Robust lane markings detection and road geometry computation

A. López[1)], J. Serrat[1)*], C. Cañero[1)], F. Lumbreras[1)] and T. Graf[2)]

[1)] Computer Vision Center & Computer Science Dept.

Edifici O, Universitat Autnoma de Barcelona,

08193 Cerdanyola, Spain

[2)] Electronic Research, Volkswagen AG, Wolfsburg D–38436, Germany.

[*] Corresponding author: e–mail `joans@cvc.uab.es`

Abstract

Detection of lane markings based on a camera sensor can be a low cost solution to lane departure and curve over speed warning. A number of methods and implementations have been reported in the literature. However, reliable detection is still an issue due to cast shadows, wearied and occluded markings, variable ambient lighting conditions etc. We focus on increasing the reliability of detection in two ways. Firstly, we employ a different image feature other than the commonly used edges: ridges, which we claim is better suited to this problem. Secondly, we have adapted RANSAC, a generic robust estimation method, to fit a parametric model of a pair or lane lines to the image features, based on both ridgeness and ridge orientation. In addition this fitting is performed for the left and right lane lines simultaneously, thus enforcing a consistent result. Four measures of interest with regard several driver assistance applications are directly computed from the fitted parametric model at each frame: vehicle yaw angle and lateral offset with regard the lane medial axis, and lane width and curvature. We have qualitatively assessed our method in video sequences captured on several road types and under very different lighting conditions. Also, we have quantitatively assessed it on synthetic but realistic video sequences for which road geometry and vehicle trajectory ground truth are known.

Keywords: driving assistance system, lane line, ridge, robust fitting

1 Introduction

A present challenge of the automotive industry is to develop low cost advanced driver assistance systems (ADAS) able to increase traffic safety and driving comfort. Since vision is the most used human sense for driving, some ADAS features rely on visual sensors (Bertozzi, Broggi and Fascioli, 2000). Specifically, lane departure warning and lateral control can be addressed by detecting the lane markings on the road by means of a forward–facing camera and computer vision techniques. In this paper we focus on this problem, which is one of the first addressed in the field of ADAS. It is a difficult and not yet completely solved problem due to shadows, large contrast variations, vehicles occluding the marks, wearied markings, vehicle ego–motion etc. Recent reviews of detection methods can be found in (Jung and Kelber, 2005; Bertozzi et al., 2000).

Many of the proposed methods share the following three steps. First, collect cues on where the lane markings can be, typically in the form of image points labeled as lane markings candidates. Second, fit a certain lane model to them, commonly straight lines or some smooth parametric curve. Third, perform some sort of tracking in order to impose temporal continuity, yield a smooth response along time and facilitate real–time (the results in the present frame guide the search in the next one).

Ideally, lane markings are white lines on a dark pavement. Thus, the first step is usually based on image edges, defined as extrema of the gradient magnitude along the gradient direction. The gradient magnitude is an *edgeness* measure and the gradient direction can be used to filter out edge points having an orientation inconsistent with the expected orientation of a lane line. However, the gradient magnitude can be misleading: cast shadows and vehicles may give rise to high gradient values, while wearied marks and poor lighting conditions (e.g. in tunnels) reduce lane markings contrast. Also, the gradient orientation tends to be noisy because of its local nature. Therefore, methods based on edge detection algorithms must devise strategies to cope with these problems (*e.g.* local adaptive and hysteresis thresholding). Otherwise, lane lines model fitting would fail or be much more difficult.

The main contributions of this paper are three. The first one is to employ a different low–level image feature, namely, *ridgeness*, to obtain a more reliable lane marking points detection under poor contrast conditions (section 2). Aside from this practical consideration, conceptually, a ridge describes better than an edge what a lane line is: the medial axis of a thick, brighter elongated structure. Secondly, we have adapted RANSAC, a generic robust estimation method, to fit a parametric model to the candidate lane marking points, using as input data both ridgeness and ridge orientation (section 3). Our model consists in a pair of hyperbolas sharing a common horizontal asymptote, which are constrained to be parallel on the road plane. We claim that a better suited feature (ridges) combined with a robust fitting method contribute to improve lane lines detection reliability. We have intentionally avoided any kind of result post processing, tracking or lane line prediction, for example through a Kalman filtering. Instead, each frame is processed independently of the others. This way we can better design the detection and fitting steps. Our aim has been to build a 'baseline' system to which later we can add filtering and data fusion to improve its performance. Thirdly, we quantitatively assess the method with regard to four geometrically meaningful quantities derived from the segmented lane markings: vehicle yaw angle and lateral offset, lane curvature and width. This is possible on synthetic sequences, for which we know exactly the value for these parameters since they are provided as input to a simulator which generates the sequences (section 4). Qualitative (visual) evaluation is also performed on a number of frames from real sequences exhibiting challenging lighting and occlusion conditions. In addition, video results are also provided in a companion material web page. Section 5 draws the main conclusions and comments future work.

2 Lane Markings as Ridges

Ridges of a grey–level image are the center lines of elongated, bright structures. In the case of a lane line is its longitudinal center. This terminology comes from considering an image as a landscape, being the intensity the z axis or height, since then these center lines correspond to the landscape's ridges (figure 1). Accordingly, ridgeness stands for a measure of how much a pixel neighborhood resembles a ridge. Therefore, a ridgeness measure must have high values along the center of the line and decrease as the boundary is approached. A binary ridge image, corresponding to the centerline, can be obtained by simple thresholding, provided we have a well–contrasted and homogeneous ridgeness measure.

This notion of ridge or medial axis is a simpler and, as we will see in short, computationally better characterization of lane lines than that provided by edges. Instead of defining (and trying to find out) a lane line as points between two parallel edge segments with opposite gradient direction, a ridge is the center of the line itself, once a certain amount of smoothing has been performed. And this amount is chosen as the scale at which ridges are sought.

There are different mathematical characterizations of ridges. In (López, Lloret, Serrat and Villanueva, 2000) a new one is proposed which compares favorably to others and that we have adapted for the problem at hand. Let $G_\sigma(\mathbf{x})$ be a 2D Gaussian of standard deviation σ and $L(\mathbf{x})$ be the grey–level image, with $\mathbf{x} = (u, v)$ the spatial coordinates (u columns, v rows). Then, ridgeness is calculated as follows ($*$ and $\cdot$ stand for convolution and the Hadamard product, respectively):

1. Compute a smoothed version of the image, namely

$$L_{\sigma_\mathrm{d}}(\mathbf{x}) = G_{\sigma_\mathrm{d}}(\mathbf{x}) * L(\mathbf{x}) \ \ , \tag{1}$$

2. Compute the gradient vector field

$$\mathbf{w}_{\sigma_\mathrm{d}}(\mathbf{x}) = (\partial_u L_{\sigma_\mathrm{d}}(\mathbf{x}), \partial_v L_{\sigma_\mathrm{d}}(\mathbf{x}))^\top \ \ . \tag{2}$$

3. Compute the structure tensor field

$$\mathbf{S}_{\sigma_\mathrm{d},\sigma_\mathrm{i}}(\mathbf{x}) = G_{\sigma_\mathrm{i}}(\mathbf{x}) * \mathbf{s}_{\sigma_\mathrm{d}}(\mathbf{x}) \ \ , \tag{3}$$

   being

$$\mathbf{s}_{\sigma_\mathrm{d}}(\mathbf{x}) = \mathbf{w}_{\sigma_\mathrm{d}}(\mathbf{x}) \cdot \mathbf{w}^\top_{\sigma_\mathrm{d}}(\mathbf{x}) \ \ . \tag{4}$$

4. Obtain the eigenvector corresponding to the highest eigenvalue of $\mathbf{S}_{\sigma_\mathrm{d},\sigma_\mathrm{i}}(\mathbf{x})$,

namely $\mathbf{w}'_{\sigma_\mathrm{d},\sigma_\mathrm{i}}(\mathbf{x})$. It is known that $\mathbf{w}'_{\sigma_\mathrm{d},\sigma_\mathrm{i}}(\mathbf{x})$ yields the *dominant gradient orientation* of the original image at $\mathbf{x}$ and is perpendicular to the *dominant image orientation* at $\mathbf{x}$ (if $\mathbf{x}$ is from a lane marking then the dominant image orientation is along it). Therefore, its is a more robust orientation measure than the image gradient itself, $\mathbf{w}_{\sigma_\mathrm{d}}(\mathbf{x})$.

It is worth to notice that $\mathbf{w}'_{\sigma_\mathrm{d},\sigma_\mathrm{i}}(\mathbf{x})$ defines an orientation field in the image but for the next step we need a vector field. For this reason we project $\mathbf{w}'_{\sigma_\mathrm{d},\sigma_\mathrm{i}}(\mathbf{x})$ into $\mathbf{w}_{\sigma_\mathrm{d}}(\mathbf{x})$ as:

$$p_{\sigma_\mathrm{d},\sigma_\mathrm{i}}(\mathbf{x}) = {\mathbf{w}'_{\sigma_\mathrm{d},\sigma_\mathrm{i}}}^{\top}(\mathbf{x}) \cdot \mathbf{w}_{\sigma_\mathrm{d}}(\mathbf{x}) \ , \tag{5}$$

and define the following vector field:

$$\tilde{\mathbf{w}}_{\sigma_\mathrm{d},\sigma_\mathrm{i}}(\mathbf{x}) = \mathrm{sign}(p_{\sigma_\mathrm{d},\sigma_\mathrm{i}}(\mathbf{x}))\mathbf{w}'_{\sigma_\mathrm{d},\sigma_\mathrm{i}}(\mathbf{x}) \ . \tag{6}$$

5. Finally, the *ridgeness* measure is defined as the positive values of

$$\tilde{\kappa}_{\sigma_\mathrm{d},\sigma_\mathrm{i}}(\mathbf{x}) = -\mathrm{div}(\tilde{\mathbf{w}}_{\sigma_\mathrm{d},\sigma_\mathrm{i}}(\mathbf{x})) \ , \tag{7}$$

where div() denotes divergence of a vector field.

