Stereo Vision Camera Pose Estimation for On-Board Applications

Sappa A.[*], Gerónimo D.[*], Dornaika F.[*] , and López A.[*] [‡]
[*]Computer Vision Center and [‡]Autonomous University of Barcelona
08193 Bellaterra, Barcelona,
Spain

1. Introduction

Traffic accidents have become an important cause of fatality in modern countries. For instance, in 2002, motor vehicle accidents represented the half of non-natural death in the United States (National Center of Health Statistics, 2002); while in 2003 there were reported almost 150,000 injured and 7,000 killed in pedestrian accidents only in the European Union (United Nations Economic Commission for Europe, 2005). In order to improve safety, the industry has progressively developed different elements of increasing complexity and performance: from turn signals and seat belts to Anti-lock Braking Systems (ABS) and internal Airbags. Recently, research has moved towards even more intelligent on-board systems that aim to anticipate and prevent the accidents, or at least, minimize their effects when unavoidable. They are referred as Advanced Driver Assistance Systems (ADAS), in the sense that they assist the driver to take decisions, provide warnings in dangerous driving situations, and even at taking automatic evasive actions in extreme cases.

One of the most prominent components of ADAS are the vision systems (monocular or stereo), which capture in a single snapshot all the surrounding information. Although monocular vision systems allow higher acquisition/processing rates, the use of on-board stereo vision heads is gaining popularity in ADAS applications. Stereo rigs are able to provide 3D information useful for facing up problems that can not be tackled with monocular systems (e.g., reliable distance estimation). Furthermore, the current technology is producing more and more inexpensive and compact stereo vision systems that let us think on a promising future.

Accurate and real time camera pose estimation is one of the common difficulties of on-board vision systems. Applications such as obstacle avoidance, pedestrian detection, or traffic signal recognition, could both speed up the whole process and make use of additional information by a precise estimation of the current camera extrinsic parameters, related to the

This work has been partially supported by the Spanish Ministry of Education and Science under project TRA2004-06702/AUT. First and third authors were supported by The Ramón y Cajal Program. Second author was supported by Spanish Ministry of Education and Science grant BES-2005-8864.

Source: Scene Reconstruction, Pose Estimation and Tracking, Book edited by: Rustam Stolkin,
ISBN 978-3-902613-06-6, pp.530, I-Tech, Vienna, Austria, June 2007

road. Most of recent works (e.g., Bertozzi et al., 2003b; Coulombeau & Laurgeau, 2002; Liang et al., 2003; Ponsa et al., 2005; Labayrade & Aubert, 2003) assume, or impose, a scene prior knowledge to simplify the problem. Although prior knowledge has been extensively used to tackle the driver assistance problem, it should be carefully used since it may lead to wrong results. Unlike previous works, this chapter presents an approach to estimate in real time stereo vision camera pose by using raw 3D data points.

This chapter is organized as follows. Section 2 summarizes some of the approaches proposed in the literature to compute on-board vision system pose. The proposed approach is described in section 3. Section 4 presents experimental results on urban scenes, together with comparisons with (Sappa et al., 2006). Finally, conclusions and further improvements are given in section 5.

2. Previous Approaches

In this work, since we only have a road plane equation, the camera pose will refer to two independent angles plus a translational distance. Several techniques have been proposed in the literature for robust vision system pose estimation. They can be classified into two different categories: *monocular* or *stereo*. In general, monocular systems are used on an off-line pose estimation basis. To this end, the car should be at rest and should face a flat road; once the camera pose is estimated its values are assumed to keep constant, or vary within a predefined range, during the on-line process (e.g., Ponsa et al., 2005; Bertozzi et al., 2003b). Although useful in most of highway scenarios, constant camera position and orientation is not a valid assumption to be used in urban scenarios since in general, vehicle pose is easily affected by road imperfections or artifacts (e.g., rough road, speed bumps), car's accelerations, uphill/downhill driving, to mention a few. Notice that since the vision system is rigidly attached to the vehicle, camera pose and vehicle pose are indistinctly used through this work.

In order to tackle urban scenarios, some monocular systems have been proposed to automatically compute camera pose by using the prior knowledge of the environment (e.g., Franke et al., 1998; Bertozzi et al., 2003a; Suttorp & Bücher, 2006). However, scene prior-knowledge not always can help to solve problems, in particular when cluttered and changing environment are considered, since visual features are not always available.

On the contrary to monocular approaches, stereo based systems in general are used on an on-line pose estimation basis. Since 3D data points are computed from every stereo pair, the corresponding vision system pose can be directly estimated related to these data whenever required. Broadly speaking, two different stereo matching schemes are used to compute 3D data points, either matching edges and producing sparse depth maps or matching all pixels in the images and producing dense depth maps (Faugeras & Luong, 2001). The final application is used to define whether preference is given to edge-based correspondences or to dense stereo correspondences. In general, for a successful reconstruction of the whole environment it is essential to compute dense disparity maps defined for every pixel in the entire image. However, the constraint of having a reduced computational complexity some times prevents the use of dense disparity maps. This very challenging problem has been usually tackled by making assumptions regarding the scene or by imposing constraints on the motion of the on-board stereo system. Furthermore, several solutions are proposed in order to compute 3D data points in a fast way based on ad hoc or application-oriented stereo vision systems. Although attractive, from the point of view of reduced processing time, the use of ad hoc stereo vision systems is limited since no other approaches could take advantage of those application-oriented 3D data points.

Different techniques relying on stereo vision systems have been proposed in the literature for driver assistance applications. For instance, the edge based *v*-disparity approach proposed in (Labayrade et al., 2002), for an automatic estimation of horizon lines and later on used for applications such as obstacle or pedestrian detection (e.g., Bertozzi et al., 2005; Broggi et al., 2003; Hu & Uchimura, 2005); it only computes 3D information over local maxima of the image gradient. A sparse disparity map is computed in order to obtain a real time performance. This *v*-disparity approach has been extended to a *u*-*v*-disparity concept in (Hu & Uchimura, 2005). In this new proposal, dense disparity maps are used instead of only relying on edge based disparity maps. Working in the disparity space is an interesting idea that is gaining popularity in on-board stereo vision applications, since planes in the original Euclidean space become straight lines in the disparity space. Up to our knowledge, all the approaches proposed to work on *v*-disparity space are based on Hough transform algorithm for extracting straight lines.

In this chapter a real time approach able to handle the whole 3D data points of a scene is presented. Hence, while the proposed technique is intended to estimate the stereo vision camera pose parameters, collision avoidance algorithms or pedestrian detection could make use of the same 3D data together with the estimated camera pose. In other words, the underlying idea of the proposed approach is to develop a standalone application that runs independently from others applications or hardware systems. In this sense, a commercial stereo pair is used, instead of relying on an ad hoc technology. This will allow us in the future to upgrade our current stereo vision sensor without changing the proposed technique.

3. Proposed Approach

The proposed approach consists of two stages. Initially, 3D data are mapped onto YZ plane (see Fig. 1), where a set of *candidate points* are selected—candidates to belong to the road. The main objective of this first stage is to take advantage of the 2D structured information before applying more expensive processing algorithms working with raw 3D data. Secondly, a RANSAC based least squares fitting is used to estimate the parameters of a plane (i.e., road plane) fitting to those candidate points. Finally, camera position and orientation are directly computed, referred to the fitted plane. Similarly to (Sappa et al., 2006), the provided results could be understood as a piecewise planar approximation, due to the fact that road and camera parameters are continuously computed and updated. Note that since on-board vision system pose is related to the current 3D road plane, camera position and orientation are equivalent to the 3D road plane parameters—3D plane parameters are expressed in the camera coordinate system. The proposed technique could be indistinctly used for urban or highway environments, since it is not based on a specific visual traffic feature extraction but on raw 3D data points. Before going into details about the proposed approach, the on-board stereo vision system is briefly introduced.

3.1 Stereovision System

A commercial stereo vision system (Bumblebee from Point Grey[1]) was used. It consists of two Sony ICX084 colour CCDs with 6mm focal length lenses. Bumblebee is a pre-calibrated system that does not require in-field calibration. The baseline of the stereo head is 12cm and it is connected to the computer by a IEEE-1394 connector. Right and left colour images were captured at a resolution of 640×480 pixels and a frame rate near to 30 fps. After capturing these right and left images, 3D data were computed by using the provided 3D reconstruction software. Fig. 1 shows an illustration of the on board stereo vision system.

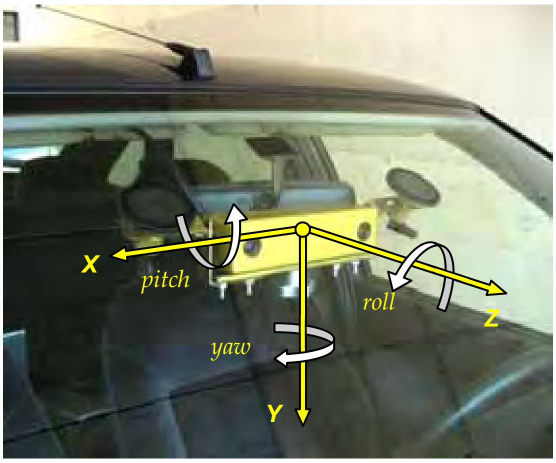

Figure 1. On-board stereo vision sensor with its corresponding coordinate system

3.2 3D data point projection and noisy data filtering

Let $S(r,c)$ be a stereo image with R rows and C columns, where each array element (r,c) ($r \in [0,(R-1)]$ and $c \in [0,(C-1)]$) is a scalar that represents a surface point of coordinates (x,y,z), referred to the sensor coordinate system. Fig. 1 depicts the sensor coordinate system attached to the vehicle's windshield. Notice that vertical variations between consecutive frames—due to road imperfections, car accelerations, changes in the road slope: *flat/uphill/downhill* driving, etc—will mainly produce changes on camera height and pitch angle (camera height is defined as the distance between the origin of the coordinate system and the road plane). In other words, yaw and roll angles are not so affected by those variations. Even though the roll angle is not plotted in this paper, its value is easily retrieved from the plane equation. The estimation of yaw angle is not considered in this work.

The aim at this stage is to find a compact subset of points, ζ, containing most of the road points. Additionally, noisy data points should be reduced as much as possible in order to avoid both a very time consuming processing and a wrong plane fitting.

Original 3D data points (x_i, y_i, z_i) are mapped onto a 2D discrete representation $P(u,v)$; where $u = \lfloor(y_i \cdot \sigma\rfloor$ and $v = \lfloor(z_i \cdot \sigma\rfloor$. σ represents a scale factor defined as: $\sigma=((R+C)/2)/((\Delta X+\Delta Y+\Delta Z)/3)$; R, C are the image's rows and columns respectively, and

[1] www.ptgrey.com

(ΔX, ΔY, ΔZ) is the working range in every dimension—on average (34×12×50) meters. Every cell of *P(u,v)* keeps a pointer to the original 3D data point projected onto that position, as well as a counter with the number of mapped points. Fig. 2(*top-right*) shows the 2D representation obtained after mapping the 3D cloud presented in Fig. 2(*left*)—every black point represents a cell with at least one mapped 3D point.

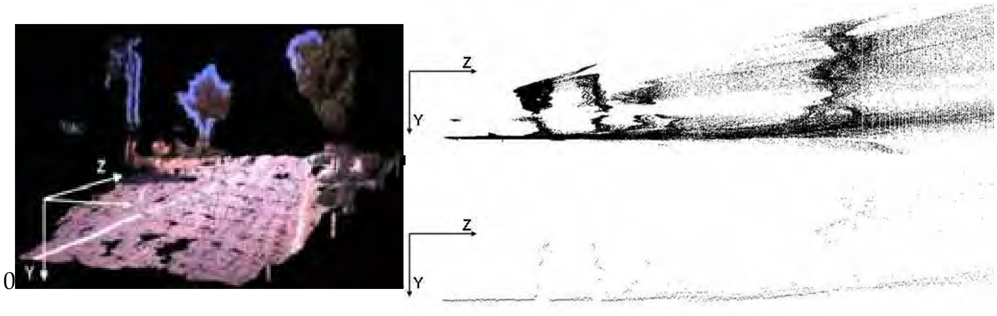

Figure 2. (*left*) 3D data points from the stereo rig. (*top-right*) Points projected to the YZ plane. (*bottom-right*) Cells finally selected to be used during the plane fitting stage (notice that one cell per column has been selected using the dynamic threshold)

Finally, points defining the 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 goes bottom-up, in the 2D representation, through every column, and picks the first cell with more points than an adaptive threshold, τ. Cells containing less mapped points than τ are filtered by setting to zero its corresponding counter—points mapped onto those cells are considered as noisy data. The value of τ is defined for every column as 80% of the maximum amount of points mapped onto the cells of that column. It avoids the use of a fixed threshold value for all columns. Recall that the density of points decreases with the distance to the sensor, hence the threshold value should depend on the depth—the column position in the 2D mapping. This is one of the differences with respect to (Sappa et al., 2006), where a constant threshold value was defined. Fig. 2(*bottom-right*) depicts cells finally selected. The ζ subset of points gathers all the 3D points mapped onto those cells.

3.3 RANSAC based plane fitting

The outcome of the previous stage is a subset of points, ζ, where most of them belong to the road. In the current stage a RANSAC based technique (Fischler & Bolles, 1981) is used for fitting a plane to those data[2], ax+by+cz=1. In order to speed up the process, a predefined threshold value for inliers/outliers detection has been defined (a band of ±5cm was enough for taking into account both 3D data point accuracy and road planarity). An automatic threshold could be computed for inliers/outliers detection following robust estimation of standard deviation of residual errors (Rousseeuw & Leroy, 1987).

[2] Notice that the general expression $ax+by+cz+d=0$ has been simplified dividing by $(-d)$, since we already know that $(d \neq 0)$.

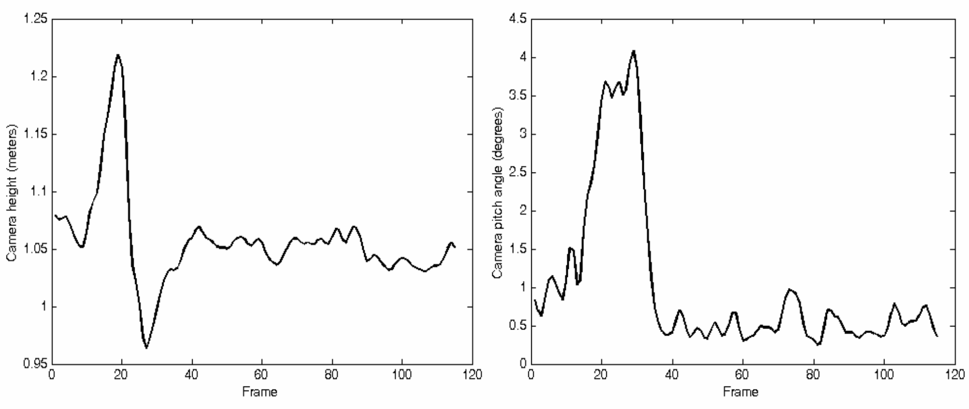

Figure 3. Results from a short video sequence: (*left*) Camera height; (*right*) Camera pitch angle

The proposed plane fitting works as follows.

Random sampling: *Repeat the following three steps K times (in our experiments K=100)*

1. Draw a random subsample of 3 different 3D points from ζ.
2. For this subsample, indexed by k $(k = 1, ... , K)$, compute the plane parameters (a,b,c).
3. For this solution $(a,b,c)_k$, compute the number of inliers among the entire set of 3D points contained in ζ, as mentioned above using ±5cm as a fixed threshold value.

Solution:

1. Choose the solution that has the highest number of inliers. Let $(a,b,c)_i$, be this solution.
2. Refine $(a,b,c)_i$ considering its corresponding inliers, by using the least squares fitting approach (Wang et al., 2001), which minimize the square residual error $(1\text{-}ax\text{-}by\text{-}cz)^2$.
3. In case the number of inliers is smaller than 10% of the total amount of points contained in ζ, those plane parameters are discarded and the ones corresponding to the previous frame are used as the correct ones. In general, this happens when 3D road data are not correctly recovered since occlusion or other external factor appears.

Finally, camera height (h) and orientation (Θ), referred to the fitted plane (a,b,c), are easily computed. Camera height is given by: $h = 1/\sqrt{a^2 + b^2 + c^2}$. Camera orientation—pitch angle—is directly computed from the current plane orientation: $\Theta = \arctan(c/b)$. Both values can be represented as a single one by means of the horizon line (e.g., Zhaoxue & Pengfei, 2004; Rasmussen, 2004a; Rasmussen, 2004b), in particular this compact representation will be used in the next section for comparisons. The horizon line position (v_i) for a given frame (i) is computed by back-projecting into the image plane a point lying over the plane, far away from the camera reference frame, $P_i(x, y, z)$. Let ($y_i = (1 - cz_i)/b$) be the y coordinate of P_i by assuming $x_i=0$. The corresponding y_i back-projection into the image plane, which define the row position of the sought horizon line, is obtained as $v_i = v_0 + f\, y_i \,/\, z_i = v_0 + f/(z_i\, b) - f\, c/b$; where, f denotes the focal length in pixels; v_0 represents the vertical coordinate of the principal point; and z_i is the depth value of P_i (in the experiments $z_i = 10{,}000$).

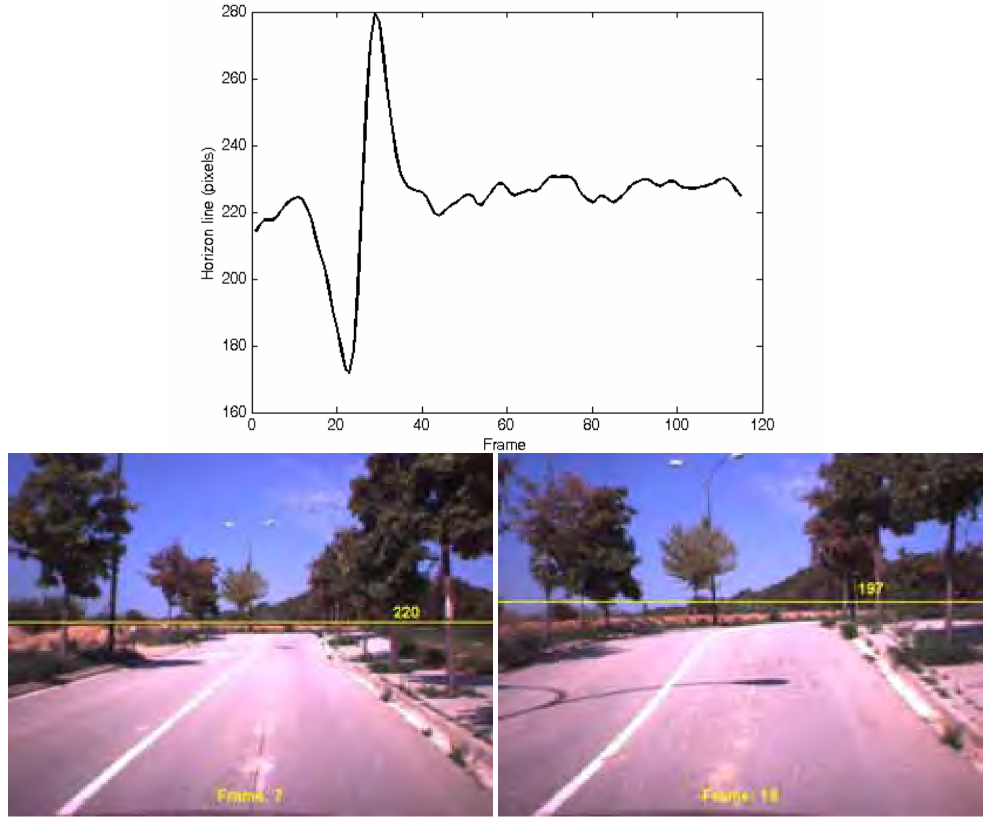

Figure 4. (*top*) Horizon line for the video sequence presented in Fig. 3. (*bottom*) Two single frames with their corresponding horizon line

4. Experimental Results

The proposed technique has been tested on different urban environments. The proposed algorithm took, on average, 350 ms per frame on a 3.2 GHz Pentium IV PC with a non-optimized C++ code.

Fig. 3 presents variations in the camera height and pitch angle during a sequence of about one minute long—only variations in the camera height position and pitch angle are plotted, both related to the current fitted plane. Notice that, although short, this video sequence contains downhill/uphill/flat scenarios (see Fig. 4 (*bottom*)). This illustration shows that variations in the camera position and orientation cannot be neglected, since they can change considerably in a short trajectory (something that does not happen on highways scenarios). These variations can be easily appreciated on the horizon line representation presented in Fig. 4 (*top*).

A comparison between the proposed technique and (Sappa et al., 2006) has been performed using a 100 frame-long-video sequence. The main difference between these techniques lies on the way cells to be fitted are selected, section 3.2. Fig. 5 presents camera height and pitch angle of both approaches. Fig. 6 depicts the corresponding horizon line position, computed with both approaches, as a function of the sequence frames. Although both approaches give similar results, values obtained with the new proposal are more reliable and fit better the current road geometry, since not only cells near to the sensor but the whole set of point on the direction of the camera optical axis is used. Finally, Fig. 7 presents four single frames of this video sequence together with their corresponding horizon line.

The proposed technique is already being used on a shape-based pedestrian detection algorithm (Gerónimo et al., 2006) in order to speed up the searching process. Although out of the scope of this paper, Fig. 8 presents illustrations of two different scenarios showing the importance of having the right estimation of camera position and orientation. In these illustrations, *(a)*, *(b)* and *(c)* columns show results by using three different, but constant, horizon line positions, while *(d)* column depicts the corresponding results obtained by using a horizon line position automatically computed by the proposed technique. Following the algorithm presented in (Ponsa et al., 2005), a 3D grid, sampling the road plane, is projected on the 2D image. The projected grid nodes are used as references to define the bottom-left corners of pedestrian sized searching windows. These windows, which have a different size according to their corresponding 3D depth, move backward and forward over the assumed plane looking for a pedestrian-like shape. Therefore, a wrong road plane orientation—i.e., horizon line—drives to a wrong searching space, so that the efficiency of the whole algorithm decreases. A few searching bounding boxes are highlighted in Fig. 8 to show their changes in size according to the distance to the camera.

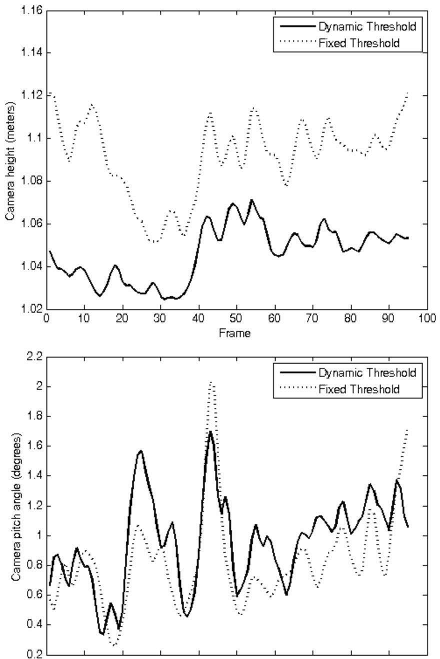

Figure 5. Results obtained by using the proposed technique (dynamic threshold) and (Sappa et al., 2006) (fixed threshold): (*top*) Camera height; (*bottom*) Camera pitch angle

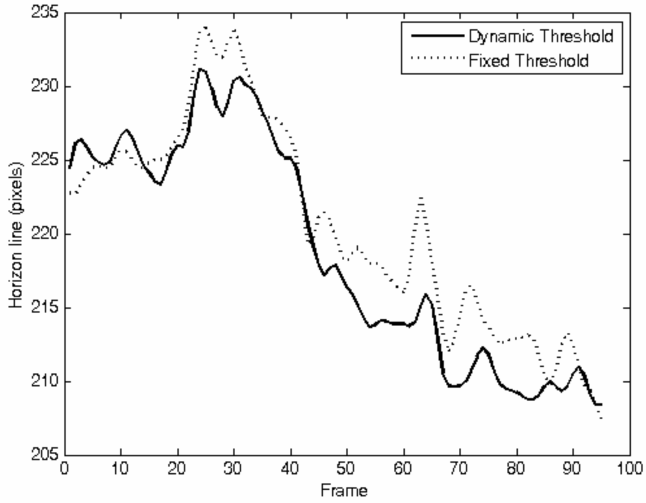

Figure 6. Horizon line position corresponding to the sequence presented in Figure 5.

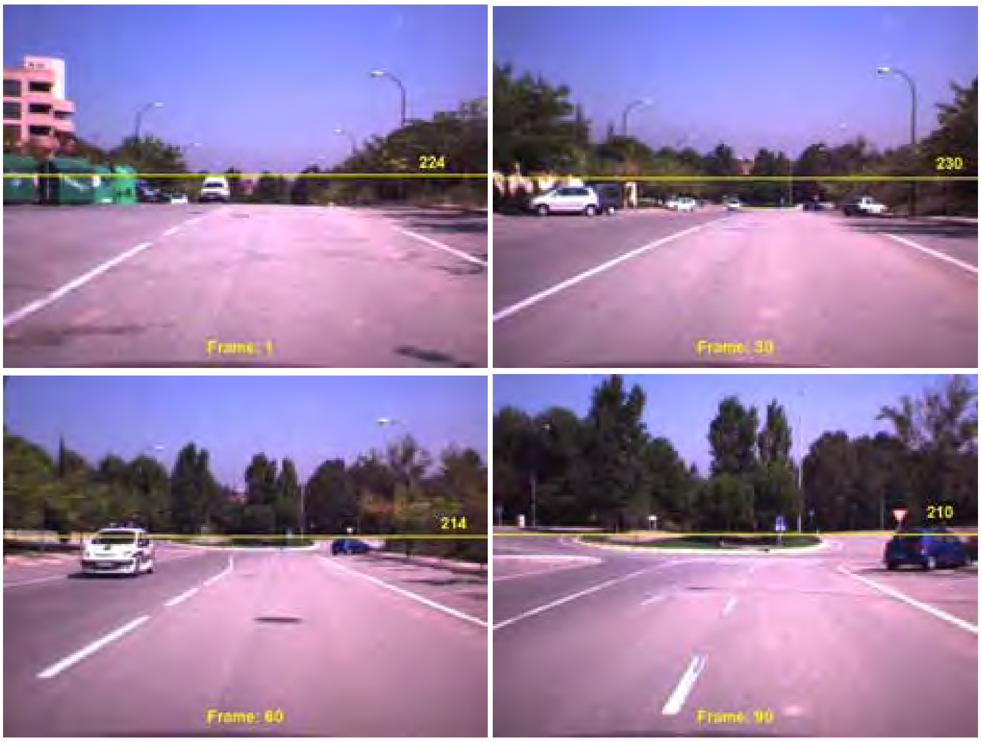

Figure 7. Horizon line for four different frames of the sequence presented in Figure 5.

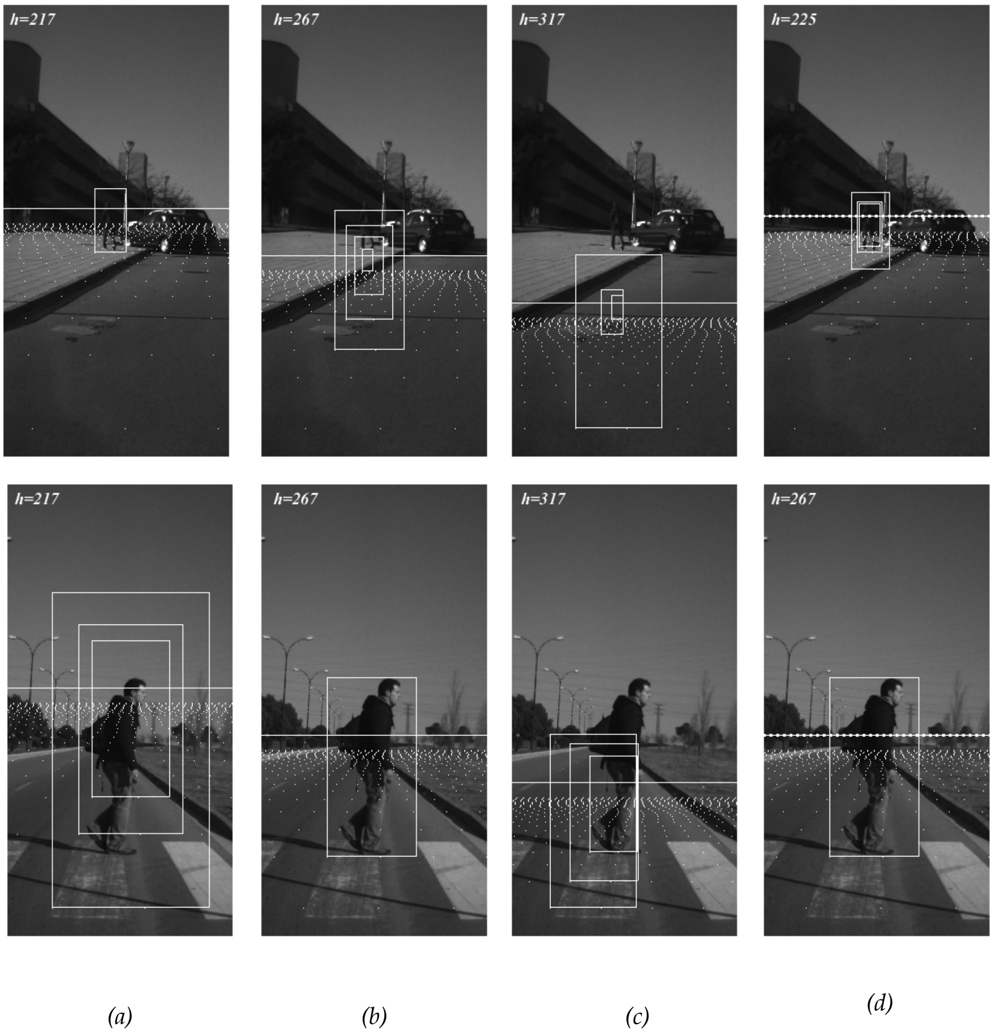

Figure 8. Searching bounding boxes using fixed and automatically computed horizon lines. In all the cases only very few bounding boxes are highlighted. Fixed horizon line ASSUMING: *(a)* an uphill road; *(b)* a flat road; *(c)* a downhill road. *(d)* Automatically computed horizon line by using the proposed technique. Notice that, only in the latter case, the horizon line position is correctly placed in both scenarios.

5. Conclusions and Further Improvements

An efficient technique for a real time pose estimation of an on-board camera has been presented. It improves a previous proposal (Sappa et al., 2006) by defining a dynamic threshold for selecting points to be fitted. After an initial mapping and filtering process, a compact set of points is chosen for fitting a plane to the road. The proposed technique can fit very well to different road geometries, since plane parameters are continuously computed and updated. A good performance has been shown in several scenarios—uphill, downhill and flat roads. Furthermore, critical situations such as car's accelerations or speed bumps were also considered. Although it has been tested on urban environments, it could be also useful on highways scenarios.

Further work will be focused on developing new strategies in order to reduce the initially chosen subset of points; for instance by using a non-constant cell size for mapping the 3D world to 2D space (through the optical axis). A reduced set of points will help to reduce the whole CPU time. Furthermore, the use of Kalman filtering techniques and other geometries for fitting road points will be explored.

6. References

Bertozzi, M.; Broggi, A.; Carletti, M.; Fascioli, A.; Graf, F.; Grisleri, P. & Meinecke, M. (2003a). IR pedestrian detection for advaned driver assistance systems, *Proceedings of 25th. Pattern Recognition Symposium*, pp. 582–590, Magdeburg, Germany, September 2003.

Bertozzi, M.; Broggi, A.; Chapuis, R.; Chausse, F.; Fascioli, A. & Tibaldi, A. (2003b). Shape-based pedestrian detection and localization, *Proceedings of IEEE Int. Conf. on Intelligent Transportation Systems*, pp. 328–333, Shangai, China, October 2003.

Bertozzi, M.; Binelli, E.; Broggi, A. & Del Rose, M. (2005). Stereo vision-based approaches for pedestrian detection, *Proceedings of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 23–28, San Diego, USA, June 2005.

Broggi, A.; Fascioli, A.; Fedriga, I.; Tibaldi, A. & Del Rose, M. (2003). Stereo-based preprocessing for human shape localization in unstructured environments, *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 410–415, Columbus, OH, USA, June 2003.

Coulombeau P. & Laurgeau, C. (2002). Vehicle yaw, pitch, roll and 3D lane shape recovery by vision, *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 619–625, Versailles, France, June 2002.

Faugeras, O. & Luong, Q. (2001). The Geometry of Multiple Images: the Laws that Govern the Formation of Multiple Images of a Scene and Some of their Applications, MIT, 2001.

Fischler M. & Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Graphics and Image Processing*, vol. 24, no. 6, pp. 381–395, June 1981.

Franke, U.; Gavrila, D.; Görzig, S.; Lindner, F.; Paetzold, F. & Wöhler, C. (1998). Autonomous driving goes downtown, *IEEE Intelligent Systems and Their Applications*, pp. 40–48, January 1998.

Gerónimo, D.; Sappa, A.; López, A. & Ponsa, D. (2006). Pedestrian detection using adaboost learning of features and vehicle pitch estimation, *Proceedings of Int. Conf. on Visualization, Imaging, and Image Processing*, pp. 400–405, Palma de Mallorca, Spain, August 2006.

Hu, Z. & Uchimura, K. (2005). U-V-Disparity: an efficient algorithm for stereovision based scene analysis, *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 48–54, Las Vegas, USA, June 2005.

Labayrade, R.; Aubert, D. & Tarel, J. (2002). Real time obstacle detection in stereovision on non flat road geometry through 'V-disparity' representation, *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 646–651, Versailles, France, June 2002.

Labayrade, R. & Aubert, D. (2003)., In-vehicle obstacles detection and characterization by stereovision, *Proceedings of 1st. Int. Workshop In-Vehicle Cognitive Computer Vision Systems*, pp. 13–19, Graz, Austria, April 2003.

Liang, Y.; Tyan, H.; Liao, H. & Chen, S. (2003). Stabilizing image sequences taken by the camcorder mounted on a moving vehicle, *Proceedings of IEEE Int. Conf. on Intelligent Transportation Systems*, pp. 90–95, Shangai, China, October 2003.

National Center of Health Statistics. (2002). National vital statistics report.

Ponsa, D.; López, A.; Lumbreras, F.; Serrat, J. & Graf, T. (2005). 3D vehicle sensor based on monocular vision, *Proceedings of IEEE Int. Conf. on Intelligent Transportation Systems*, pp. 1096–1101, