An Efficient Approach to Onboard Stereo Vision System Pose Estimation

Angel Domingo Sappa, *Member, IEEE*, Fadi Dornaika, Daniel Ponsa, David Gerónimo, and Antonio López

***Abstract*—This paper presents an efficient technique for estimating the pose of an onboard stereo vision system relative to the environment's dominant surface area, which is supposed to be the road surface. Unlike previous approaches, it can be used either for urban or highway scenarios since it is not based on a specific visual traffic feature extraction but on 3-D raw data points. The whole process is performed in the Euclidean space and consists of two stages. Initially, a compact 2-D representation of the original 3-D data points is computed. Then, a RANdom SAmple Consensus (RANSAC) based least-squares approach is used to fit a plane to the road. Fast RANSAC fitting is obtained by selecting points according to a probability function that takes into account the density of points at a given depth. Finally, stereo camera height and pitch angle are computed related to the fitted road plane. The proposed technique is intended to be used in driver-assistance systems for applications such as vehicle or pedestrian detection. Experimental results on urban environments, which are the most challenging scenarios (i.e., flat/uphill/downhill driving, speed bumps, and car's accelerations), are presented. These results are validated with manually annotated ground truth. Additionally, comparisons with previous works are presented to show the improvements in the central processing unit processing time, as well as in the accuracy of the obtained results.**

***Index Terms*—Camera extrinsic parameter estimation, ground plane estimation, onboard stereo vision system.**

I. Introduction

SEVERAL vision-based advanced driver-assistance systems (ADASs) have been proposed in the literature during recent years. These systems can be broadly classified into two different categories, namely, *monocular and stereo*, each one having its own advantages and disadvantages.

Monocular systems[1] have a lower cost and a higher working rate than stereo systems. However, the latter system provides rich 3-D information of the observed scene, which allows it to face up to problems that cannot be tackled with monocular systems without having prior knowledge of the environment. Drawbacks of stereo vision systems, such as their lower working rate, are related to the current technology capabilities, whereas drawbacks of monocular vision systems are due to their monocular nature. Therefore, taking into account the fast evolution of technology in the computer vision field, it is assumed that most of the stereo system drawbacks will soon be surpassed. Actually, some of the current commercial stereo vision systems can provide dense 3-D maps at rates of 30 frames/s or higher (e.g., [1] and [2]).

A system is either stereo or monocular, and one important demand of vision-based ADAS is the estimation of the pose of the acquisition system with respect to the road being observed—note that this road does not necessarily correspond to the vehicle supporting ground surface but to a plane fitting car's neighborhood. This pose information is of particular interest for applications such as obstacle avoidance, vehicle/pedestrian detection, or traffic sign recognition since it allows the establishment of a precise mapping between the acquired images and the observed road, thus reducing the processing area required to tackle a desired final application.

The estimation of this pose information relative to the observed road is an ill-posed problem for monocular vision systems. Due to this, a quite common approach is to consider a discrete set of feasible camera poses instead of estimating it and use it to solve a given task. For instance, this is the approach followed by Ponsa *et al.* [3] in the context of vehicle detection. On the other hand, some authors propose to estimate the camera pose with respect to the road assuming prior knowledge of the acquired environment. For instance, Coulombeau and Laurgeau [4] assume that the road observed on images has a constant known width; Liang *et al.* [5] assume that the vehicle is driven along two parallel lane markers; and Bertozzi *et al.* [6] assume that a calibration process determines an initial horizon line position, which is then updated over time by matching the edge information around it in successive frames. Obviously, the performance of these methods depends on fulfillment of assumptions, which, in general, cannot be taken for granted.

Manuscript received October 25, 2006; revised June 25, 2007, November 20, 2007, February 28, 2008, and March 15, 2008. This work was supported in part by the Government of Spain under the Ministry of Education and Science (MEC) under Research Project TRA2007-62526/AUT and Research Program Consolider Ingenio 2010: Multimodal Interaction in Pattern Recognition and Computer Vision (CSD2007-00018). A. D. Sappa was supported by the Ramón y Cajal Program. D. Gerónimo was supported by the MEC under Grant (FPI) BES-2005-8864. The Associate Editor for this paper was S. Nedevschi.

A. D. Sappa, D. Ponsa, and D. Gerónimo are with the Computer Vision Center, Universitat Autònoma de Barcelona (UAB), 08193 Bellaterra, Barcelona, Spain.

F. Dornaika is with the French National Geographical Institute, 94165 Saint-Mandé, France.

A. López is with the Computer Vision Center and Computer Science Department, UAB, 08193 Bellaterra, Barcelona, Spain.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2008.928237

[1]Note that by *system*, we refer not only to the camera but also to the whole acquisition hardware and required software for obtaining the images (e.g., color or intensity images in the monocular case and color or intensity images plus (x, y, z) information in the stereo case).

1524-9050/$25.00 © 2008 IEEE

By using a stereo system, the pose estimation is no longer an ill-posed problem; as 3-D information can be extracted from the stereo pair, the pose of the acquisition system referred to the environment can be determined. Broadly speaking, two different stereo matching schemes are used to compute 3-D information: 1) matching edges and producing sparse depth maps or 2) matching all pixels in the images and producing dense depth maps [7]. The final application is used to define whether preference is given to edge-based correspondences (e.g., [8] and [9]) or to dense stereo correspondences (e.g., [10] and [11]). In general, for a successful reconstruction of complex surfaces, it is essential to compute dense disparity maps defined for every pixel in the entire image. However, when real-time constraints are imposed, sparse depth maps are very appealing since their computation and posterior processing will be faster. Examples of this strategy are in [9], [12], and [13]. Although attractive, from the point of view of reduced processing time, the use of sparse depth maps is limited since other ADAS modules will not take advantage of those application-oriented 3-D data points.

Whether computing dense or sparse data, this information has been used in the camera pose estimation problem by means of two different representation frameworks: 1) the disparity space and 2) the Euclidean space. In the disparity space, the 3-D information of the scene is inherently represented in a disparity image, which details the distance along the epipolar line between corresponding elements in the stereo pair. On the other hand, in the Euclidean space, the 3-D coordinates corresponding to these disparity values are explicitly expressed.

Proposals formulated in the disparity space commonly generate what has been named as a v-disparity image [14]. A v-disparity image is computed by accumulating the points with the same disparity value that occur on a given image line. The interesting point in this representation is that planes in the Euclidean space become straight lines in the v-disparity image. By identifying this straight line, the observed road can be characterized, and consequently, the camera pose can be estimated. From this perspective, Labayrade *et al.* [14] proposed a method to estimate the horizon line on images, which has been shown useful for applications such as obstacle or pedestrian detection (e.g., [15]–[17]). Recently, this v-disparity approach has been extended to a $u-v$ disparity concept by Hu and Uchimura [18]. Most of these approaches are based on the use of the Hough transform to compute the straight-line parameters. The underlying assumption is that the road geometry perfectly fits to a plane—or a succession of planes—i.e., a piecewise linear curve [14]. When this assumption does not hold, the straight line extracted in the disparity space does not correspond to the best-fitted plane in the Euclidean space (more details are given in Section III). This problem has recently been addressed by Broggi *et al.* [19]. To the best of our knowledge, [19] is the first work that studies the quality of plane extraction in the v-disparity space with respect to road flatness.

In the Euclidean space, the camera pose estimation problem is generally tackled by fitting a 3-D road model to the acquired 3-D data. To efficiently do this, the amount of available 3-D information is typically reduced by some means. Nedevschi *et al.* [12] propose to consider only 3-D information on edge points and fit a clothoid model of the road surface using a lateral projection of 3-D points. A similar approach is presented by Danescu *et al.* [20] in the context of guardrail and fence detection. The main drawback of these approaches lies on the use of edge points; although it helps reach a real-time performance, it becomes useless in those areas where lanes are not well defined. Another possibility to reduce the processing time is to impose restrictive assumptions to the fitting problem, such as the presence of detectable lane markings on the images (e.g., [9] and [21]).

From a different perspective, we proposed in [22] a RANSAC-based approach to identify 3-D data points belonging to the road and then to fit a plane to them. This approach assumes that the observed road is the dominant structure observed in the images, which is fulfilled in most situations. Exceptions are, for instance, sharp curves, where the road area is missing in front of the vehicle, leaving the off-road area predominant in the distance, or crowded traffic conditions, where the road is highly occluded. However, despite these exceptional cases, a robust performance is achieved in most situations. The major drawback of this technique is the high CPU time required to process the whole set of 3-D data points.

In this paper, we propose a variant of our previous proposal [22], with the aim of reducing its computational requirements and improving its performance capabilities. The proposed technique can be indistinctly used for urban or highway environments since it is not based on a specific visual traffic feature extraction but on 3-D raw data points. As generic 3-D information is used, other modules in ADAS systems, such as collision avoidance algorithms or vehicle/pedestrian detectors, can make use of the same 3-D data, together with the estimated camera pose parameters. The underlying idea of the proposed approach is to develop a robust standalone estimator that independently runs from other applications or hardware systems. In this sense, a commercial stereo pair is used to obtain the 3-D information instead of relying on *ad hoc* technology. In the future, this will allow us to upgrade the current stereo vision sensor without changing the pose-estimation algorithm.

The remainder of this paper is organized as follows: Section II briefly describes the stereo vision system used and formalizes its extrinsic parameters with respect to the observed road. Section III gives a short discussion about disparity and Euclidean representations. Section IV presents the proposed technique, and Section V details the results obtained in urban scenarios (i.e., car's accelerations and flat/uphill/downhill driving), together with validations with ground truth data and a comparative study of its performance with respect to other approaches. Finally, conclusions and further improvements are given in Section VI.