The parameter σ_d is the *differentiation scale*, in opposition to σ_i which is the *integration scale*. The former must be tuned to the size of the target structures, while the later determines the size of the neighborhood we want to use in order to compute the dominant orientation.

Positive values of $\tilde{\kappa}_{\sigma_{\rm d},\sigma_{\rm i}}(\mathbf{x})$ measure the similarity of a neighborhood to a ridge structure. In fact, it has been shown (López et al., 2000) that these values lie in the range $[0, 2.0]$, where 0 means not at all ridge, around 1.0 quite and 2.0 perfect local maximum. Besides, these values are homogeneously distributed along the center lines, thus facilitating thresholding. We only take into account those pixels $\mathbf{x}$ for which $\tilde{\kappa}_{\sigma_{\rm d},\sigma_{\rm i}}(\mathbf{x}) > 0.25$, a value fixed experimentally but with a large margin before the selected pixels change significantly.

Due to perspective, the imaged lane lines width decreases with distance. In order not to miss them when computing $L_{\sigma_{\rm d}}$ in Eq. (1), we want the upper rows to be less smoothed than lower rows, but just along the horizontal direction. This is achieved by an anisotropic Gaussian smoothing, with covariance matrix $\Sigma = \mathrm{diag}(\sigma_{\rm dx}, \sigma_{\rm dy})$ where $\sigma_{\rm dy}$ is constant and $\sigma_{\rm dx}$ increases with the row number. Table 1 contains the actual values for all the detection parameters.

Since the dominant orientation of a lane marking is perpendicular to the dominant gradient orientation, and therefore perpendicular to $\tilde{\mathbf{w}}_{\sigma_{\rm d},\sigma_{\rm i}}(\mathbf{x})$, this vector field allows to discard pixels whose associated orientation is inconsistent with that expected by a lane markings model instantiation. We discard a ridge point $\mathbf{x}_r$ if $\tilde{\mathbf{w}}_{\sigma_{\rm d},\sigma_{\rm i}}(\mathbf{x}_r)$ is in $[3\pi/4, 5\pi/4]$, which happens for lane markings with a large horizontal component. Of course, this could be a problem for curves with a very high curvature. However, we have experimentally checked that this criterion performs well even for curves that can not be driven safely at more than 40 km/h.

Other interesting properties of $\tilde{\kappa}_{\sigma_{\rm d},\sigma_{\rm i}}(\mathbf{x})$ are invariance to image translation and rotation, as one would expect, but also to monotonic grey–level transforms. The later greatly helps in lane detection in presence of shadows and low contrast conditions, opposite to gradient–based measures. However, this means that ridgeness also enhances some bright and elongated irregularities in the pavement. Fortunately, this can be solved up to a large extent by discarding those ridges surrounded by a very low gradient magnitude neighborhood, less than a certain threshold t_{grad}. We want to remark that the threshold value for selecting relevant ridge points has been fixed once and used in all the sequences, even though the cameras, lenses, vehicles and lighting conditions varied. Figure 2 shows the resulting candidate lane line points in some specially difficult situations like worn off paint, shadows, high contrast variation and tire marks.

3 Lane model and fitting

3.1 Lane lines model

A number of geometrical models for the projected lane lines have been proposed, from simple straight lines to quadratic, spline and other polynomial curves, with the aim of performing a good image segmentation. However, few are built on a sound geometrical base like in (Guiducci, 1999). There it is shown that, under the assumptions of flat road and constant curvature, a lane line is projected onto the image plane as an hyperbola. Admittedly, this is not a new model, but what that work reveals are the relationships among model parameters and meaningful and interesting geometrical entities such as lane width, curvature and the vehicle's lateral position, which we want to compute in order to validate our method, aside of their own evident applicability for driver assistance.

Assume the road is on a plane, that is, there is not vertical curvature neither torsion. Furthermore, the curvature is either constant or varies linearly with the arc length s:

$$C = \frac{1}{R} = C_0 + C_1 s \ . \tag{8}$$

This is consistent with a road formed by segments of constant curvature connected by clothoids (Dickmanns and Mysliwetz, 1992). Assume too that changes in the road direction are smooth, being C_1 small enough with regard to s so that approximations $C_1/C_0 \ll 1$ and $C_1 \ll 1$ hold.

World and camera coordinate systems share a common origin but have different orientation (figure 3). For the world coordinate system the Z axis is parallel to the road tangent, Y axis points downwards and is orthogonal to the road plane, whereas the X axis is parallel to the road plane and orthogonal to the road tangent and therefore to the lane lines. The camera coordinate system has Y axis coincident with the vehicle's direction and sustains an angle $\theta \ll 1$ radians with the road tangent line (also referred as yaw angle). It also forms an angle φ with the road plane (pitch angle). The lane has width L and the camera is located at a horizontal distance of d_r meters from the right border and at height H above the ground. Of course, L, d_r, θ and φ may vary over time, but H is supposed constant. Finally, let E_u and E_v be the focal lengths in pixels/meter along the horizontal and vertical camera axes, and the image origin centered in the principal point (intersection of the optical axis with the image plane). Then, the following equation relates (u_r, v_r), the pixel coordinates where the right lane line is imaged, to the road parameters it belongs to (Guiducci, 1999):

$$u_r = E_u \left(\frac{\theta}{\cos\varphi} + \frac{d_r \cos\varphi}{HE_v}(v_r + E_v \tan\varphi) + \frac{E_v H C_0 / \cos^3\varphi}{4(v_r + E_v \tan\varphi)} + \frac{E_v C_1 H^2 / \cos^5\varphi}{6(v_r + E_v \tan\varphi)} \right) . \tag{9}$$

Let's make a final simplifying assumption, namely that the linear term of the curvature is negligible, $C_1 \simeq 0$. Hence, our road model is simply a succession of segments of constant curvature. The former equation clearly follows the formulation of a hyperbola with a horizontal asymptote at $v = v_0$:

$$u - u_0 = a(v - v_0) + \frac{b}{(v - v_0)} . \tag{10}$$

In order to enforce parallelism of lane borders, we introduce a new variable x_c, which is the signed distance along the X axis between the camera projection on the road plane and the central axis of the left lane line (figure 3). It follows that $d_r = x_c - L$, $d_l = x_c$ and we have the following couple of equations, for points $(u_l, v_l), (u_r, v_r)$ on the left and right border, respectively:

$$\begin{aligned} u_l &= E_u \left(\frac{\theta}{\cos\varphi} + \frac{\cos\varphi}{HE_v} x_c (v_l + E_v \tan\varphi) + \frac{E_v H C_0 / \cos^3\varphi}{4(v_l + E_v \tan\varphi)} \right) , \\ u_r &= E_u \left(\frac{\theta}{\cos\varphi} + \frac{\cos\varphi}{HE_v} (x_c - L)(v_r + E_v \tan\varphi) + \frac{E_v H C_0 / \cos^3\varphi}{4(v_r + E_v \tan\varphi)} \right) . \end{aligned} \tag{11}$$

Since parameters E_u, E_v, H and φ can be estimated through a camera calibration process (Zhang, 2000; Bouguet, 2008), Eq. (12) is linear with respect to the four unknowns θ, x_c, L and C_0. It can be compactly rewritten as:

$$\begin{bmatrix} 1 & 0 & v'_l & 1/v'_l \\ 1 & -v'_r & v'_r & 1/v'_r \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} u_l \\ u_r \end{bmatrix} , \tag{12}$$

with $v'_r = v_r/E_v + \tan\varphi, v'_l = v_l/E_v + \tan\varphi$ and

$$\theta = \frac{\cos\varphi}{E_u} a_1, \quad L = \frac{H}{E_u \cos\varphi} a_2, \quad x_c = \frac{H}{E_u \cos\varphi} a_3, \quad C_0 = \frac{4\cos^3\varphi}{E_u H} a_4 \ . \tag{13}$$

Note that according to this model, four points, not all on the same line, define *a pair* of hyperbolas sharing the same horizontal asymptote. In addition, they correspond to two parallel curves L meters apart, when backprojected to the road plane. This implies that we are going to fit *both left and right* lane lines at the same time and enforcing parallelism, that is, consistency in the solution. Besides, the sparsity of candidates in one lane side due to occlusions or dashed lane marks can be compensated by those in the other side. The parallelism constraint, however, is a potential drawback at places where the present lane bifurcates, like in highway exits and lane splitting, as shown in figure 6.

3.2 Model fitting

We would like to separate ridge points on each side of the lane in order to adjust the corresponding curve only to them. Since the camera is located at the center of the windshield screen and forward–facing, we can make a guess based on the horizontal coordinate with respect to a fixed image column u_{vanish} (figure 4). This column corresponds to the u–coordinate of the image vanishing point when the vehicle is centered ($x_c = 0$, $\theta = 0$) in a straight lane ($C_0 = 0$). In curves, of course, this simple criterion is reliable only near the vehicle. Therefore, from rows v_{min} to v_{common} we cannot tell which side a ridge point belongs to, and we assume both are possible. It is only below row v_{common} that the image is safely divided by u_{vanish} into left and right lane regions.

A minimum of four points are necessary in order to solve Eq. (12), provided there is at least one point on each curve. If more points are known, we get an over constrained system that is solved in the least–squares sense. The problem, of course, is the selection of the right points among all candidates from the previous detection step. We need a robust technique in the sense of, simultaneously, classify candidate points into lane points (inliers) and not lane points (outliers, at least for that side), and perform the fitting only to the former ones. RANSAC, which stands for Random Sample Consensus (Fischler and Bolles, 1981), is a general estimation technique for such purpose based on the principle of hypotheses generation and verification.

Let be a parametric model M needing at least n points to be instantiated and a set $\mathcal{P}$ of more than n points, possibly containing outliers with regard to the right model instantiation. Also, let be n_{trials} a predetermined maximum number of trials, $d(p, M)$ a distance function between a given point $p = (u, v)$ and an instantiated model M, and t_{dist} a distance threshold. The idea is to randomly select n points from $\mathcal{P}$, instantiate a model M_i with them and measure its support: how many points $p \in \mathcal{P}$ lie in a close vicinity of M_i, that is, $d(p, M_i) < t_{\text{dist}}$. The instantiation with the largest support, or the first one which exceeds a predetermined high support threshold $t_{\text{consensus}}$, defines the set of inliers from which to re–instantiate the model and produce the final result.

In our particular case, $n = 4$, p are candidate ridge points, models M_i are pairs of hyperbolae parametrized by $a_1 \ldots a_4$ and easily instantiated from a set $\mathcal{S}_i$ of four points by solving the linear system of Eq. (12). As for the needed number of trials n_{trials}, let p_w be the a priori probability that a given data point is an inlier. If we want to ensure with probability p_z that at least one of the $\mathcal{S}_i, i = 1 \ldots n_{\text{trials}}$ will only contain inliers, then we need to make at least $\log(1 - p_z)/\log(1 - p_w^n)$ trials (Fischler and Bolles, 1981). We have estimated p_w from thousands of frames captured from sequences corresponding to different lighting conditions (at day and night) and roads (including highways, motorways and local roads). Table 1 lists the value for this and all other fitting parameters which we describe in the following.

Before proceeding to assess the support of M_i, we check whether the lane width obtained from Eq. (13) is a reasonable quantity, that is, within an interval $[L_{\min}, L_{\max}]$. Otherwise, M_i is discarded and a new iteration begins.

The aim of the distance function $d(p, M)$ is to classify points p into inliers and outliers. To do so, we combine two factors: distance to the instantiate lane line and orientation similarity. There is not a simple expression for the geometric distance between a point and a hyperbola since there are up to four lines through a point which are perpendicular to a conic. Instead, we are going to use the Sampson's distance (Sampson, 1982), which lies between the algebraic and the geometric distances in terms of complexity but which gives a close approximation to the later. Let be $\mathbf{p} = (u, v, 1)^\top$ a point p in homogeneous coordinates and $\mathtt{C}$ a symmetric 3×3 matrix. Then, $\mathbf{p}^\top \mathtt{C}\mathbf{p} = 0$ is the equation of a conic section. The Sampson's distance is defined by:

$$d_S(p, \mathtt{C}) = \frac{(\mathbf{p}^\top \mathtt{C}\mathbf{p})^2}{4((\mathtt{C}\mathbf{p})_1^2 + (\mathtt{C}\mathbf{p})_2^2)} \ , \tag{14}$$

where $(\mathtt{C}\mathbf{p})_i$ denotes the i–th component of the vector $\mathtt{C}\mathbf{p}$. For those ridge points satisfying $d_S(p, \mathtt{C}) < t_{\text{dist}}$, a second test is applied before being admitted as inliers: at the point p' on $\mathtt{C}$ closest to p, the line $\mathbf{l} = \mathtt{C}\mathbf{p}'$ tangent to the conic must be parallel to the dominant image orientation at p, or equivalently, perpendicular to the dominant gradient orientation field $\mathbf{w}'_{\sigma_\mathrm{d},\sigma_\mathrm{i}}$ at that point. A maximum deviation of α is allowed. For the sake of computational simplicity p' is taken as the point of $\mathtt{C}$ in the same row as p.

A final observation must be made concerning the lane model of Eq. (12). In it we supposed the pitch angle φ to be known from the calibration process, but it actually suffers variations around its nominal value due to non–planar roads, acceleration, brake actioning etc. To account for this fact, quite influential in the instantiated model because it changes its horizontal asymptote, we test several possible values for φ, taking n_φ equispaced samples within $\varphi \pm \Delta\varphi$, for a certain margin value $\Delta\varphi$ (again, refer to table 1 for the actual parameters values).

4 Validation and results

As pointed out in a recent survey on video–based lane departure warning (McCall and Trivedi, 2006), results in the literature are often presented only in the form of several frames, where the reader can check the correspondence between detected lane lines and real lane markings. We also present results in this qualitative way, but just to show examples of challenging situations due to occlusions, shadows, reflections and poor lighting conditions, both in daytime and nighttime, where our method succeeds, at least by visual comparison (figure 5 and 6). Complete sequences from which these frames have been extracted can be viewed at
`http://www.cvc.uab.es/adas/projects/lanemarkings/IJAT/videos.html`.

However, since our fitted model has a direct relation to geometrically meaningful parameters of interest in the context of ADAS, we base the evaluation on the comparison of these computed values with the actual ones. And here we face the main difficulty in obtaining quantitative results for this kind of work: the lack of ground truth, that is, the precise knowledge of the road shape, the camera position and the viewing direction at each frame. True, they can be approximated by means of additional sensors like differential or high precision GPS, accelerometers etc. (Wang, Schroedl, Mezger, Ortloff, Joos and Passegger, 2005), but the construction of digital road maps at lane line resolution is still a research issue in itself and out of the scope of this paper. Therefore, we have resorted to build a simulator which generates sequences of synthetic but realistic images from exactly known road geometry and camera dynamics. This has the evident advantage of controlling every possible factor, from the 3D road shape and contrast to the camera trajectory and pose along time.

Specifically, the simulator, implemented in Matlab, is based on four models:

1. Road photometry. The road is composed of two lanes and thus it has two border and one central lane lines. Each of them can be continuous or dashed. Variable lighting conditions are simulated by sudden gray level changes of long road patches (figure 7b).

2. Road geometry. The road is divided into segments of varying length and constant but random curvature, and a linear interpolation of curvature is performed at their ends to get smooth transitions. Likewise, it is also divided into segments of fixed random slope, independently of curvature. Slope is later smoothed to avoid sudden unrealistic changes.

3. Camera model (intrinsic parameters). The camera is simulated by a central projection, according to the pin–hole model. The principal point is located at the image center. No radial distortions have been considered for the sake of simplicity. A focal length of 1200 pixels for both axes and a resolution of 640×480 pixels yield a field of view very similar to that of real cameras we have used. However, images are sampled to half resolution to speed up processing (40 ms/frame in a 2.0 GHz Pentium IV).

4. Camera dynamics (extrinsic parameters). Camera location changes due to the simulated vehicle motion which is longitudinally 1 meter per frame (thus, at 30 fps the simulated vehicle speed is 108 Km/h). In order to determine the lateral displacement with respect the lane central axis, the whole road is divided into segments of varying length. Each is assigned a constant lateral offset, and then a Gaussian smoothing is performed to avoid sudden vehicle direction changes. With regard to camera pose, we have fixed the roll angle to zero (the horizon line is parallel to the image horizontal axis) and set at each frame the yaw angle θ such that the camera is always forward facing, that is, the optical axis is parallel to the vehicle trajectory tangent (figure 3). Finally, the pitch angle φ has not been fixed, since it is responsible for the horizon line vertical motion, which is not static in real sequences. Besides, it turns out that this parameter is quite influential on the results. Thus, we have randomly varied the pitch angle so as to mimic the effects of an uneven road surface and of acceleration and brake actioning, both observed in real sequences. Specifically, the pitch variation is generated by adding two random signals: the first one of high frequency and small amplitude ($\leq 0.2°$) and the second one of low frequency but larger amplitude (between $0.5°$ and $1°$), which account respectively for the two former pitch variation sources (figure 7d).

Table 2 contains the specific values for the most relevant parameters of the simulator, some of which correspond to a real road. We have performed several tests on synthetic sequences in order to calculate the error in the estimation of C_0, L, θ, x_c and also φ. Figure 7a shows the whole 5 Km long synthetic road from which quantitative results have been drawn at each meter. Figure 7b shows a typical frame. Error computation has not been the unique goal of testing, but we wanted also to assess the error contribution due to the departure from the assumed road model and to errors in image lane line detection. Specifically, we have conducted the following tests, increasingly approaching the real testing scenario:

- Non–ideal road. We have generated a sequence of a synthetic road with piecewise constant curvature and slope. Camera pitch has been fixed to a known nominal value. Lane lines detection is ideal, since we obtain one point per row points from the central projection of the generated lane lines world coordinates, instead of the road image itself.

- Non–ideal camera. Like the former case but now the camera pitch is allowed to vary from its nominal value, in the way explained above.

- Non–ideal detection. The only difference with the previous case is that detection is performed on actual images of the sequence. This test assesses the influence of detection, performed as described in sections 2 and 3.2.

- Best pitch search. Like the former case of non–ideal detection, but now we do not rely on the nominal pitch. Instead, we test a fixed number of values n_φ equispaced around the nominal value φ. This is equivalent to look for the best horizon line near that produced by the nominal pitch angle.

Figs. 8 to 11 (top) show that the difference between computed parameters

x_c, C_0, θ, L and their corresponding ground truth is very small if the road follows the ideal model of constant (but also linearly varying) curvature, flat surface and known camera pitch angle, thus confirming the suitability of the proposed method. At the same time, deviations from this ideal case due to sudden slope change introduce large errors, though logically localized in time. The larger the slope variation, the larger the error, but the sign of change does matter. The slope variation at times $t = 450$ and 850 is almost equal (figure 7c) but the error is much smaller in the first case, for all four parameters. The reason is that, when the camera approaches a negative slope change (the vehicle goes uphill and almost reaches the 'top'), the number of image rows depicting road surface is reduced (figure 12). In principle, this should not cause any problem, since the detection of the lane line points is ideal, that is, the $(u_l, v_l), (u_r, v_r)$ are exact. However, these points are taken only from the road visible region, one per row and side. If they have similar v coordinate, the over constrained linear system built by stacking pairs of equations (12) becomes ill–conditioned. When the slope change is positive (the camera faces a 'ramp', like in figure 7b), the lane line points do not fit into the flat road model and, consequently, there is some error, but the system is well conditioned.

The second row of Figs. 8 to 11 shows the error introduced by variations in the camera pitch (figure 7d). At frames where the pitch variation has its largest peaks, the error is small for x_c, moderate for L but large for C_0 and θ. The reason is that x_c and L are local road measures very close to the camera position and thus not affected by the global lane line shape, specially its shape at a large distance, close to the horizon line. On the contrary, C_0 and

θ do depend on the global shape (according to the road model, the curvature is supposed to be constant) which is in turn dependent on the shared lane lines horizontal asymptote.

When the lane line points (u_l, v_l), (u_r, v_r) are extracted from the images, in the non–ideal detection scenario, the former two types of errors are somehow amplified for all four parameters (third row of Figs. 8 to 11). In addition, a small amplitude noise appears everywhere. This latter could be attributed to the detection process, but experiments have shown that it is mainly due to the small amplitude pitch variation. We have tried to minimize the effect of pitch changes (both large and small) by considering φ another parameter to estimate, as explained in the next paragraph. Recall that the last testing scenario consists of looking for the best pitch angle (in terms of RANSAC, maximize the size of the consensus set) among n_φ possible values within the nominal pitch $\pm\Delta\varphi$. However, the estimated φ is *not* used to recompute the four parameters, it is just a way to check the success of the best pitch search procedure when comparing the estimated φ to the ground truth (figure 13).

The fourth row of Figs. 8 to 11 shows the result for $n_\varphi = 7$ and $\Delta\varphi = 1°$. From figure 13 we conclude that the best pitch search is often able to correctly estimate the pitch (the four largest pitch variations are well detected), but not always. The most prominent errors are localized around the four slope changes, where this simple approach of guessing the best pitch fails. Elsewhere, a sort of impulsive error is observed, caused by a small number of inliers. In addition, depending on the value of n_φ, the estimated φ suffers from a quantization noise: the $2°$ interval is too wide for just 7 possible values. Increasing n_φ yields a better estimation but the computational cost precludes a high processing rate. In spite of it, a causal median filter (median of a number of pitch estimations before the current frame) produces an acceptable result, even for $n_\varphi = 7$. Likewise, the causal median filtering of x_c, C_0, θ and L (bottom row of Figs. 8 to 11) produces more accurate values due to the impulsive and zero mean nature of the error induced by the pitch estimation. Finally, figure 14 shows the root–mean square error (RMSE) between computed and ground truth for $n_\varphi = 1, 3, 7, 41$ and also for their median filtered versions. Whereas there is only a slight improvement, or even no improvement at all, when n_φ increases, the error of the filtered parameters clearly decreases. Therefore, it seems that it does not pay to look for the best pitch if no filtering is performed afterward. But the important thing to note is that even in the simplest case of $n_\varphi = 1$, the RMSE of x_c, θ, C_0 and L is only 25 cm, 1.1°, 0.0027 m^{-1} and 22cm, respectively.

5 Conclusions

We have developed a new method for the extraction of lane lines from video sequences, with robustness and quantitative evaluation as the main considerations. Robustness is achieved both in the feature detection phase, where we employ an image feature well suited to this problem, and in the model fitting phase, which we have addressed with a RANSAC approach. This method relies just on images, that is, we do not take into account data from other vehicle sensors, like the steering angle, yaw rate or vehicle speed. In addition, each frame is processed independently of the others, since our goal has been to build a 'baseline' system on which to later add filtering, to enforce temporal consistency, and data fusion, to improve reliability.

Our lane line extraction method has the advantage of computing four road and vehicle trajectory parameters which are of interest in the context of ADAS: road curvature, lane width, lateral vehicle offset and heading angle with respect the road medial axis. We have compared the computed values with ground truth from a synthetic but realistic road, in several testing scenarios, increasingly closer to a real test. From the experiments we conclude that it is possible to compute reasonable estimations of these parameters, even in the case where the road does not exactly follow the assumed model of flatness, constant curvature and known camera pitch. We have also performed extensive visual testing on real sequences from different roads, with varying traffic density and lighting conditions (day, night, tunnels, cast shadows), and also recorded with several camaras (CCD, CMOS). Results have shown the robustness of our method to these factors.

The weak point of our method, with regard the correct computation of the road geometric parameters, is the estimation of the pitch angle. Its prediction on the basis of previous frames seems a promising way. Future work also includes the design of a post–processing phase which incorporates the temporal continuity of detected lane lines and parameters. Some promising results have already been obtained with a Kalman filter.

Acknowledgments

This research has been partially funded by grants TRA2007–62526/AUT of the Spanish Education and Science Ministry, and Consolider Ingenio 2010:

MIPRCV (CSD2007–00018).

References

Bertozzi, M., Broggi, A. and Fascioli, A. (2000), 'Vision–based intelligent vehicles: state of the art and perspectives', *Robotics and Autonomous Systems* **32**, 1–16.

Bouguet, Y. (2008), 'Camera calibration toolbox for matlab', `http://www.vision.caltech.edu/bouguetj/`.

Dickmanns, E. and Mysliwetz, B. (1992), 'Recursive 3D road and relative ego–state recognition', *IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. , pg. ,* **14**(2), 199–213.

Fischler, M. and Bolles, R. (1981), 'Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography', *Communications of the ACM* **24**(6), 381–395.

Guiducci, A. (1999), 'Parametric model of the perspective projection of a road with application to lane keeping and 3D road reconstruction', *Computer Vision and Image Understanding* **73**(3), 414–427.

Jung, C. and Kelber, C. (2005), 'Lane following and lane departure using a linear–parabolic model', *Image and Vision Computing* **23**(13), 1192–1202.

López, A., Lloret, D., Serrat, J. and Villanueva, J. (2000), 'Multilocal creaseness based on the level–set extrinsic curvature', *Computer Vision and Image Understanding* **77**(2), 111–144.

McCall, J. and Trivedi, M. (2006), 'Video–based lane estimation and tracking for driver assistance: survey, system and evaluation', *IEEE Trans. on Intelligent Transportation Systems* **1**(7), 20–37.

Sampson, P. (1982), 'Fitting conic sections to very scattered data: an iterative refinement of the bookstein algorithm', *Computer Graphics and Image Processing* **18**, 97–108.

Wang, J., Schroedl, S., Mezger, K., Ortloff, R., Joos, A. and Passegger, T. (2005), 'Lane keeping based on location technology', *IEEE Trans. on Intelligent Transportation Systems* **6**(3), 351–356.

Zhang, Z. (2000), 'A flexible new technique for camera calibration', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(11), 1330–1334.

Tables

detection	$\sigma_{\mathrm{dx}}(u), u = u_{\mathrm{vanish}} \ldots M$	$0.5 \ldots 6.0$
	σ_{dy}	0.5
	σ_i	0.5
	t_{gradient}	2.0
fitting	u_{vanish}	160
	v_{min}	137
	v_{common}	187
	n_{trials}	$<$1000 for $p_Z = .90$
	$[L_{\mathrm{min}}, L_{\mathrm{max}}]$	$[220, 300]$ pixels
	α	$15°$
	t_{dist}	2 pixels
	$t_{\mathrm{consensus}}$	8 points
	$\pm\Delta\varphi$	$\pm 1°$ (approximately ± 10 rows)

Table 1: Parameter values of the feature detection and model fitting phases. Original images of 640×480 pixels are sampled to half resolution in order to speed up processing. Row v_{min} is at 35 to 40 meters from the camera.

camera model	image dimensions (cols×rows)		640 × 480
	bits/pixel		8
	focal lengths E_u, E_v		1200 pixels
	camera height H		1.6 m
	nominal camera pitch φ		1.6°
camera (vehicle) motion	maximum amplitude of low frequency pitch noise		1.0°
	idem, high frequency		0.2°
	roll angle		0°
	maximum lateral offset in % of lane width		80%
road geometry and photometry	length		5 Km
	length of segments of constant curvature and slope		300–600 m
	maximum slope (magnitude)		7%
	minimum radius of curvature		50.0 m
	lane width		3.65 m
	road border line	width	0.2 m
		length	20 m
		gap	4 m
	inter–lanes line	width	0.15 m
		length	4 m
		gap	7 m
	mean road surface intensity (1=white,0=black)		0.2
	idem, lane line		0.9

Table 2: Main simulator parameters.

Figures

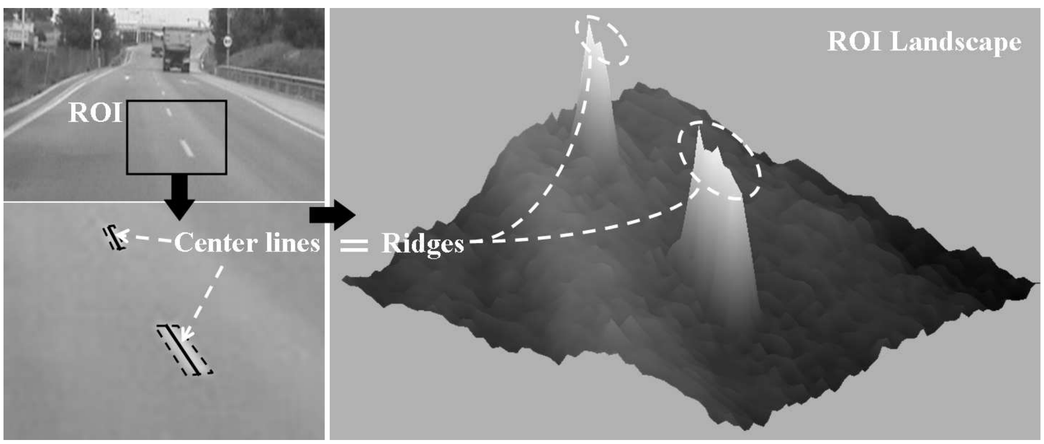

Figure 1: Left: road image with a region of interest (ROI) outlined. Right: intensity of ROI seen as a landscape, where lane markings resemble mountains and ridges correspond to the center of the lane markings.

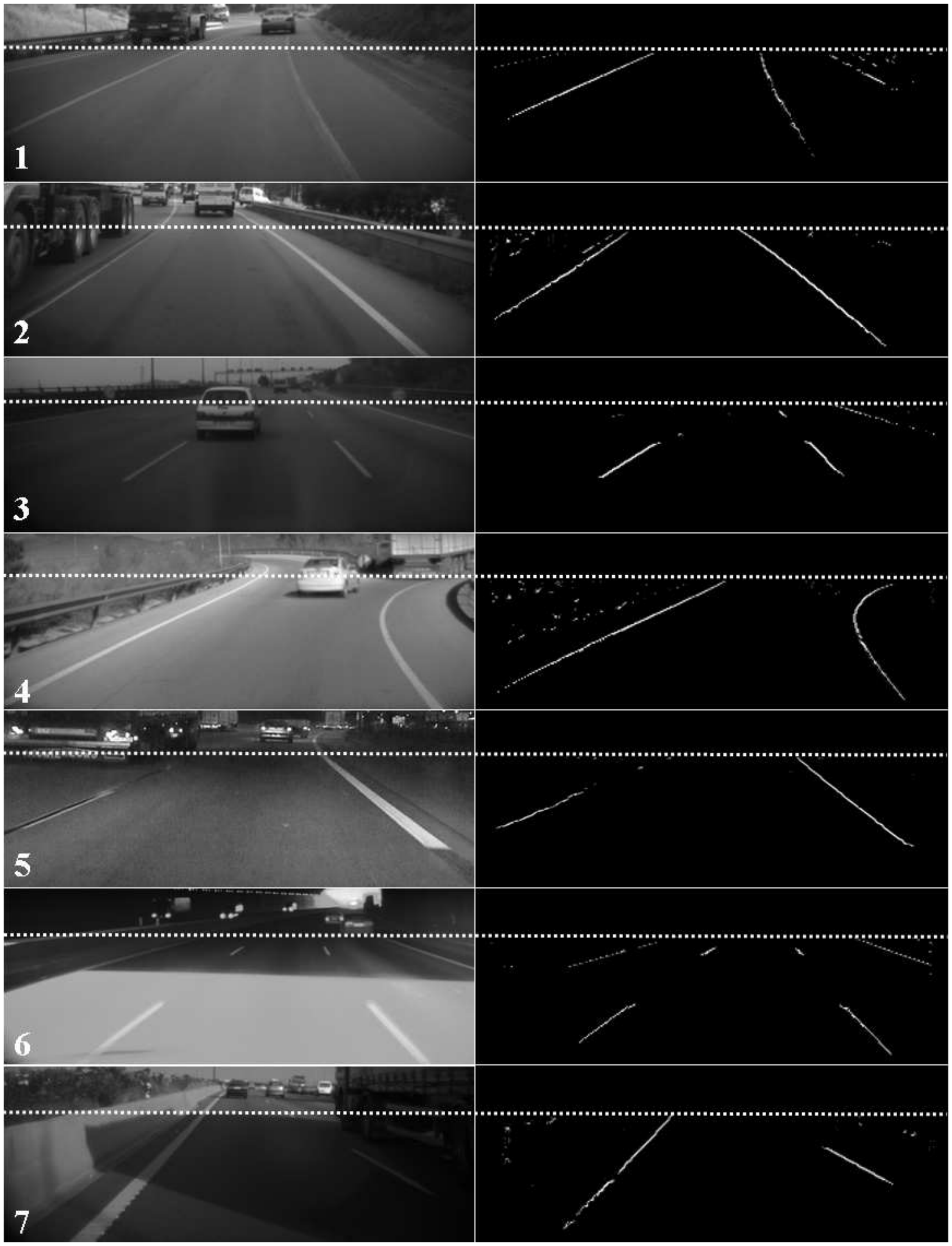

Figure 2: Detection in challenging conditions: (1) worn off paint, (2) tire marks, (3) white vehicle, (4) high curvature, (5) at night, (6) entering a tunnel, (7) shadow cast by a truck. Right column: ridgeness of points selected as candidate lane markings. The dotted line is the fixed initial row under which candidate points are sought ($v_{\min}$ in figure 4)

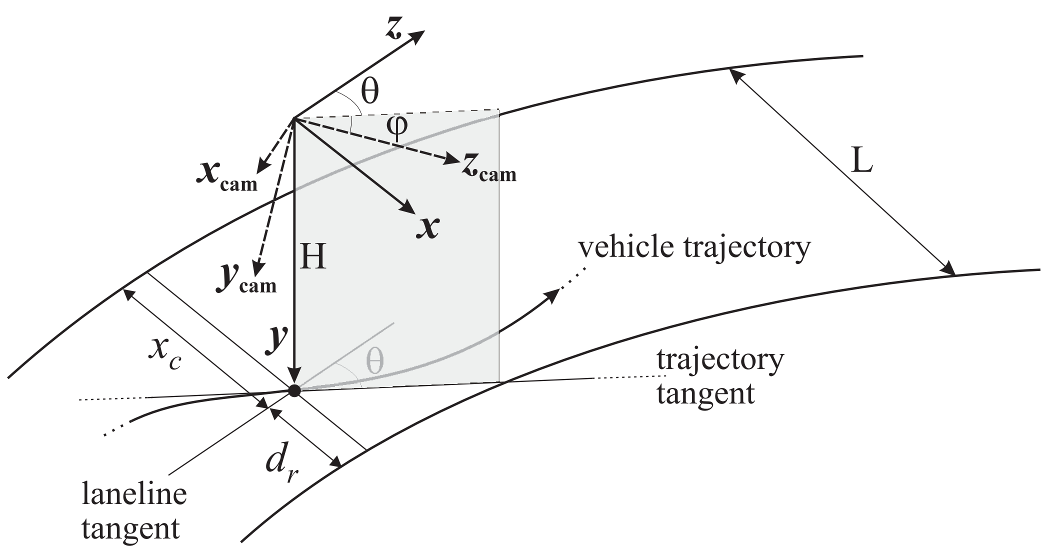

Figure 3: Image acquisition geometry.

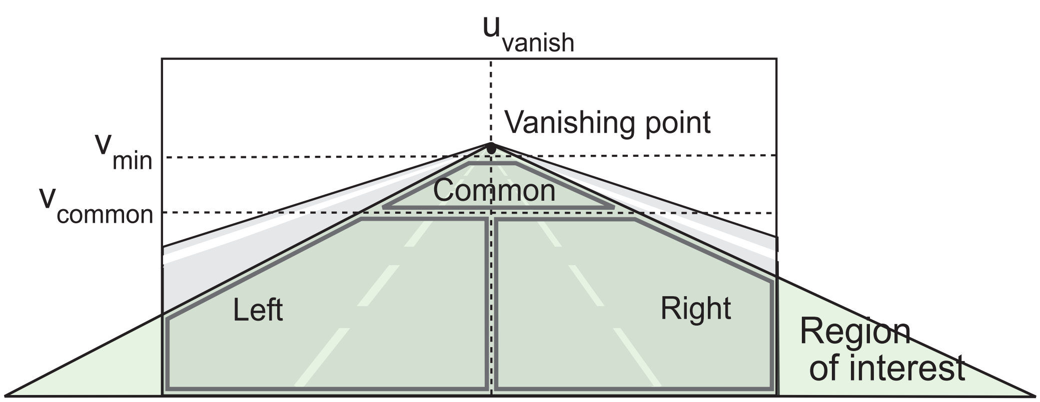

Figure 4: The detected features are divided into three groups, depending on their position in the image. Ridgeness is computed from row ($v_{\min}$) downwards.

Figure 5: Segmented lane curves on frames acquired by different cameras: (1–3) dashed lines, (4) occlusion. (5,6) tunnel exit and entrance, (7) horizontal marks, (8) cones, (9,10) special road marks, (11,12) shadows, (13,14) night images with low contrast and reflections.

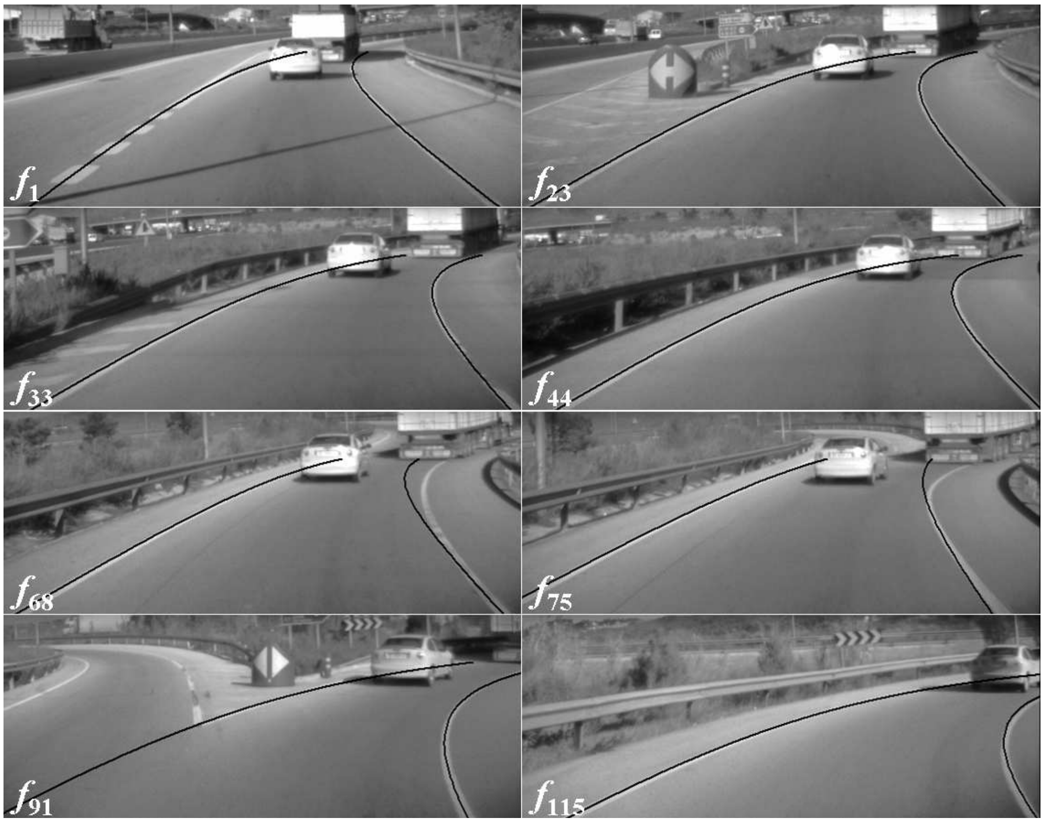

Figure 6: Several frames of a sequence at 10 fps. The vehicle leaves a highway, finds a bifurcation and turns to the right. Notice that in spite of the high curvature of the lane lines the method is able to detect them very well, except in the frames where the two lines are not parallel (f_{68} and f_{75}) due to the approaching bifurcation.

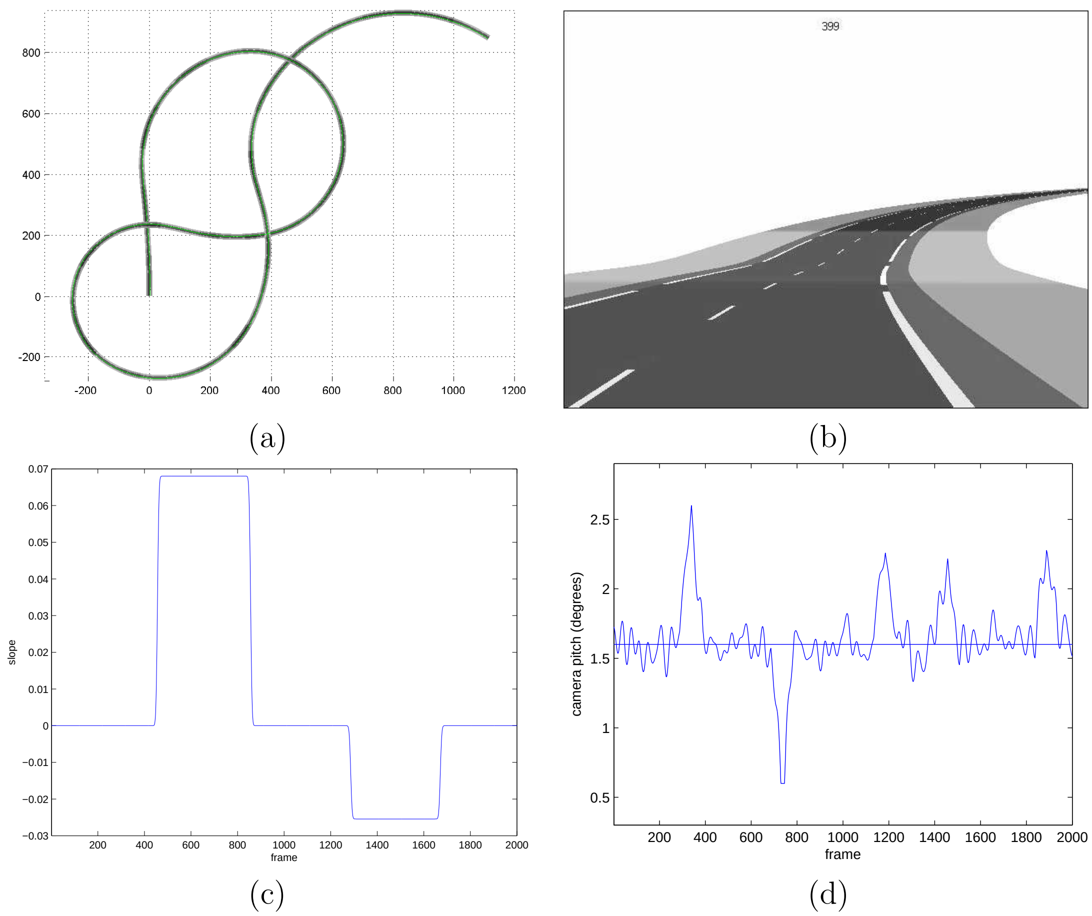

Figure 7: (a) Planar projection of the synthetic 3D road, (b) sample frame of a place close to changes in slope, curvature and road contrast, (c) road slope, (d) nominal and real pitch angle φ. For viewing the complete video sequence, see `http://www.cvc.uab.es/adas/projects/lanemarkings/IJAT/videos.html`.

(a)

(b)

(c)

(d)

(e)

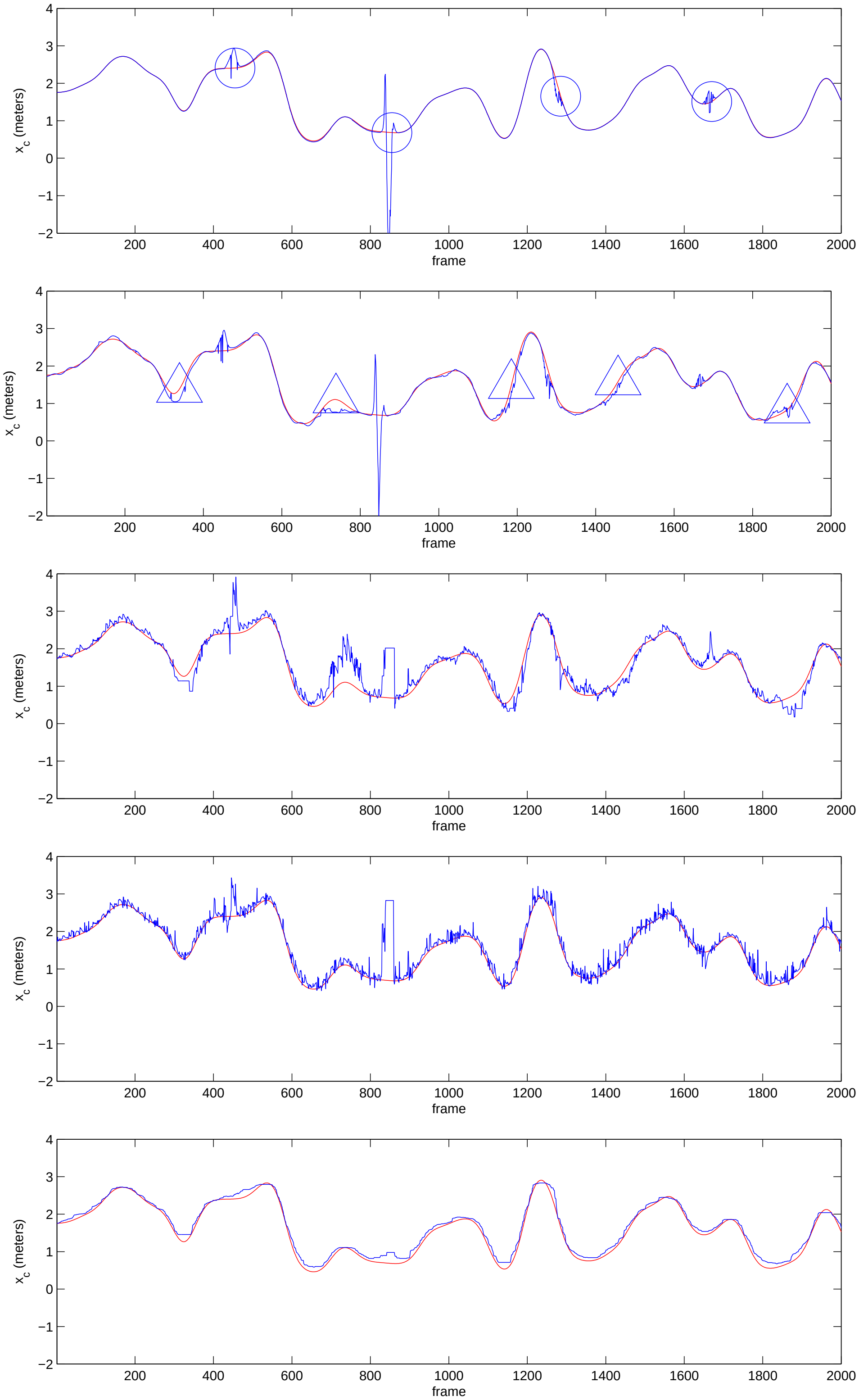

Figure 8: Ground truth and computed x_c in meters for (top to bottom) : a) non–ideal road, b) non–ideal camera, c) non–ideal detection with nominal pitch (n_φ=1), d) non–ideal detection trying n_φ=7 pitch angles around nominal camera pitch and median filtering of this latter result, e) causal median filtering of (d). To achieve a proper zoom only the first 2 Km are shown. Circles and triangles mark the locations of slope and large pitch variation, respectively.

(a)

(b)

(c)

(d)

(e)

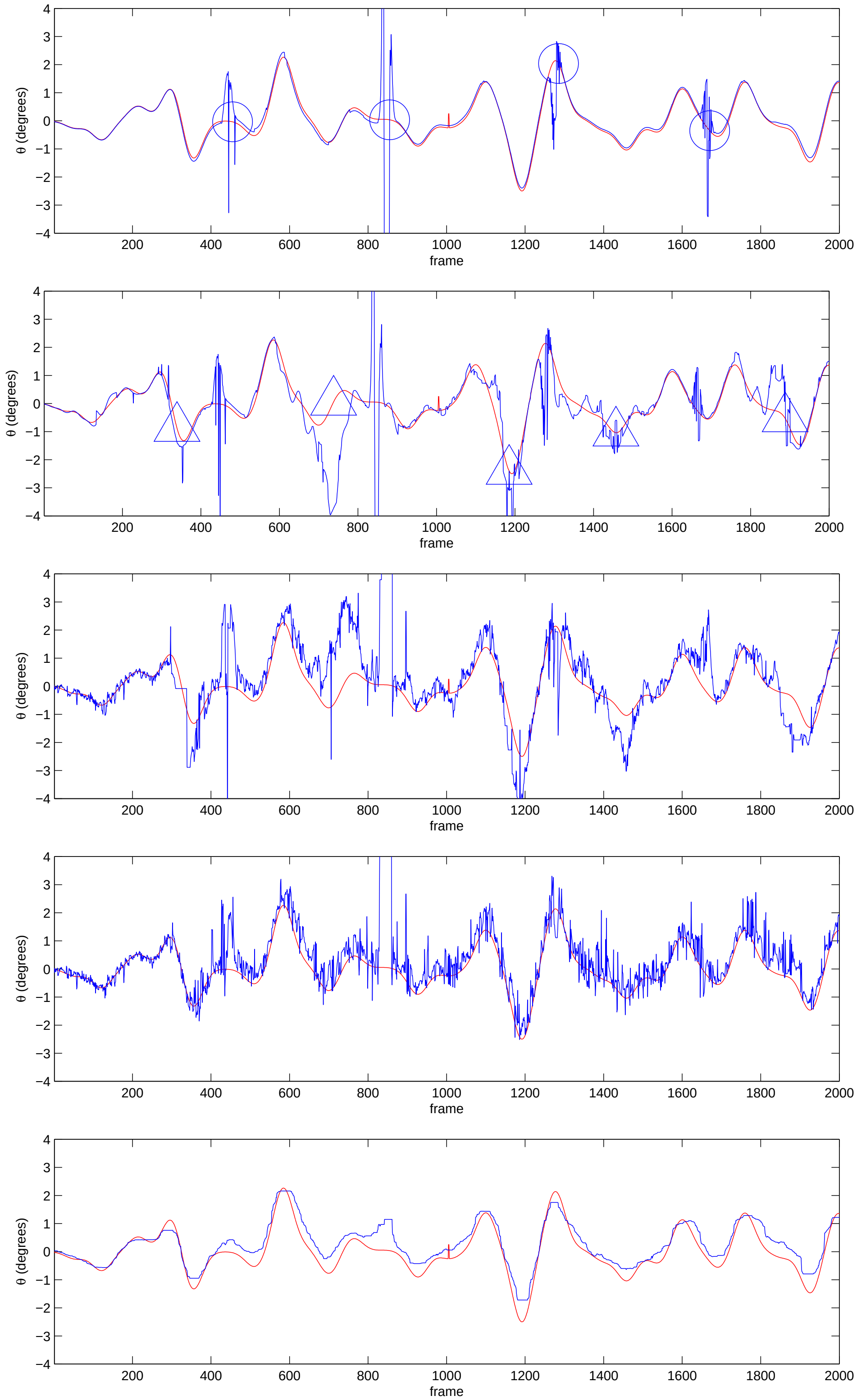

Figure 9: Ground truth and computed θ in degrees, same cases as figure 8.

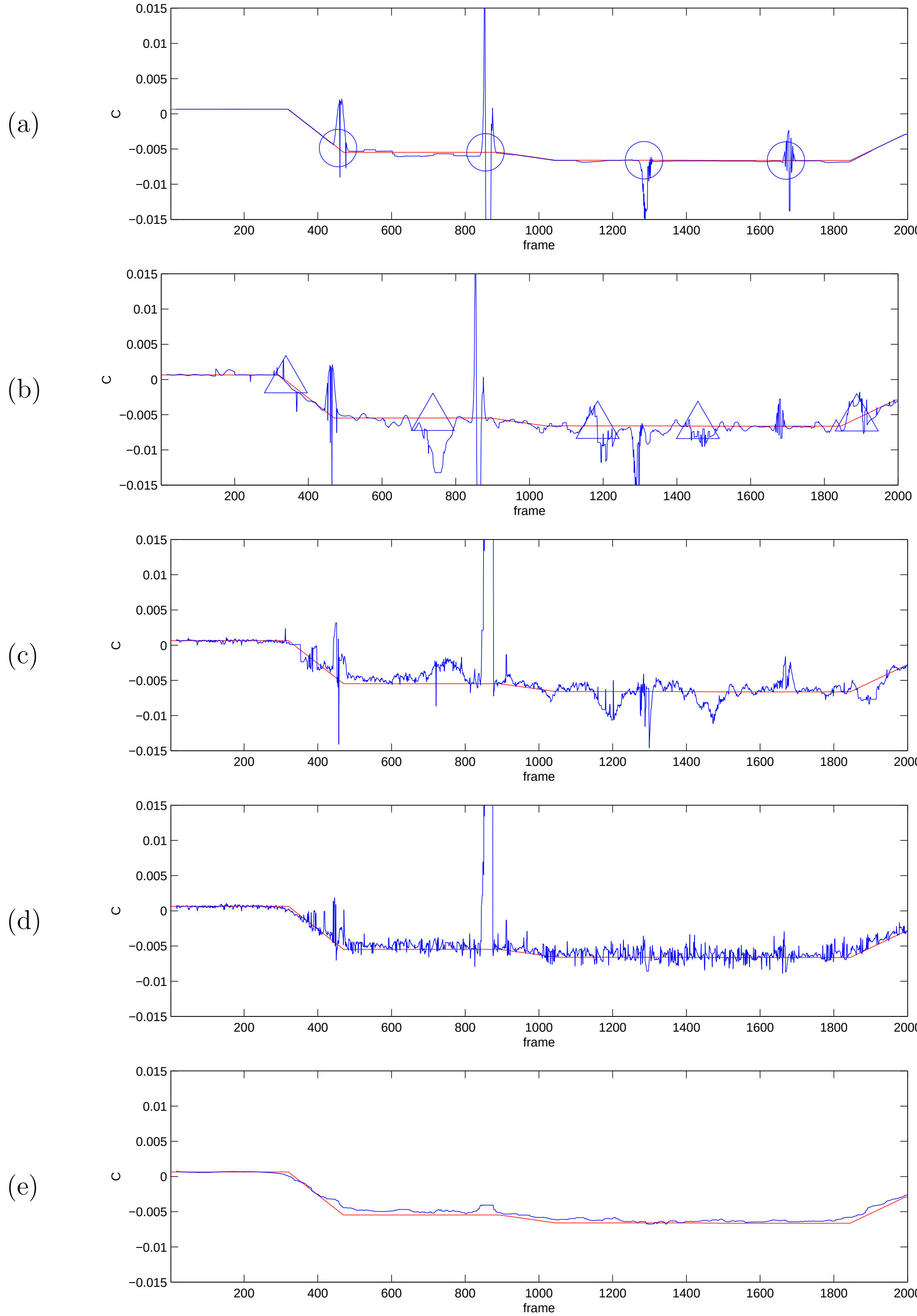

Figure 10: Ground truth and computed C_0 in m^{-1}, same cases as figure 8.

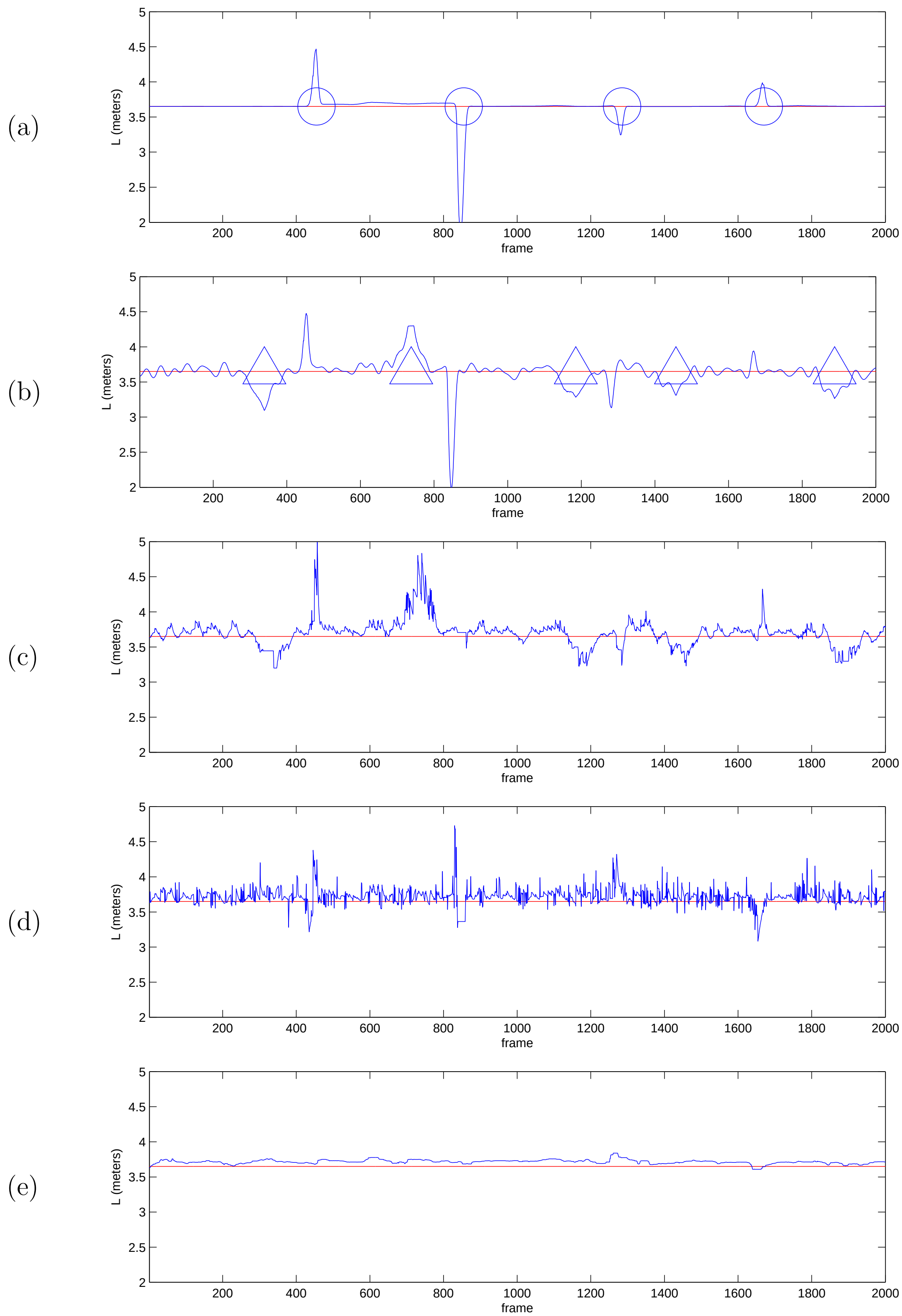

Figure 11: Ground truth and computed L in meters, same cases as figure 8.

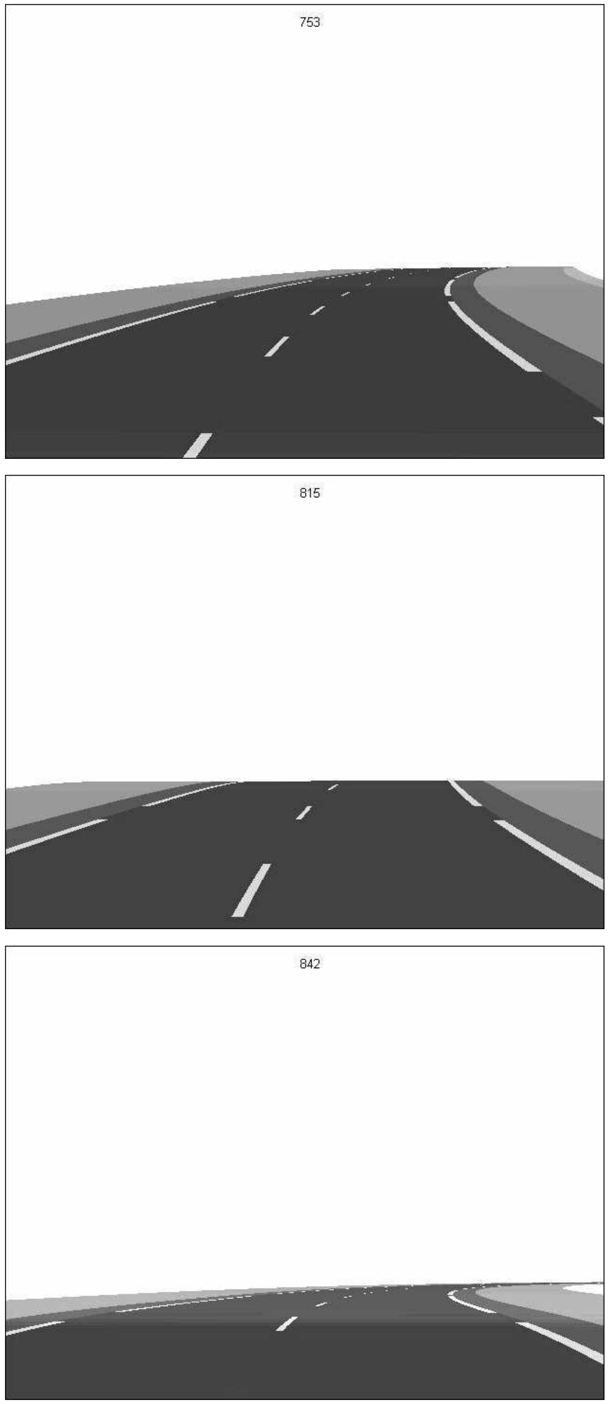

Figure 12: A negative slope change shrinks the road region from the image, causing large errors in the computed parameters.

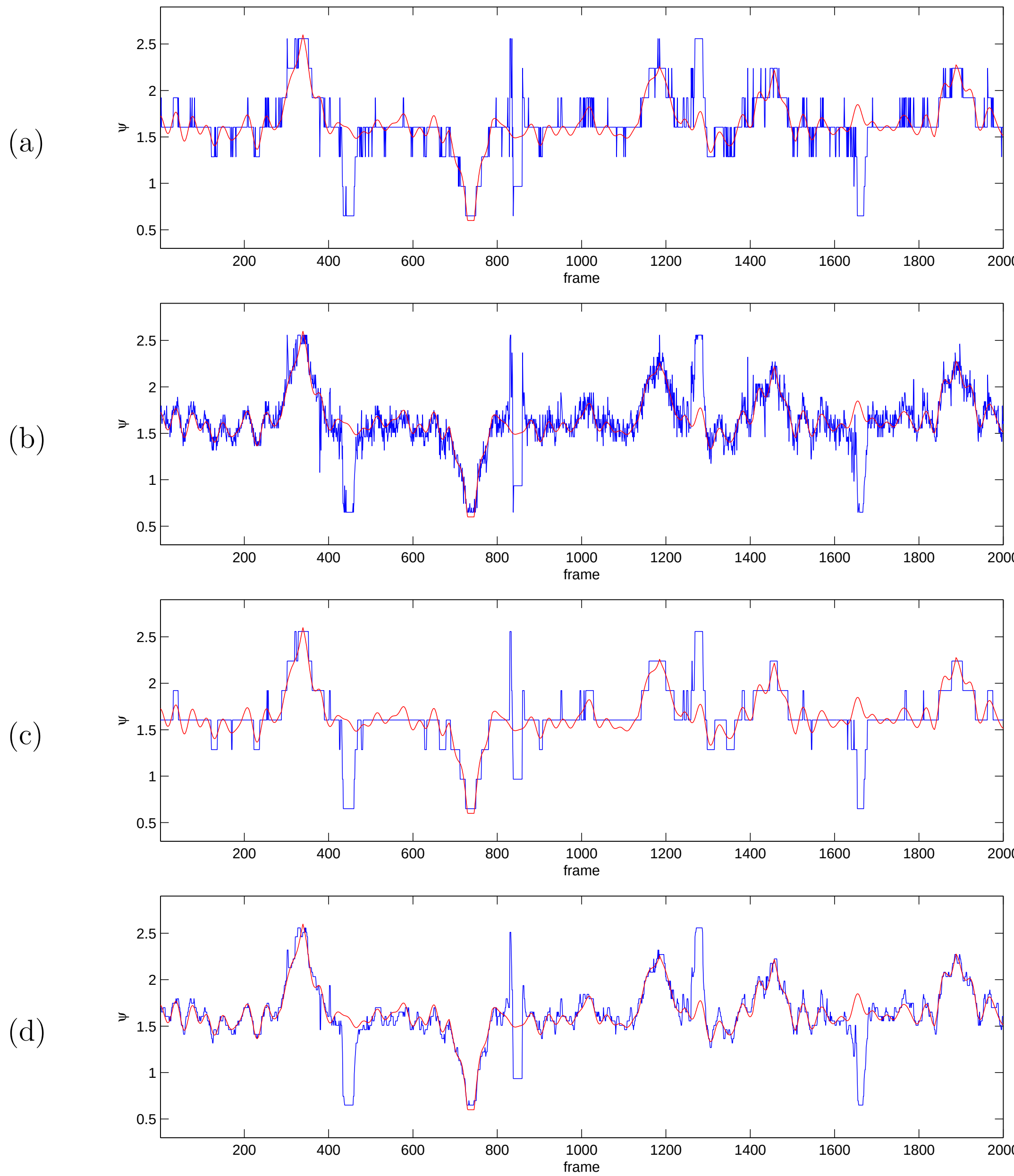

Figure 13: Ground truth and computed φ in degrees, for (a) n_φ=7, (b) n_φ=41, (c), (d) causal median filtering of (a) and (b).

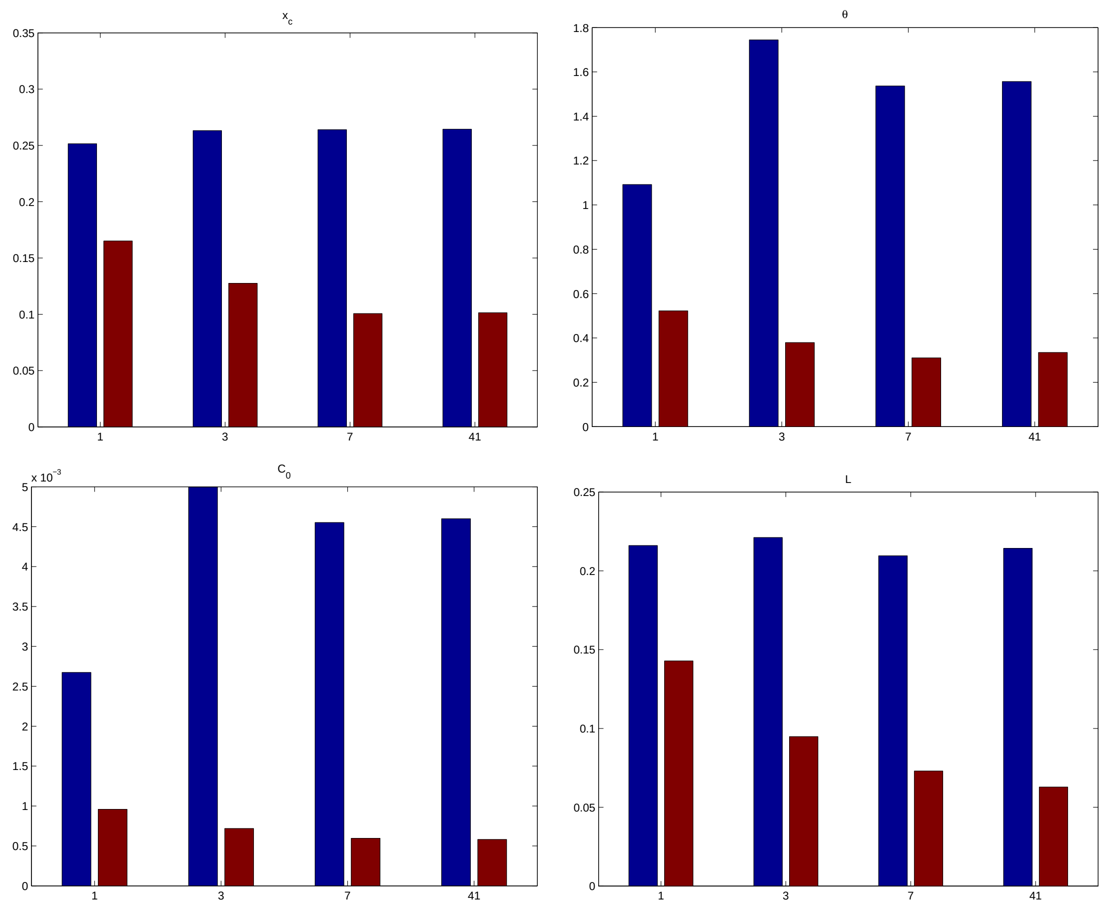

Figure 14: Root-mean square error of computed x_c, θ, C_0 and L (angles in degrees) for the case of non–ideal detection and n_φ =1 (non–ideal detection), 3, 7 and 41 (best pitch search). In each pair of bars, left bar corresponds to the computed value and right bar to its median filtered version.