this behavior for 520 observations in which straight and curved sections are included.

Note that there is a large variation between observations, since a parameterization of the lanes is performed with the information obtained at time $t$, which may not be reliable mainly due to illumination changes. Figure 14 illustrates the variation using a fully connected pre-trained CNN model for lane detection (Zou *et al.*, 2020). While there is an improvement in smoothness compared with the inter-lane variations of the measurement model, the behavior is not entirely smooth.



(a) (b)

(c) (d)

(e) (f)

Fig. 12. Curves section, estimates in image frames 126 to 130 are made without observation (solid black line): (a), (b), (c), (d), (e) display estimates without observation, (f) shows the effect of the incorporation of an observation.

Detection issues still persist, resulting in significant variations. Additionally, the computation time to process a frame using deep learning (DL) was 4.1 seconds on hardware with an RTX 2070 GPU and 32GB of RAM, compared with the proposed method which processes in real-time at 0.04 seconds per frame. The process time analysis was also performed for the Guadalajara (GDL) and Mexico City (CDMX) databases, obtaining similar results.

The Kalman filter considers the previous information given by the proposed measurement model and regulates the contribution of the new observation, resulting in a smooth behavior of the transitions as shown in Fig. 15.

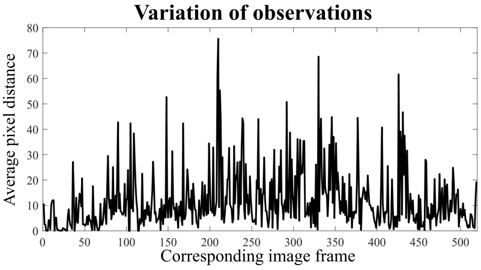


Fig. 13. Average difference in pixels between deterministic observations. Variation corresponds to a significant change in the lane.

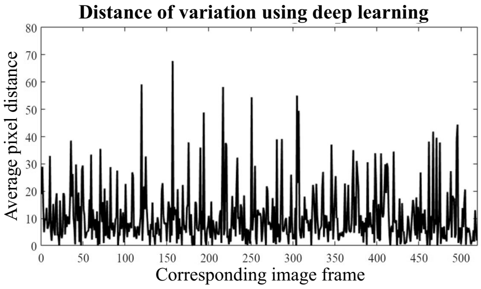


Fig. 14. Average pixel difference between lanes detected using deep learning. The variation obtained in lane detection through DL remains significant.

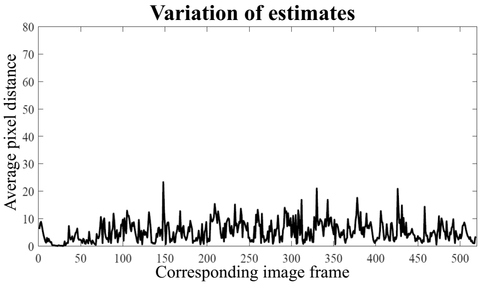


Fig. 15. Average difference in pixels between estimates after applying the Kalman filter. A smooth transition is observed between estimates.

## 5. Conclusions

In this paper, we proposed a probabilistic segmentation method based on the Kalman filter, which was designed in a reduced dimensional space using principal component analysis to validate a linear model that allows the reprojection of routes in this space. The importance of this model lies in the fact that it comes from route data collected in three different autonomous vehicle competitions, which is of relevance in the study of principal variations of routes for scale vehicle competitions. We have performed an experimental evaluation that demonstrates the performance efficiency of the method for shape and lane dimension estimations.

In the experiments, the regulation of the method to achieve a smooth transition between estimates becomes visible, including the input of an outlier observation made by the deterministic lane detector. In addition, the method shows predictive capabilities when facing image frames with occlusions due to the strategy developed for the covariance update. As future work, we plan to incorporate additional information from other sensors into the measurement model, use more sophisticated techniques for curvature parameterization, and include a deep learning-based measurement model.

## References

Åkesson, B.M., Jørgensen, J.B., Poulsen, N.K. and Jørgensen, S.B. (2008). A generalized autocovariance least-squares method for Kalman filter tuning, *Journal of Process Control* **18**(7–8): 769–779, DOI: 10.1016/j.jprocont.2007.11.003.

Assidiq, A., Khalifa, O.O., Islam, M.R. and Khan, S. (2008). Real time lane detection for autonomous vehicles, *2008 International Conference on Computer and Communication Engineering, Kuala Lumpur, Malaysia*, pp. 82–88, DOI: 10.1109/iccce.2008.4580573.

Barshan, E., Ghodsi, A., Azimifar, Z. and Jahromi, M.Z. (2011). Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds, *Pattern Recognition* **44**(7): 1357–1371, DOI: 10.1016/j.patcog.2010.12.015.

Borkar, A., Hayes, M. and Smith, M.T. (2009). Robust lane detection and tracking with RANSAC and Kalman filter, *2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt*, pp. 3261–3264, DOI: 10.1109/icip.2009.5413980.

Chiang, W.-L., Liu, X., Si, S., Li, Y., Bengio, S. and Hsieh, C.-J. (2019). Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks, *25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD'19, Anchorage, USA*, pp. 257–266, DOI: 10.1145/3292500.3330925.

Chiu, K.-Y. and Lin, S.-F. (2005). Lane detection using color-based segmentation, *Intelligent Vehicles Symposium, Las Vegas, USA*, pp. 706–711, DOI: 10.1109/ivs.2005.1505186.

Danescu, R. and Nedevschi, S. (2009). Probabilistic lane tracking in difficult road scenarios using stereovision, *IEEE Transactions on Intelligent Transportation Systems* **10**(2): 272–282, DOI: 10.1109/tits.2009.2018328.

Finlayson, G.D., Hordley, S.D. and Drew, M.S. (2002). Removing shadows from images, *in* A. Heyden *et al.* (Eds), *Computer Vision, ECCV 2002*, Springer, Berlin/Heidelberg, pp. 823–836, DOI: 10.1007/3-540-47979-1_55.

Gao, L., Wu, L. and Meng, X. (2022). EL-GAN: Edge-enhanced generative adversarial network for layout-to-image generation, *Computer Graphics Forum* **41**(7): 407–418, DOI: 10.1111/cgf.14687.

Green, B. (2002). Canny edge detection tutorial, `https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html`.

Johnson, S.C. (1967). Hierarchical clustering schemes, *Psychometrika* **32**(3): 241–254.

Kluge, K. and Lakshmanan, S. (1995). A deformable-template approach to lane detection, *Intelligent Vehicles '95 Symposium, Detroit, USA*, pp. 54–59, DOI: 10.1109/ivs.1995.528257.

Kreucher, C., Lakshmanan, S. and Kluge, K. (1998). A driver warning system based on the LOIS lane detection algorithm, *IEEE International Conference on Intelligent Vehicles, Stuttgart, Germany*, Vol. 1, pp. 17–22.

Li, M., Li, Y. and Jiang, M. (2018). Lane detection based on connection of various feature extraction methods, *Advances in Multimedia* **2018**(1): 8320207, DOI: 10.1155/2018/8320207.

Liu, G., Wörgötter, F. and Markelić, I. (2010). Combining statistical Hough transform and particle filter for robust lane detection and tracking, *2010 IEEE Intelligent Vehicles Symposium, La Jolla, USA,* pp. 993–997, DOI: 10.1109/ivs.2010.5548021.

Liu, G., Wu, S., Zhu, L., Wang, J. and Lv, Q. (2022). Fast and smooth trajectory planning for a class of linear systems based on parameter and constraint reduction, *International Journal of Applied Mathematics and Computer Science* **32**(1): 11–21, DOI: 10.34768/amcs-2022-0002.

Macias, J. and Gomez, A. (2006). Self-tuning of Kalman filters for harmonic computation, *IEEE Transactions on Power Delivery* **21**(1): 501–503, DOI: 10.1109/tpwrd.2005.860411.

Mammeri, A., Boukerche, A. and Lu, G. (2014). Lane detection and tracking system based on the MSER algorithm, Hough transform and Kalman filter, *17th ACM international conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM'14, Montreal, Canada*, pp. 259–266, DOI: 10.1145/2641798.2641807.

McCall, J. and Trivedi, M. (2004). An integrated, robust approach to lane marking detection and lane tracking, *IEEE Intelligent Vehicles Symposium, Parma, Italy*, pp. 533–537, DOI: 10.1109/ivs.2004.1336440.

Meuter, M., Muller-Schneiders, S., Mika, A., Hold, S., Nunn, C. and Kummert, A. (2009). A novel approach to lane detection and tracking, *12th International IEEE Conference on Intelligent Transportation Systems, St. Louis, USA*, pp. 1–6, DOI: 10.1109/itsc.2009.5309855.

Nieto, M., Cortés, A., Otaegui, O., Arróspide, J. and Salgado, L. (2012). Real-time lane tracking using Rao-Blackwellized particle filter, *Journal of Real-Time Image Processing* **11**(1): 179–191, DOI: 10.1007/s11554-012-0315-0.

Odelson, B., Lutz, A. and Rawlings, J. (2006). The autocovariance least-squares method for estimating covariances: Application to model-based control of chemical reactors, *IEEE Transactions on Control Systems Technology* **14**(3): 532–540, DOI: 10.1109/tcst.2005.860519.

Parashar, A., Rhu, M., Mukkara, A., Puglielli, A., Venkatesan, R., Khailany, B., Emer, J., Keckler, S.W. and Dally, W.J. (2017). SCNN: An accelerator for compressed-sparse convolutional neural networks, *ACM SIGARCH Computer Architecture News* **45**(2): 27–40, DOI: 10.1145/3140659.3080254.

Phueakjeen, W., Jindapetch, N., Kuburat, L. and Suvanvorn, N. (2011). A study of the edge detection for road lane, *8th Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI), Khon Kaen, Thailand,* pp. 995–998, DOI: 10.1109/ecticon.2011.5948010.

Rhouma, T., Keller, J.-Y. and Abdelkrim, M.N. (2022). A Kalman filter with intermittent observations and reconstruction of data losses, *International Journal of Applied Mathematics and Computer Science* **32**(2): 241–253, DOI: 10.34768/amcs-2022-0018.

Sumithra, S. and Vadivel, R. (2021). An optimal innovation based adaptive estimation Kalman filter for accurate positioning in a vehicular ad-hoc network, *International Journal of Applied Mathematics and Computer Acience* **31**(1): 45–57, DOI: 10.34768/amcs-2021-0004.

Sun, T.-Y., Tsai, S.-J. and Chan, V. (2006). HSI color model based lane-marking detection, *2006 IEEE Intelligent Transportation Systems Conference, Ontario, Canada*, pp. 1168–1172, DOI: 10.1109/itsc.2006.1707380.

Tang, J., Li, S. and Liu, P. (2021). A review of lane detection methods based on deep learning, *Pattern Recognition* **111**(4): 107623, DOI: 10.1016/j.patcog.2020.107623.

Tharrault, Y., Mourot, G. and Ragot, J. (2008). Fault detection and isolation with robust principal component analysis, *16th Mediterranean Conference on Control and Automation, Ajaccio, France*, pp. 59–64, DOI: doi.org/10.1109/med.2008.4602224.

Thorpe, C., Herbert, M., Kanade, T. and Shafer, S. (1991). Toward autonomous driving: The CMU Navlab. I: Perception, *IEEE Expert* **6**(4): 31–42, DOI: 10.1109/64.85919.

Thrun, S., Burgard, W. and Fox, D.D. (2005). *Probabilistic Robotics*, MIT Press, Cambridge, pp. 16–39.

Truong, Q.-B. and Lee, B.-R. (2008). New lane detection algorithm for autonomous vehicles using computer vision, *2008 International Conference on Control, Automation and Systems, Seoul, Korea (South)*, pp. 1208–1213, DOI: 10.1109/iccas.2008.4694332.

Wang, Y., Teoh, E.K. and Shen, D. (2004). Lane detection and tracking using B-snake, *Image and Vision Computing* **22**(4): 269–280, DOI: 10.1016/j.imavis.2003.10.003.

Yenikaya, S., Yenikaya, G. and Düven, E. (2013). Keeping the vehicle on the road, *ACM Computing Surveys* **46**(1): 1–43, DOI: 10.1145/2522968.2522970.

Zakaria, N.J., Shapiai, M.I., Ghani, R.A., Yassin, M.N.M., Ibrahim, M.Z. and Wahid, N. (2023). Lane detection in autonomous vehicles: A systematic review, *IEEE Access* **11**: 3729–3765, DOI: 10.1109/ACCESS.2023.3234442.

Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L. and Wang, Q. (2020). Robust lane detection from continuous driving scenes using deep neural networks, *IEEE Transactions on Vehicular Technology* **69**(1): 41–54, DOI: 10.1109/tvt.2019.2949603.

**Carlos Acuña Ocampo** earned his PhD in robotics and advanced manufacturing from the Center for Research and Advanced Studies of the National Polytechnic Institute, Saltillo Campus. His research focuses on pattern analysis using computer vision and machine learning algorithms. He is a faculty member in the Computer Science Department at the Monterrey Institute of Technology (Tecnologico de Monterrey), Saltillo Campus.

**Gustavo Arechavaleta Servín** received his PhD degree from Institut National des Sciences Appliquées de Toulouse, France, for his work on the computational principles of human walking via optimal control within the GEPETTO Group, Laboratoire d'Analyse et d'Architecture des Systèmes, CNRS, Toulouse, in 2007. In 2008 he joined the Robotics and Advanced Manufacturing Group at CINVESTAV, Saltillo, Mexico, where he is a researcher. His scientific interests are mainly focused on robot trajectory optimization and vision-based robot navigation.

**Mario Castelán** holds a PhD in computer science from the University of York, UK (2006). He is a researcher of robotics and advanced manufacturing at CINVESTAV, Mexico. His expertise lies in pattern analysis for scientific applications, with the focus on materials science, biotechnology, affective computing, and neuroscience.

Received: 25 January 2024
Revised: 15 July 2024
Re-revised: 14 October 2024
Accepted: 16 October 2024