II. Stereo Vision System

To acquire the 3-D information of the scene in front of the host vehicle, a commercial stereo vision system (i.e., Bumblebee from Point Grey [1]) has been used, which consists of two Sony ICX084 Bayer pattern charge-coupled devices with 6-mm focal length lenses. Bumblebee is a precalibrated system that does not require in-field calibration. The baseline of the stereo

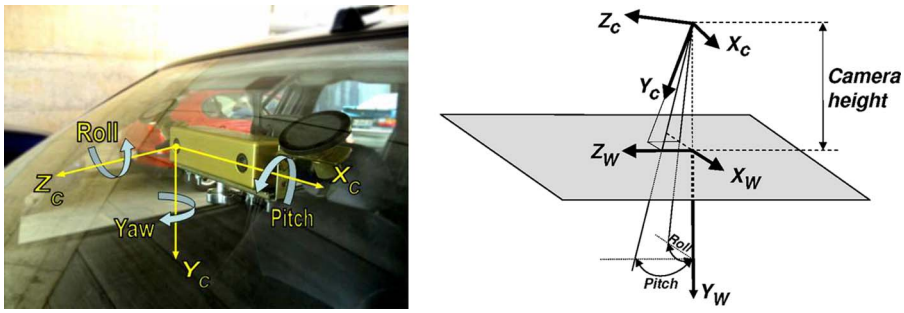

Fig. 1. (Left) Onboard stereo vision sensor with its corresponding coordinate system (the right camera coordinate system is used as reference). (Right) Camera coordinate system (X_C, Y_C, Z_C) and world coordinate system (X_W, Y_W, Z_W).

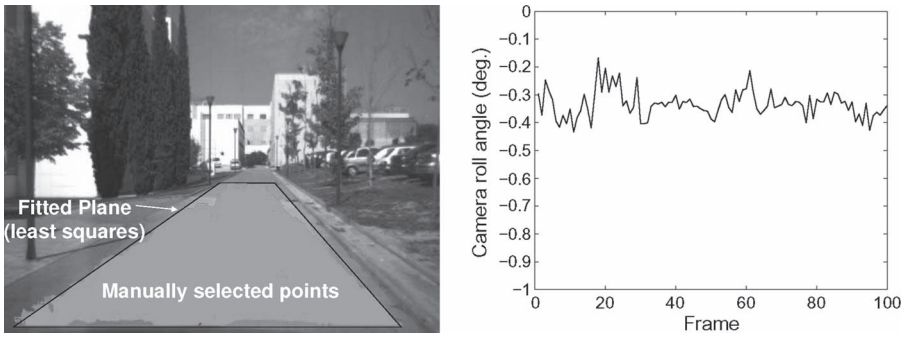

Fig. 2. (Left) Manually selected 3-D points with the fitted plane. (Right) Roll angle variation of a short sequence, showing both a small misalignment introduced during the stereo head mounting stage and a quite-small roll variation (less than 0.5°).

head is 12 cm, and it is connected to the computer by an IEEE-1394 interface. Right and left color images were captured at a resolution of 640 × 480 pixels. Camera control parameters were set to automatic mode to compensate for global changes in the light intensity. After capturing these right and left images, 3-D data were computed by using the provided 3-D reconstruction software.

Extrinsic camera parameters are computed relative to a world coordinate system (X_W, Y_W, Z_W), which is defined for every acquired stereo image, in such a way that the $X_W Z_W$ plane is contained in the current road fitted plane, just under the camera coordinate system (X_C, Y_C, Z_C). Since the Y_W axis contains the origin of the camera coordinate system, and since the yaw angle is fixed to zero in this paper, the six extrinsic parameters[2] $(x_0, y_0, z_0, yaw, roll, pitch)$ that relate the camera coordinate system (X_C, Y_C, Z_C) to the world coordinate system (X_W, Y_W, Z_W) reduce to just three $(0, y_0, 0, 0, roll, pitch)$, which is denoted in the following as (h, Φ, Θ) (i.e., camera height, roll, and pitch). Fig. 1 shows the onboard stereo vision system and its pose with respect to the road plane.

From the (h, Φ, Θ) parameters, the value of Φ will be very close to zero in most situations, since during camera mounting, a specific procedure is followed to ensure an angle at rest within a given range, i.e., ideally zero, and in regular driving conditions, this value scarcely varies. Section IV-A presents a study of the maximum allowed roll misalignment. Although it is beyond the scope of this paper, if the roll angle at rest was higher than the allowed maximum value, a preprocessing stage should be performed to remove this misalignment (e.g., by using all 3-D road points from the first frame, a roll angle value could be computed and then used through the whole video sequence to remove this misalignment). Fig. 2 (right) shows the roll angle variation observed through a short video sequence. Roll angle values were computed by fitting a plane to a subset of 3-D points. This subset corresponds to the road region included in a trapezoid that was manually selected in the image [see Fig. 2 (left)]. It can be appreciated that the procedure followed during the stereo head mounting stage gives an acceptable setup, i.e., only $-0.35°$ of misalignment (see Section IV-A). On the other hand, the roll variation remains quite small (less than 0.5°), motivating us to focus the pose estimation on just (h, Θ), which notably vary on frames due to road imperfections, car accelerations, changes in the road slope (flat/uphill/downhill driving), etc. Fig. 3 illustrates some of these situations.

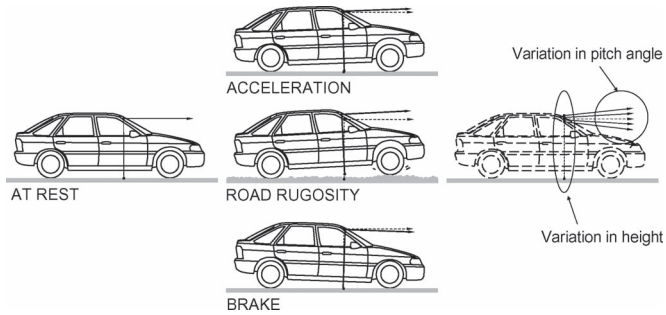

Fig. 3. Typical camera pose variations in an urban scenario.

[2]A 3-D translation and a 3-D rotation.

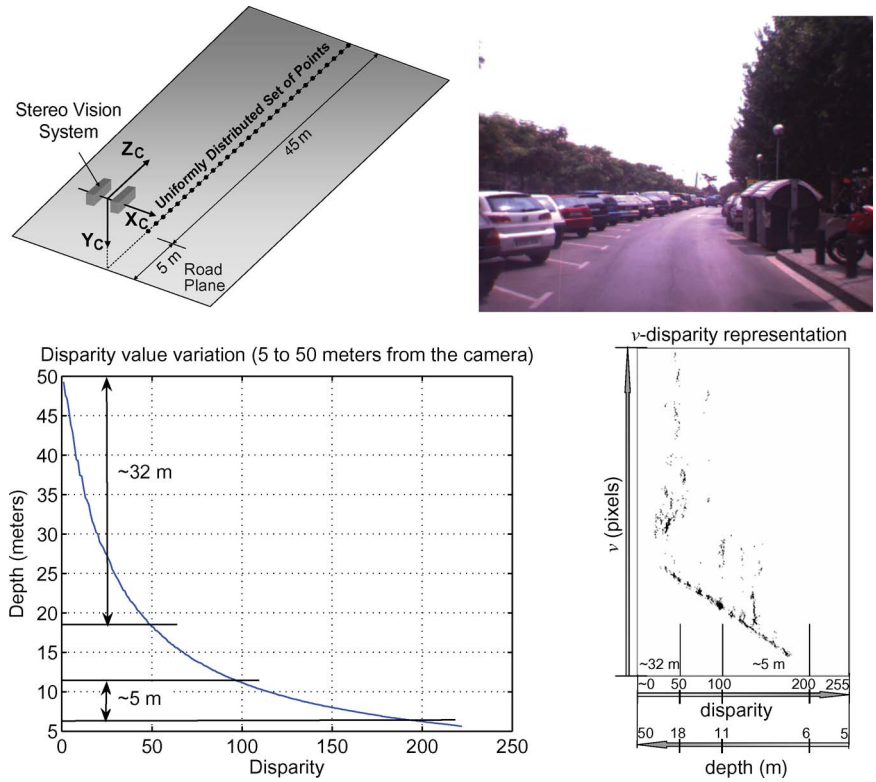

Fig. 4. (Top left) Illustration of the experimental setup. (Top right) Real scene used as test bed. (Bottom left) Disparity value variation for a set of points in the camera optical axis direction, from 5 to 50 m from the camera, in the real scene presented. (Bottom right) v-disparity representation of the real scene presented; points ranging between 5 and 50 m from the camera are depicted. Highlighted regions are only included to emphasize the nonlinear mapping effect.

III. Disparity Space Versus Euclidean Space

Before going into detail about the proposed technique, a brief discussion of working in the 2-D disparity space versus the 3-D Euclidean space is presented.

In this section, we try to emphasize the fact that computing plane parameters in the disparity space can easily be affected by two problems. The first problem is caused by the use of the Hough transform to detect a straight line, which is assumed to be the road plane projection. Without going into detail on the accuracy of straight-line extraction based on the Hough transform (see [23] for an extensive study), it should be noted that the Hough transform allows us to find the largest subset of points that define a straight line, as compared to fitting techniques that allow us find the best straight line for the whole set of points (e.g., least-squares fitting). Although voting and fitting schemes would give the same result when a planar surface is considered (ideal case), differences will appear when nonplanar surfaces are processed. The segment that passes through more points will be obtained by using the Hough transform, which does not necessarily correspond to the most representative point for the whole set of points. On the contrary, when a least-squares fitting approach is used, by previously removing outliers, the plane that minimizes the sum of the squares of the offsets ("the residuals") of the whole set of points is obtained.

In turn, the second problem is due to the nonlinear representation in the disparity space. Let $\mathbf{P}$ be a set of uniformly distributed collinear points belonging to the road plane in the direction of the camera optical axis that is assumed to be parallel to the road plane, i.e., $\{\ldots, P_{i(x_{(i)},y_{(i)},z_{(i)})}, P_{i+1(x_{(i)},y_{(i)},z_{(i)+\Delta})}, \ldots\}$. While their $z_{(i)}$ coordinates linearly increase, their corresponding disparity values $d_{(i)}$ follow the function

$$d_{(i)} = f\frac{b}{z_{(i)}} \tag{1}$$

where b is the baseline distance, f is the focal length, and $z_{(i)}$ is the depth value. Fig. 4 (top left) presents an illustration of a set of points uniformly distributed over a road plane in the camera optical axis direction. Points ranging between 5 and 50 m are considered. Fig. 4 (bottom left) presents the disparity value variation of a real scene [Fig. 4 (top right)] containing a set of points such as those presented in Fig. 4 (top left) and by using our onboard stereo vision sensor. Notice that less than a quarter of disparity values (from 0 up to 50, having a total span higher than 200 values) are used to represent more than 70% of the depth values (distances from 18 up to 50 m). In our setup, a disparity value of 255 corresponds to the nearest point, which is about 5 m. This nonlinear mapping prevents us from equally considering all points. Hence, the use of the Hough transform in the v-disparity space may lead to incorrect results since more attention is paid to the nearest points (almost half of disparity values, i.e., from 100 to 200, are used to represent distances in between 6 and 11 m, which is about 11% of the depth values). Fig. 4 (bottom right) presents the v-disparity representation [17] of a real scene [Fig. 4 (top right)]. The nonlinear effect can also be appreciated in this representation.

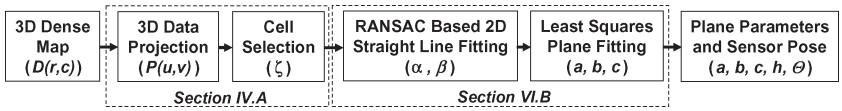

Fig. 5. Algorithm's stages.

Highlighted regions correspond to the aforementioned 70% of the depth values (ranging from 18 to 50 m), which are mapped in less than a quarter of the whole disparity map. It can easily be compared with the 5-m region that covers almost half of the v-disparity representation.

To avoid the aforementioned problems, in this paper, a fitting approach is directly used in the Euclidean space.

IV. Proposed Technique

The proposed technique tackles the camera pose estimation problem using the 3-D Euclidean information provided by a stereo system. The followed strategy basically consists of fitting a plane to 3-D points belonging to the road and then determining the camera pose with respect to that plane from its parameterization. This means that the road region in front of the vehicle (up to 50 m away) is approximated along frames as a piecewise linear curve, since the plane parameters are continuously computed and updated. Road data points are identified by assuming that the road surface is the most predominant geometry in the scene, which holds in most situations.

The proposed approach consists of two stages. Initially, 3-D raw data are mapped onto a 2-D space ($Y_C Z_C$ plane), where a subset of candidate points ζ is selected. The main objective of this first stage is to take advantage of the 2-D structured information before applying more expensive processing algorithms working with 3-D raw data. In a second stage, a RANSAC-based approach is used both to fit a 2-D straight line to a compact representation of those mapped points and to identify inlier points. Original 3-D points, which correspond to those inliers, are finally used to compute the least-squares fitting plane parameters. Camera extrinsic parameters are directly obtained from plane parameters. Fig. 5 presents a flowchart illustrating the algorithm's stages, which are described next.

A. 3-D Data Point Projection and Cell Selection

Let $D(r,c)$ denote the output of the stereo system, which represents a 3-D map with R rows and C columns (the image size). Each array element $D(r,c)$ ($r \in [0,R)$ and $c \in [0,C)$) is a three vector that represents a scene point (x,y,z) in the camera coordinate system. The aim at this first stage is to find a compact subset of points $\zeta \in D(r,c)$ containing most of the road's points. Additionally, the amount of noisy data points[3] should be reduced as much as possible to avoid both very time-consuming processing and erroneous plane fits. To speed up the whole algorithm, most of the processing at this stage is performed over a 2-D space.

[3]Here, "noisy" data points refer to the following: 1) 3-D points belonging to the road and having inaccurate coordinates or 2) 3-D points that do not belong to the road.

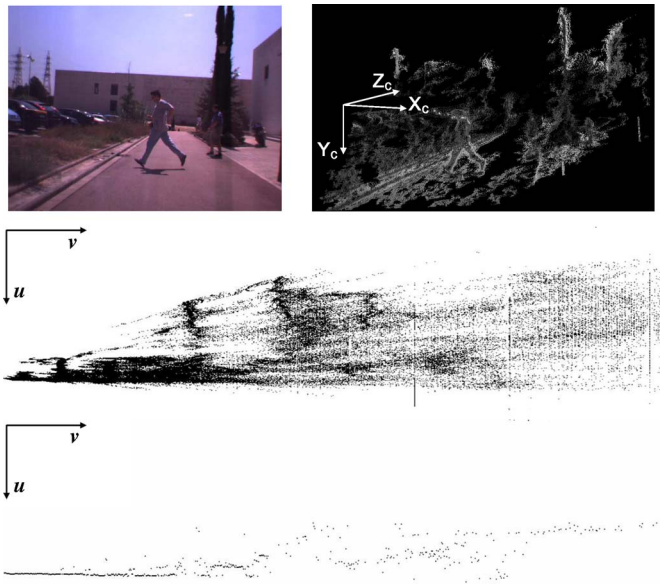

Fig. 6. (Top) (right camera) Single frame together with the 3-D data points computed with the stereo rig—notice that the 3-D image contains a large number of holes due to occlusions and noisy regions. (Middle) $Y_C Z_C$ projection. (Bottom) Cells finally selected to be used during the fitting stage (line and plane fitting).

Since it is assumed that roll angle misalignment remains within a given tolerance, original 3-D data points $D(r,c)$ are mapped onto a 2-D discrete representation $P(u,v)$ in the $Y_C Z_C$ plane, where $u = \lfloor D_y(r,c) \cdot \sigma \rfloor$, and $v = \lfloor D_z(r,c) \cdot \sigma \rfloor$. The parameter σ is a scaling factor that controls the size of the bins according to the current depth map. It is defined as

$$\sigma = \frac{(R+C)/2}{(\Delta X + \Delta Y + \Delta Z)/3} \tag{2}$$

where R and C are the image rows and columns, respectively, and $(\Delta X, \Delta Y, \Delta Z)$ is the working range in a 3-D space (that is, ΔX is the width of the range of x values present in the original data set). Note that the working range depends on the current road scenario; hence, since it could considerably change, the parameter σ is used to scale representations. Every cell of $P(u,v)$ keeps a pointer to the original 3-D data point projected onto that position, as well as a counter with the number of mapped 3-D points. Fig. 6 (middle) shows a 2-D representation obtained after mapping the 3-D cloud presented in Fig. 6 (top right)—every black point in Fig. 6 (middle) represents a cell with at least one mapped 3-D point. In the sequences used to test our proposal, on average, ($\Delta X = 3400$ cm, $\Delta Y = 1200$ cm, $\Delta Z = 5000$ cm), and consequently, every cell of

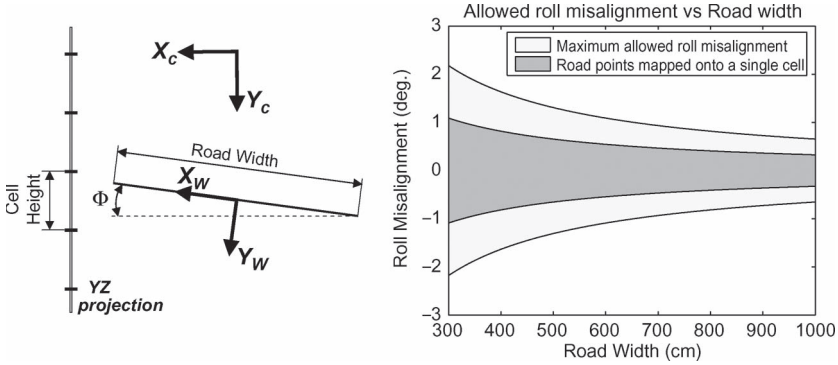

Fig. 7. (Left) Roll angle Φ between camera and world coordinate systems. (Right) Road points mapped assuming a cell height of 5.7 cm.

$P(u, v)$ corresponds to an area of about 5.7 cm $\times$ 5.7 cm in the $Y_C Z_C$ plane.

Points defining the first ζ subset are selected by picking up one cell per column. This selection process is based on the assumption that the road surface is the predominant geometry in the given scene—urban or highway scenarios. Hence, it picks the cell with the largest number of points in each column of the 2-D projection. It avoids the use of a fixed threshold value for the whole 2-D space. This is one of the differences with respect to [22], where a constant threshold value was used in the cell-selection process.

The aforementioned maximum allowed roll angle misalignment is directly related to both the cell size and road width. Fig. 7 (left) presents an illustration where all road points contained in a given line are mapped onto a single cell. In this case, the roll angle can be expressed as $|\Phi| \leq \arcsin(ch/rw)$, where ch represents the height of the cells, and rw is the width of the road. Roll misalignment between camera and world coordinate systems could even take larger values without invalidating the proposed approach. The limit of this situation happens when the road points contained in a line are mapped onto two cells; in such a case, the selection process will not be able to pick the cell corresponding to the predominant geometry since two consecutive cells in a given column contain the same number of mapped points. This maximum allowed roll angle is given by $|\Phi| < \arcsin(2 * ch/rw)$. Fig. 7 (right) shows the range of values of roll misalignment, assuming a cell height of 5.7 cm. One can notice that even in the case where the 3-D points project onto more than two cells, the approach will not break down; in this case, a partial set of points will be used instead of most of the road points.

To reduce the processing time, every selected cell is represented by the 2-D barycenter $(0, (\sum y_i)/n, (\sum z_i)/n)$ of its n mapped points. The set of these barycenters corresponds to a compact representation of the selected subset of points ζ. This data compression step is another difference with [22], where all points mapped onto the selected cells were used for the fitting process. Using both a single point per selected cell and a 2-D representation, a considerable reduction in the CPU time is reached during the road plane fitting stage. Moreover, using barycenters, some smoothing is performed on the original 3-D points. Fig. 6 (bottom) depicts the cells that were finally selected.

B. Road Plane Fitting

The outcome of the previous stage is a compact subset of points ζ, where most of them belong to the road. The road plane fitting stage is split up into two steps. The first step consists of a RANSAC-based [24] fitting process applied over a compact 2-D representation (2-D barycenters). It is intended to remove outlier cells. The second step finally computes the plane parameters by means of least-squares fitting over all 3-D data points contained into inlier cells. Both steps are described next.

Initially, every selected cell is associated with a value $f_{(i)}$ that takes into account the amount of points mapped there, defining a discrete probability function f. These values are computed as

$$f_{(i)} = \frac{n_{(i)}}{N} \tag{3}$$

where $n_{(i)}$ represents the number of points mapped onto the cell i [Fig. 8 (left)], and N represents the total amount of points contained in the selected cells. Recall that we have one cell per column i. This function f will be used in the random sampling stage of the RANSAC algorithm to increase the chance of selecting cells with a large amount of mapped points. This way, the RANSAC algorithm will find the consensus among the whole set of points easier—in other words, it is assumed that a cell containing a reduced set of mapped points could correspond to an outlier. This is managed in the following way: The cumulative distribution function of f, which is denoted by F, is defined in terms of the factors

$$F_{(j)} = \sum_{i=0}^{j} f_{(i)}. \tag{4}$$

If n values of F are randomly sampled using a uniform distribution, the application of the inverse function F^{-1} to those values leads to a set of n points that are adaptively distributed according to $f_{(i)}$. Fig. 8 (right) illustrates this principle.

At the first step, a RANSAC-based approach is applied to find the largest set of cells that fit a 2-D straight line with a user-defined tolerance. Although an automatic threshold could be computed for inlier/outlier detection, following robust

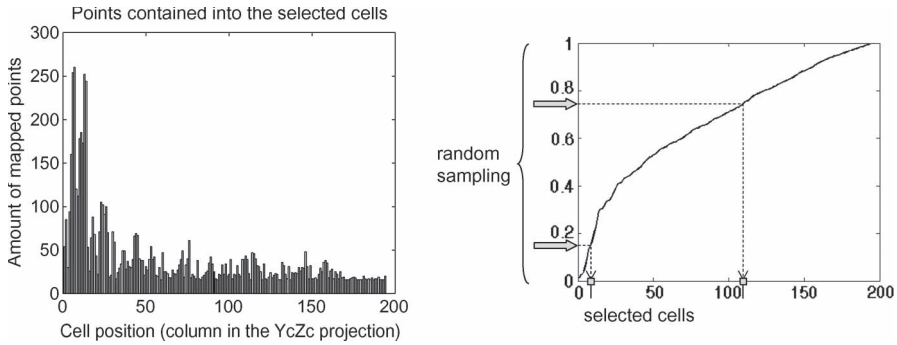

Fig. 8. (Left) Bar diagram showing the amount of points mapped onto the selected cells—recall that only one cell per column is picked up. (Right) Cumulative distribution function computed from the amount of points mapped onto every single cell.

estimation of the standard deviation of residual errors [25], we finally decided to define a fixed value to reduce the CPU time. Note that robust estimation of the standard deviation involves computationally expensive algorithms such as the sorting function. Hence, a predefined threshold value for inlier/outlier detection has been defined (a band of ± 10 cm was enough to take into account both data point accuracy and road planarity). The proposed approach works as follows.

1) Dominant 2-D Straight-Line Parameterization: Repeat the following three steps K times (in our experiments, K was set to 80).

1) Draw a random subsample of two different barycenter points $(P_1, P_2)_k$, where every point is drawn according to the discrete probability function $f_{(i)}$.
2) For this subsample, compute the 2-D straight-line parameters[4] $(\alpha, \beta)_k$.
3) For this solution $(\alpha, \beta)_k$, compute the number of inliers among the entire set of barycenter points from ζ, as aforementioned, using ± 10 cm as a fixed threshold value.

The value of K to be used in this process depends on the percentage of outliers in the data to be processed [7]. It has to be chosen such that the probability P that at least one of the K samples is good is very close to 1 (e.g., $P = 0.99$). A sample (P_1, P_2) is *good* if it consists of two nonoutlier points. Assuming that the whole set of cells may contain up to a fraction ϵ of outliers, the probability that at least one of the K samples is good is given by $P = 1 - [1 - (1 - \epsilon)^2]^K$. Given *a priori* knowledge about the percentage of outliers ϵ, the corresponding K can be computed by

$$K = \frac{\log(1 - P)}{\log(1 - (1 - \epsilon)^2)}. \tag{5}$$

For example, when $P = 0.995$ and $\epsilon = 40\%$, we get $K = 12$ samples. Thus, for the same ϵ and for $K = 80$, we get $P \sim 1$. That is, in practice, it can be assured that the dominant 2-D straight line is estimated.

[4]Notice that the general 2-D straight-line expression $\alpha x + \beta y + \delta = 0$ has been simplified by dividing by $(-\delta)$ since we already know that there is always a distance between the camera and the road plane ($\delta \neq 0$).

2) Road Plane Parameterization: At the second step, plane parameters are computed by using all 3-D data points contained in inlier cells.

1) Choose the computed straight-line parameterization that has the highest number of inliers. Let $(\alpha, \beta)_i$ be this parameterization.
2) Compute $(a, b, c)_i$ plane parameters by using the whole set of 3-D points contained in the cells considered as inliers instead of using the corresponding barycenters. To this end, the least-squares fitting approach [26], which minimizes the square residual error $(1 - ax - by - cz)^2$, is used.
3) In the case in which the number of inliers is smaller than 40% of the total amount of 3-D points contained in ζ, those plane parameters are discarded, and the plane parameters corresponding to the previous frame are used as the correct parameters. In general, this happens when 3-D road data are not correctly recovered since severe occlusion or other external factors appear (data become contaminated with a high percentage of outliers).

Finally, once the road plane has been characterized, the extrinsic camera parameters height h, pitch Θ, and roll Φ angles, referring to the world coordinate system (X_W, Y_W, Z_W), are directly computed. The camera height is given by $h = 1/\sqrt{a^2 + b^2 + c^2}$, and the pitch and roll angles are computed from the current plane orientation, i.e., $\Theta = \arctan(c/b)$ and $\Phi = \arctan(a/b)$.

V. Experimental Results

A. Validation With Manually Annotated Ground Truth

From the estimated camera pose parameters, the pitch angle can be feasibly validated by means of the horizon line position in the image plane (e.g., [27]–[29]). The horizon line position v_i for a given frame i is computed by backprojecting into the image plane a point $P_{(i)}(x, y, z)$ lying over the fitted plane, which is at an infinite distance z from the camera reference frame. From the estimated plane parameters (a, b, c), the $y_{(i)}$ coordinate of a point $P_{(i)}(x, y, z)$ at $z = z_{(i)}$ and $x = 0$ corresponds to $y_{(i)} = (1 - cz_{(i)})/b$. The backprojection of $y_{(i)}$ into the image plane when $z_{(i)} \to \infty$ defines the

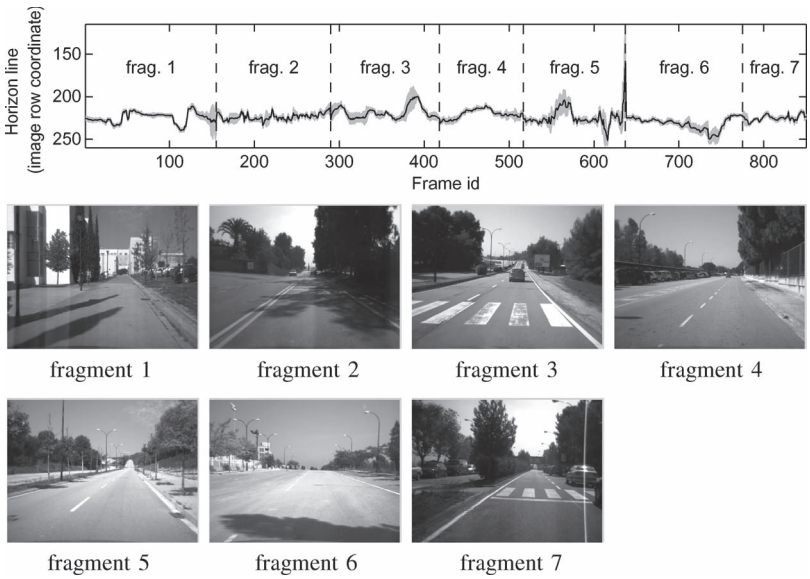

Fig. 9. (Top) Horizon line distribution computed from the annotation of nine users. In general, sudden changes of the horizon are mainly due to vehicle accelerations and the presence of potholes on the road, whereas the more progressive changes reflect variations on the slope of the road. (Bottom) First frames of the seven sequence fragments considered.

row coordinate v_i of the horizon in the image and is computed by

$$v_{(i)} = v_{(0)} + f\frac{y_{(i)}}{z_{(i)}} = v_{(0)} + \frac{f}{z_{(i)}b} - f\frac{c}{b} \tag{6}$$

where f denotes the focal length in pixels, $v_{(0)}$ represents the vertical coordinate of the camera principal point, and $z_{(i)}$ is the depth value of $P_{(i)}$. Since $(z_{(i)} \to \infty)$, the row coordinate of the horizon line in the image is finally computed as $v_{(i)} = v_{(0)} - f(c/b)$.

The performance of the proposed approach has been quantitatively validated as follows. First, a video sequence of an urban scenario was acquired, from which seven different fragments were extracted. These fragments were generated by cutting and subsampling the original sequence into smaller sequences. Each fragment only contains frames where, at least apparently, a human can annotate the location of the horizon line with reasonable confidence. The cutting/subsampling process was intended to cover different scenarios, avoiding redundant information given by consecutive frames. As a result, a group of 850 testing frames was collected and then manually annotated by nine different users. Users were asked to locate a vanishing point in every frame, taking advantage of parallel structures observed on the road region neighboring the vehicle holding the camera (mainly lane borders and lane markings). Therefore, the annotated vanishing point corresponds to a plane fitting the road surface observed in front of the host vehicle. Typically, this surface corresponds to the ground plane on which the vehicle is leaning. Notice, however, that this is not necessarily the only possible case. Section V-D analyzes in detail one of such singular cases. From the collected annotations, the Gaussian distribution of the most likely horizon line location at each frame was determined. Fig. 9 (top) shows the computed horizon line distribution and the first frame of each testing fragment. The plot shows the mean of the user annotations at each frame, depicting with a gray region its 95% confidence interval, reflecting the variance in the annotations provided by the different users. Frames with a higher variance are the frames that the users found more ambiguous to annotate.

The 850 annotated testing frames were processed by the proposed technique, computing at each frame the position of the horizon line corresponding to the estimated camera pose. Every frame has a resolution of 640 $\times$ 480 pixels, and the camera focal length is 824 pixels. A 3.2-GHz Pentium IV PC with a nonoptimized C++ code has been used. The proposed algorithm took, on average, 78 ms/frame, including both 3-D point computation and onboard pose estimation. Finally, the performance of the proposed approach has been quantified by comparing the estimated horizon line with its most likely location according to the user annotation. Fig. 10 shows the disparity of the proposed method against the mean of the horizon line annotation. In 72% of the frames, the horizon line computed with the proposed technique lies inside the bounds of the 95% confidence interval of the ground truth horizon line annotation. By analyzing the statistics of the localization errors of the proposed method, it is observed that in nearly half of the sequence, the horizon line is estimated with an error smaller than or equal to one pixel, and in 90% of the frames, the error is smaller than or equal to four pixels, which is a very remarkable performance.

The proposed method has an error larger than 11 pixels in only five testing frames. By analyzing these cases, it is found that they correspond to frames where the variance of the user annotations is large, thus reflecting the difficulty to determine the horizon line location on them without ambiguity. Fig. 11 shows the two situations that have led to these larger errors: 1) a frame that contains a cross between two roads with different slopes (frame 155) and 2) frames where the observed road is not planar, showing notable changes of slope in a short distance (frames 632–635).

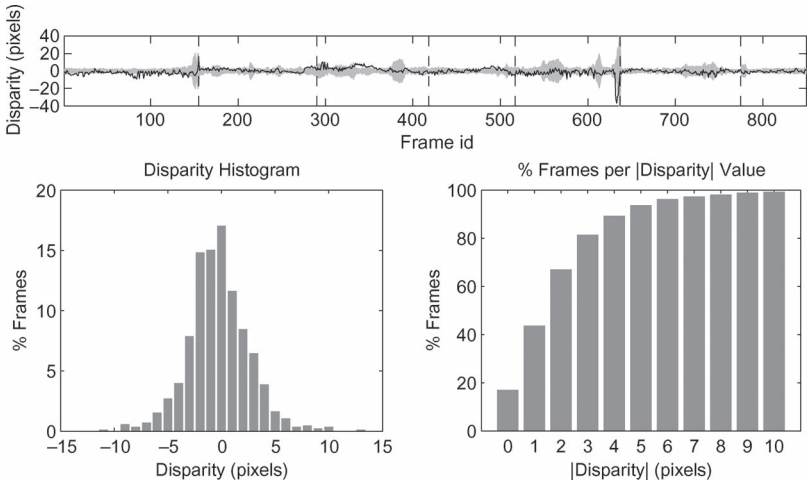

Fig. 10. (Top) Horizon line disparity between the estimated value and the mean of manually annotated values. (Bottom left) Histogram of the disparity. (Bottom right) Percentage of frames where a disparity is smaller than or equal to a given value.

Fig. 11. Frames corresponding to maximum errors between the (brighter image zone highlights the 95% confidence interval of the annotation, whereas dotted-squared lines correspond to the mean value) manually annotated horizon line and (dashed line) automatically computed horizon line. (Left) Cross between roads with different slopes. (Right) Road containing a sharp uphill slope. Notice from the confidence interval that even human evaluation cannot be so accurate in these frames.

B. Detailed Performance Analysis

This section analyzes the performance of the proposed method in two fragments of the testing sequence, which show variations of the horizon line location mainly due to changes in vehicle velocity (Fig. 12) and to changes in road geometry (Fig. 13), respectively. These fragments are also of particular interest because they have a small percentage of horizon line estimations inside the confidence region of the ground truth (i.e., they correspond to the worst results), and the cause of this is analyzed. In both figures, two graphics are plotted: one showing the estimated vehicle velocity at each frame (the translational velocity of the camera (V_x, V_y, V_z) computed from the differences of 3-D points) and another one specifying the ground truth confidence region of the annotated horizon line and the horizon location computed with the proposed technique. The information plotted in this second graphic is represented in some selected frames to provide a clearer understanding of the method performance.

The sequence fragment in Fig. 12 shows a flat road, where the host vehicle performs quite sudden changes in its velocity. It is clearly observed that the image row coordinate of the horizon line notably increases when the vehicle accelerates [selected frames in ranges a–c and h–i], decreasing more abruptly in the decelerations [selected frames in ranges c–d and i–j] due to the behavior of the vehicle suspension system. The fragment also shows horizon line variations due to a change in the road pavement (selected frame e) and to the crossing of roads with different slopes (selected frame k). The major disparity between the algorithm output and the ground truth annotation is detailed on the selected frame g. In this case, the 3-D road points provided by the acquisition system are notably sparse and noisy,[5] leading a less-precise plane characterization. However, the disparity on the horizon line location is smaller than ten pixels.

The sequence fragment in Fig. 13 mainly shows the effect that the road geometry has on the estimated horizon line. Two different interesting situations can be observed. In the first part of the sequence (up to the selected frame h), the road is apparently flat, and the horizon line changes according to vehicle velocity. However, the estimated horizon line has a clear offset with respect to the annotated ground truth because the observed road is, in fact, composed of two surfaces with different heights, due to the reasphaltation of the left part of the lane (the step between the two road parts is clearly visible in the selected frame a). This makes the manual estimation of the horizon line ambiguous. Given that the road parallel structures observed on images lie on different surfaces, a systematic error between the user annotation and the algorithm estimation is made. However, in spite of this conflictive situation, in the worst case, the disparity between the ground truth horizon and the estimated horizon does not exceed ten pixels. In the last part of the sequence (from the selected frame h until the end), the horizon line variation is caused by a change of road slope. First, in the range between the selected frames h–j, the host vehicle lies on a plane, whereas the observed road ahead corresponds to a different plane with a higher slope. As a consequence, the row coordinate of the horizon line decreases. Once the host vehicle enters on this slopped part of the road, the horizon line recovers its position, given that the wheels lie on the same plane as the observed road.

[5]Due to the road homogeneity observed in this frame, very few road points can be reliably matched in the images of the acquired stereo pair.

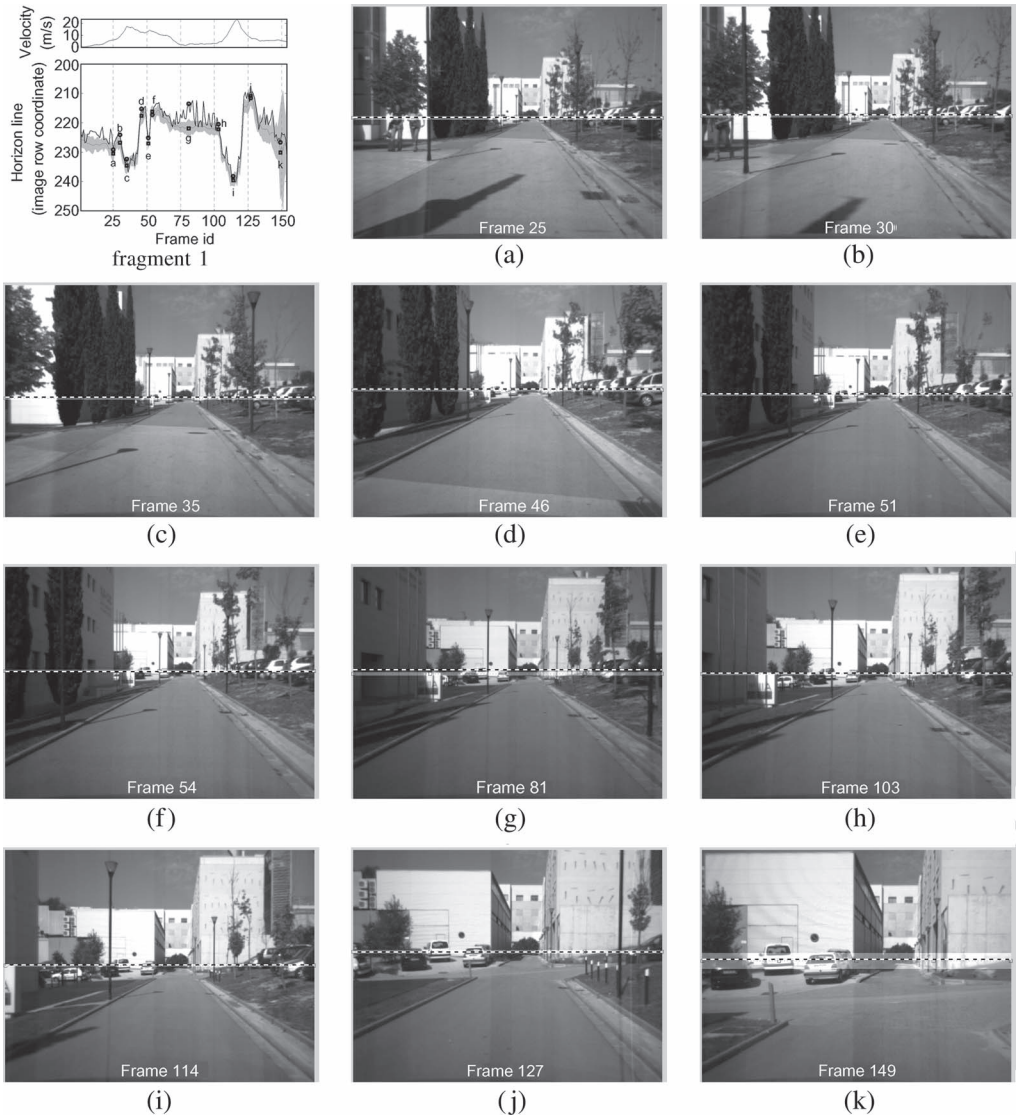

Fig. 12. Horizon line variation along the first fragment of the testing sequence. (Top left) Plot displaying for each frame in the fragment the (topmost plot) estimated velocity of the vehicle hosting the camera and the (gray-shaded area) 95% confidence region of the annotated horizon line position, together with the (solid line) horizon line estimated by the proposed method. Labeled squares and circles highlight the annotated and estimated horizon line positions in some selected frames, respectively. These frames are displayed next to illustrate the variation of the horizon line position along time, showing the (a dotted-squared line in a brighter image region) annotated horizon line confidence region and (dashed line) estimation provided by the proposed method.

C. Comparative Performance Analysis

A comparative study of the performance of the proposed method with respect to two previous approaches (one stereo and one monocular) has been done. First, the proposed approach has been compared with our previous stereo-based proposal [22]. In this case, the algorithm also fits a plane in the Euclidean space, but it uses almost all 3-D road data points instead of a compact representation, such as in the current presented technique. Fig. 14 graphically compares the performance of both approaches. The disparity histogram shows that the disparity of the current proposal is more densely distributed around zero, with a smaller variance. Consequently, the percentage of frames where the horizon is estimated with a disparity smaller than or equal to a given value is clearly bigger with the current proposal. Analyzing the performance of both algorithms frame by frame, it is observed that the current proposal outperforms [22] in 64% of the testing frames, requiring less than a quarter of the processing time. We claim that this better performance is due to two reasons. On the one hand, the cell-selection process (the first stage of current proposal) picks up cells through the whole road surface by avoiding the use of a fixed threshold. On the other hand, also during this first stage, the use of cell's barycenter helps filter noisy data. Improvements on the CPU time are mainly due to using barycenters during the sampling process instead of using the whole set of points. The probability function used during the second stage also helps reduce the CPU time by achieving a faster consensus among the whole set of points, which speeds up the RANSAC-based fitting process.

The described proposal has also been compared with the histogram-based approach proposed by Bertozzi *et al.* [6]. This is a monocular-based method that, given the horizon line position at the first frame of a sequence (in the current experiments, the annotated ground truth has been used), first

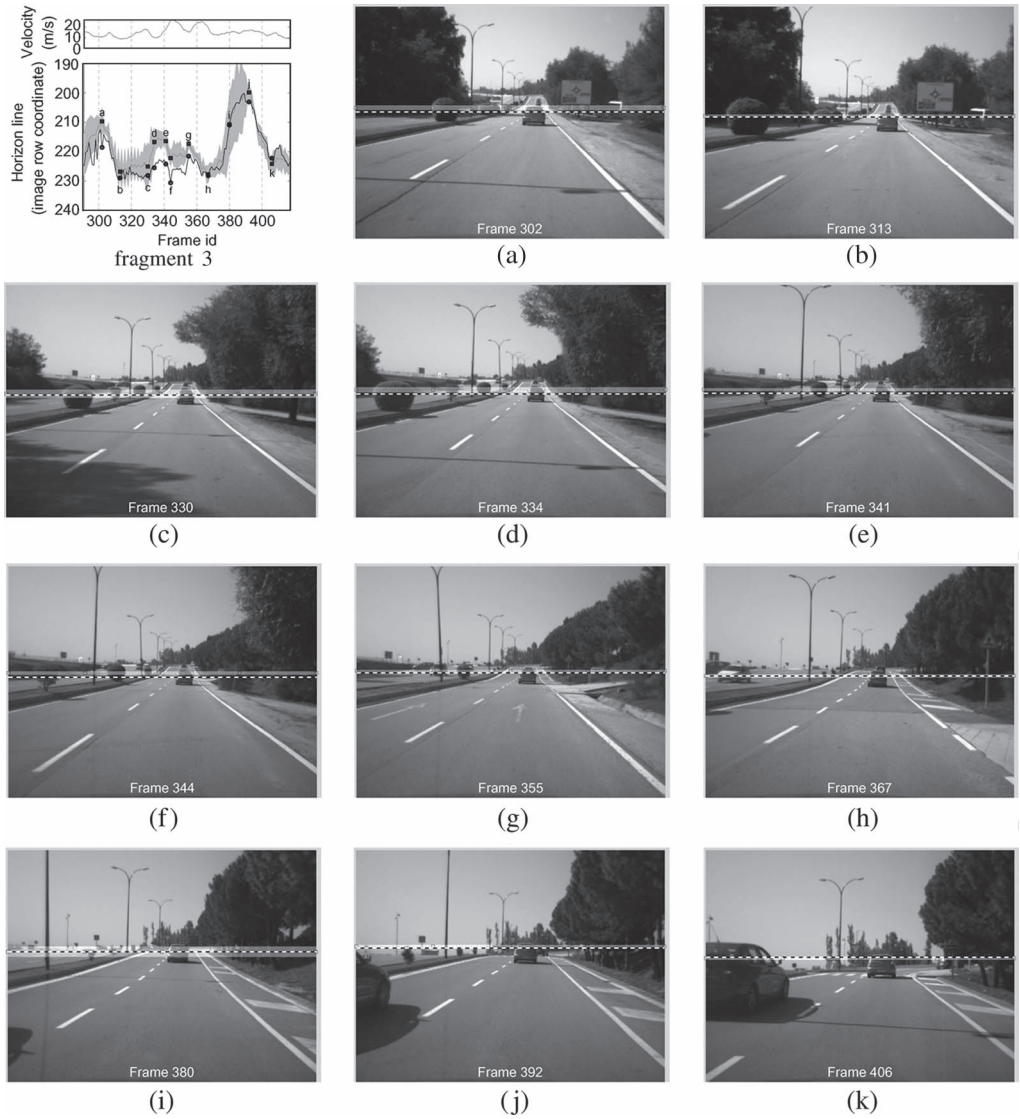

Fig. 13. Horizon line variation along the third fragment of the testing sequence. (Top left) Plot displaying for each frame in the fragment the (topmost plot) estimated velocity of the vehicle hosting the camera and the (gray-shaded area) 95% confidence region of the annotated horizon line position, together with the (solid line) horizon line estimated by the proposed method. Labeled squares and circles highlight the annotated and estimated horizon line positions in some selected frames, respectively. These frames are displayed next to illustrate the variation of the horizon line position along time, showing the (a dotted-squared line in a brighter image region) annotated horizon line confidence region and (dashed line) estimation provided by the proposed method.

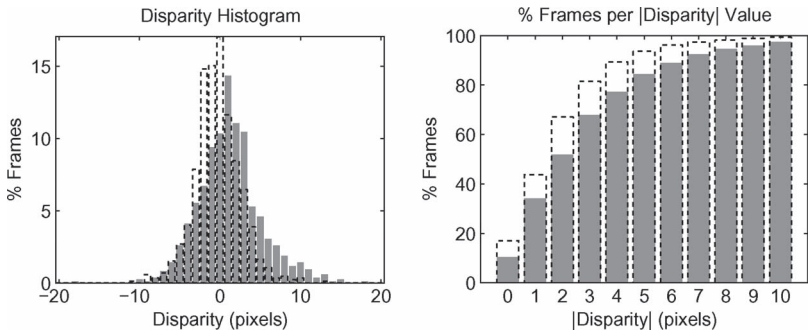

Fig. 14. Performance comparison of the (dotted lines) proposed method and (gray bars) proposal in [22]. (Left) Histogram of the disparity. (Right) Percentage of frames where a disparity is smaller than or equal to a given value.

constructs a 1-D reference pattern from the y-projection of the image edges around this position. This pattern conforms to a row-wise edge histogram, which is matched to the y-projection of the edges in the next frame to determine the horizon line position. The pattern is then updated using the edge information of the processed frame to adapt to the progressive change of the observed scene. Fig. 15 graphically compares the performance of both approaches.

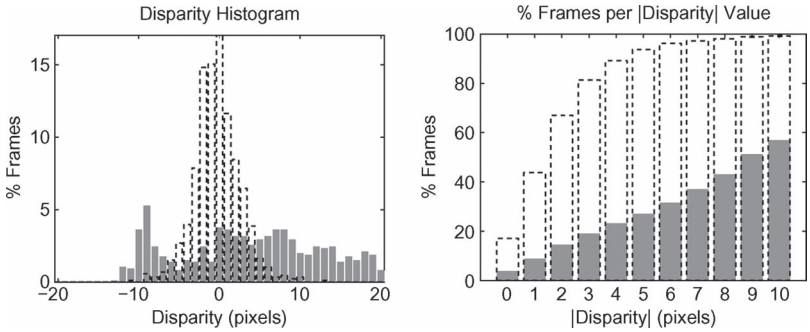

Fig. 15. Performance comparison of the (dotted lines) proposed method and (gray bars) proposal in [6]. (Left) Histogram of the disparity. (Right) Percentage of frames where a disparity is smaller than or equal to a given value.

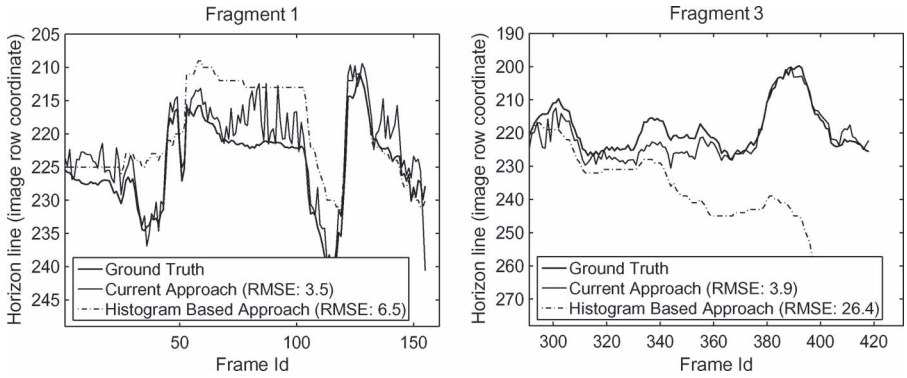

Fig. 16. Performance comparison of the (thin line) proposed method and (dot-dashed line) method in [6] with respect to (thick line) ground truth in fragments 1 and 3 of the testing sequence.

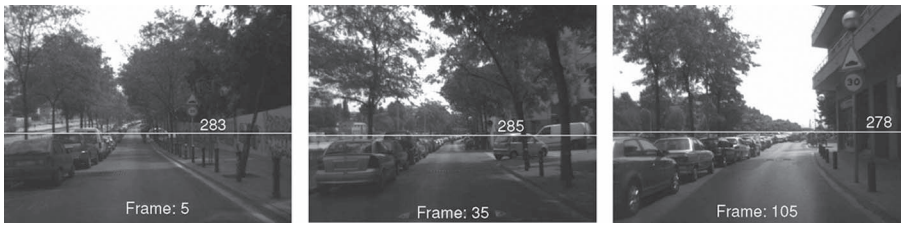

Fig. 17. (White line) Estimated horizon line in frames acquired in a narrow urban road. The robustness of the proposed technique can be qualitatively appreciated in spite of the car-cluttered street.

Results clearly show the superiority of the proposed approach with respect to the histogram-based approach, which performs very poorly. Notice that for about 40% of the processed frames, the error in locating the horizon is bigger than ten pixels. This is due to the need to update the histogram reference pattern along frames. When this updating is inaccurate for some reason, an incorrect location of the horizon line is obtained and then propagated along the rest of the sequence, thus causing a drift in its position. Fig. 16 details the performance of the histogram-based approach in fragments 1 and 3 of the testing sequence. Although fragment 1 seems to be a sequence that is favorable to this method, a remarkable error is produced. Fragment 3 is even a clearer example of the unreliability of the histogram-based approach. The presence of moving vehicles in the scene, as well as the presence of a big traffic signal at the medium distance, shifts the reference pattern with respect to the real horizon line location, and this error is accumulated in the rest of the sequence.

D. Performance on Singular Situations

Once the performance of the proposed method has been quantitatively evaluated, some interesting situations are qualitatively evaluated. First, the performance in narrow urban scenarios has been checked, showing a significantly robust performance. Fig. 17 shows the robust location of the horizon line in frames where the image regions corresponding to the road are notably smaller than the image regions present in the previous testing frames.

The effect that road irregularities have on the estimated camera pitch and height is shown in Fig. 18. The method performance is analyzed in a short sequence captured at nearly constant speed in a road segment presenting a rumble strip and a speed bump. The frames of the sequence where the camera pose is altered by these artifacts have been inferred from the estimated vehicle velocity. From both elements, only the crossing of the speed bump remarkably alters the pose of the acquisition system (frames 10–15). A pitch oscillation

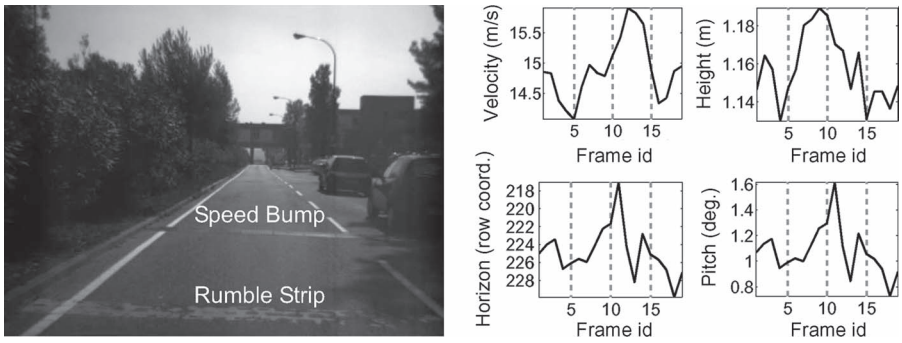

Fig. 18. Method performance in the presence of a rumble strip and a speed bump. (Left) First frame of the sequence. (Right) Evolution along frames of the estimated vehicle velocity, camera height, horizon line position, and camera pitch.

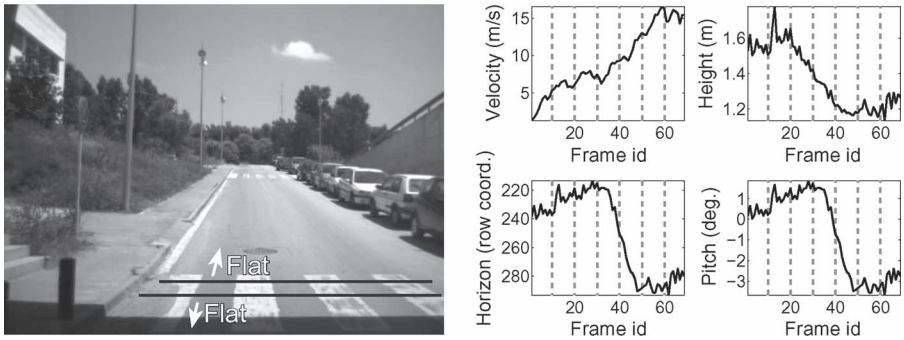

Fig. 19. Method performance in the transition between two ground planes of different slopes. (Left) First frame of the sequence. (Right) Evolution along frames of the vehicle velocity, camera height, horizon line position, and camera pitch.

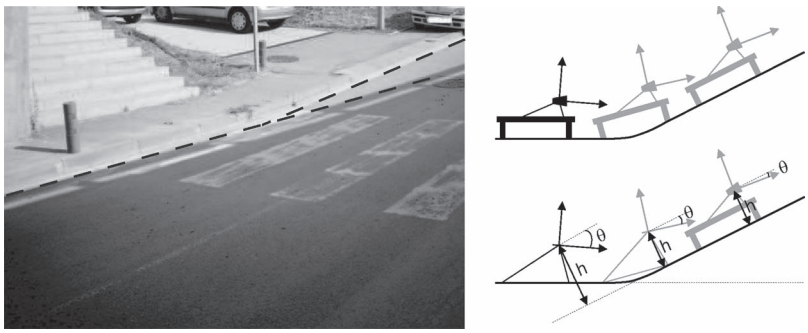

Fig. 20. (Left) Details on the connection between the two road surfaces. (Right) Sketch illustrating the variation of the onboard camera pose at a slope discontinuity by neglecting car dynamics. In a vehicle, the response of the suspension system to the road slope and to vehicle accelerations provokes additional deviations of the pitch angle.

is observed, which is caused by the response of the vehicle suspension system to the speed bump.

Finally, it is interesting to analyze how a change of road slope alters the estimated camera pitch and height. Fig. 19 displays the results obtained in a sequence where, starting from rest, a vehicle progresses from a planar to a hill-climbing road. In the first 30 frames, the vehicle moves on the first flat road segment, but most of the surface on the acquired images corresponds to the hill-climbing road ahead. In this situation, the pitch and height are relatively estimated to a world coordinate system coherently placed to the observed hill-climbing road. Fig. 20 shows a typical sketch illustrating the variation of the onboard camera pose at a slope discontinuity by neglecting car dynamics. Due to this, notice that the estimated height value is significantly distant from its *neutral* value (which is approximately 1.2 m). Between frames 30 and 50, the vehicle progresses in the transition between the two road surfaces, which causes a remarkable variation of the estimated pitch and height. Once the vehicle is completely on the hill-climbing the road (from frame 50 until the end), the estimated camera height approximately recovers its neutral value. The pitch variation over time mimics the height behavior, whereas the specific pitch value at each frame reflects the effect of vehicle acceleration (quite constant along the sequence), as well as the effect of the suspension system when the vehicle is on the uphill road.

VI. Conclusion

A technique for an efficient pose estimation of an onboard camera has been presented. The estimated values refer the

camera coordinate system to a world coordinate system lying in a plane-fitting car's neighborhood, which is supposed to be the road surface. Although the roll angle is assumed to be kept constant, the proposed approach can handle small roll variations. The input data are a set of 3-D points obtained with an onboard stereo camera. After an initial mapping and filtering process, a reduced set of 3-D points is chosen as road candidate points. Then, a two-stage fitting approach computes road plane parameters and, accordingly, camera extrinsic parameters: 1) RANSAC-based 2-D straight-line fitting and 2) least-squares plane fitting. The proposed technique can fit very well to different road geometries since plane parameters are continuously computed and updated in the Euclidean space. A good performance has been shown in several scenarios—uphill, downhill, and flat roads. Furthermore, critical situations such as a car's accelerations or speed bumps were also considered and validated with manually estimated ground truth by nine different experts. Although it has been tested on urban environments, it could also be useful on highway scenarios. A considerable reduction in the CPU processing time was achieved by both working with a reduced set of points and selecting reliable points according to a continuously updated probability distribution function. The latter approach drives to a faster convergence during the RANSAC fitting stage. The proposed technique is already being used on an appearance-based pedestrian-detection algorithm to speed up the searching process [30].

Further work will address the use of other geometries for fitting road points; for instance, the distribution of the error between road points and the currently computed surface will be used to select the best surface primitive for the next fitting stage (e.g., plane, piecewise planar approximation, and quadratic surface). In addition, the use of Kalman filtering will be explored to both speed up and add temporal information to the process.

ACKNOWLEDGMENT

The authors would like to thank the three anonymous referees for their valuable and constructive comments, as well as the CVC-ADAS team for the manually annotated ground truth.

REFERENCES

[1] [Online]. Available: http://www.ptgrey.com
[2] [Online]. Available: http://www.videredesign.com
[3] D. Ponsa, A. López, F. Lumbreras, J. Serrat, and T. Graf, "3D vehicle sensor based on monocular vision," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Vienna, Austria, Sep. 2005, pp. 1096–1101.
[4] P. Coulombeau and C. Laurgeau, "Vehicle yaw, pitch, roll and 3D lane shape recovery by vision," in *Proc. IEEE Intell. Veh. Symp.*, Versailles, France, Jun. 2002, pp. 619–625.
[5] Y. Liang, H. Tyan, H. Liao, and S. Chen, "Stabilizing image sequences taken by the camcorder mounted on a moving vehicle," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Shangai, China, Oct. 2003, pp. 90–95.
[6] M. Bertozzi, A. Broggi, M. Carletti, A. Fascioli, T. Graf, P. Grisleri, and M. Meinecke, "IR pedestrian detection for advanced driver assistance systems," in *Proc. 25th Pattern Recog. Symp.*, Magdeburg, Germany, Sep. 2003, pp. 582–590.
[7] O. Faugeras and Q. Luong, *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. Cambridge, MA: MIT Press, 2001.
[8] N. Hautière, R. Labayrade, and D. Aubert, "Real-time disparity contrast combination for onboard estimation of the visibility distance," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 2, pp. 201–212, Jun. 2006.
[9] S. Nedevschi, C. Vancea, T. Marita, and T. Graf, "Online extrinsic parameters calibration for stereovision systems used in far-range detection vehicle applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 4, pp. 651–660, Dec. 2007.
[10] W. van der Mark and D. Gavrila, "Real-time dense stereo for intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 38–50, Mar. 2006.
[11] S. Krotosky and M. Trivedi, "On color-, infrared-, and multimodal-stereo approaches to pedestrian detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 4, pp. 619–629, Dec. 2007.
[12] S. Nedevschi, R. Danescu, D. Frentiu, T. Graf, and R. Schmidt, "High accuracy stereovision approach for obstacle detection on non-planar roads," in *Proc. IEEE Intell. Eng. Syst.*, Cluj Napoca, Romania, Sep. 2004, pp. 211–216.
[13] N. Simond, "Reconstruction of the road plane with an embedded stereo-rig in urban environments," in *Proc. IEEE Intell. Veh. Symp.*, Tokyo, Japan, Jun. 2006, pp. 70–75.
[14] R. Labayrade, D. Aubert, and J. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through 'V-disparity' representation," in *Proc. IEEE Intell. Veh. Symp.*, Versailles, France, Jun. 2002, pp. 646–651.
[15] M. Bertozzi, E. Binelli, A. Broggi, and M. Del Rose, "Stereo vision-based approaches for pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, San Diego, CA, Jun. 2005, pp. 16–22.
[16] A. Broggi, A. Fascioli, I. Fedriga, A. Tibaldi, and M. Del Rose, "Stereo-based preprocessing for human shape localization in unstructured environments," in *Proc. IEEE Intell. Veh. Symp.*, Columbus, OH, Jun. 2003, pp. 410–415.
[17] R. Labayrade and D. Aubert, "A single framework for vehicle roll, pitch, yaw estimation and obstacles detection by stereovision," in *Proc. IEEE Intell. Veh. Symp.*, Columbus, OH, Jun. 2003, pp. 31–36.
[18] Z. Hu and K. Uchimura, "U-V-disparity: An efficient algorithm for stereovision based scene analysis," in *Proc. IEEE Intell. Veh. Symp.*, Las Vegas, NV, Jun. 2005, pp. 48–54.
[19] A. Broggi, C. Caraffi, P. Porta, and P. Zani, "The single frame stereo vision system for reliable obstacle detection used during the 2005 DARPA grand challenge on TerraMax," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Toronto, ON, Canada, Sep. 2006, pp. 745–752.
[20] R. Danescu, S. Sobol, S. Nedevschi, and T. Graf, "Stereovision-based side lane and guardrail detection," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Toronto, ON, Canada, Sep. 2006, pp. 1156–1161.
[21] J. Collado, C. Hilario, A. de la Escalera, and M. Armingol, "Self-calibration of an on-board stereo-vision system for driver assistance systems," in *Proc. IEEE Intell. Veh. Symp.*, Tokyo, Japan, Jun. 2006, pp. 156–162.
[22] A. Sappa, D. Gerónimo, F. Dornaika, and A. López, "On-board camera extrinsic parameter estimation," *Electron. Lett.*, vol. 42, no. 13, pp. 745–747, Jun. 2006.
[23] V. Shapiro, "Accuracy of the straight line Hough transform: The non-voting approach," *Comput. Vis. Image Underst.*, vol. 103, no. 1, pp. 1–88, Jul. 2006.
[24] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Graph. Image Process.*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
[25] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*. New York: Wiley, 1987.
[26] C. Wang, H. Tanahashi, H. Hirayu, Y. Niwa, and K. Yamamoto, "Comparison of local plane fitting methods for range data," in *Proc. IEEE Comput. Vis. Pattern Recog.*, Kauai, HI, Dec. 2001, pp. 663–669.
[27] C. Zhaoxue and S. Pengfei, "Efficient method for camera calibration in traffic scenes," *Electron. Lett.*, vol. 40, no. 6, pp. 368–369, Mar. 2004.
[28] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, Washington, DC, Jun. 2004, pp. 470–477.
[29] C. Rasmussen, "Texture-based vanishing point voting for road shape estimation," in *Proc. Brit. Mach. Vis. Conf.*, London, U.K., Sep. 2004.
[30] D. Gerónimo, A. Sappa, A. López, and D. Ponsa, "Adaptive image sampling and windows classification for on-board pedestrian detection," in *Proc. Int. Conf. Comput. Vis. Syst.*, Bielefeld, Germany, 2007.

Angel Domingo Sappa (S'93–M'99) received the electromechanical engineering degree from the National University of La Pampa, General Pico, Argentina, in 1995 and the Ph.D. degree in industrial engineering from the Polytechnic University of Catalonia, Barcelona, Spain, in 1999.

In 2003, after holding research positions in France, the U.K., and Greece, he joined the Computer Vision Center, Universitat Autònoma de Barcelona, where he is currently a member of the Advanced Driver Assistance Systems Group. His research interests span a broad spectrum within 2-D and 3-D image processing. His current research focuses on stereo image processing and analysis, 3-D modeling, and model-based segmentation.

Fadi Dornaika received the B.S. degree in electrical engineering from Lebanese University, Tripoli, Lebanon, in 1990 and the M.S. and Ph.D. degrees in signal, image, and speech processing from the Institut National Polytechnique de Grenoble, Grenoble, France, in 1992 and 1995, respectively.

He has worked at several research institutes, including INRIA Rhône-Alpes, Saint Ismier, France; the Chinese University of Hong Kong, Hong Kong; Linköping University, Linköping, Sweden; and the Computer Vision Center, Universitat Autònoma de Barcelona, Barcelona, Spain. He is currently with the French National Geographical Institute, Saint-Mandé, France. He has published about 90 papers in the field of computer vision and pattern recognition. His research concerns geometrical and statistical modeling, with focus on 3-D object pose, real-time visual servoing, calibration of visual sensors, cooperative stereo motion, image registration, facial gesture tracking, and facial expression recognition.

Daniel Ponsa received the B.Sc. degree in computer science, the M.Sc. degree in computer vision, and the Ph.D. degree from the Universitat Autònoma de Barcelona (UAB), Barcelona, Spain, in 1996, 1998, and 2007, respectively.

From 1996 to 2003, he was a Teaching Assistant with the Computer Science Department, UAB, where he participated in different industrial projects within the Computer Vision Center (CVC). He is currently a full-time Researcher with the CVC research group on advanced driver-assistance systems by computer vision. His research interests include tracking, pattern recognition, machine learning, and feature selection.

David Gerónimo received the B.Sc. degree in computer science from the Universitat Autònoma de Barcelona (UAB), Barcelona, Spain, in 2004 and the M.Sc. degree from the Computer Vision Center, UAB, Barcelona, in 2006. He is currently working toward the Ph.D. degree on the project "Pedestrian Detection in Advanced Driver Assistance Systems," under the Research Training Program (FPI) of the Spanish Ministry of Science and Technology, at the Computer Vision Center, UAB.

His research interests include feature selection, machine learning, and object recognition.

Antonio López received the B.Sc. degree in computer science from the Polytechnic University of Catalonia, Barcelona, Spain, in 1992 and the M.Sc. degree in image processing and artificial intelligence and the Ph.D. degree from the Universitat Autònoma de Barcelona (UAB) in 1994 and 2000, respectively.

Since 1992, he has been giving lectures at the Computer Science Department, UAB, where he is currently an Associate Professor. In 1996, he participated in the foundation of the Computer Vision Center, UAB, where he has held different institutional responsibilities and is presently responsible for the research group on advanced driver-assistance systems by computer vision. He has been responsible for public and private projects. He has coauthored more than 50 papers, all in the field of computer vision.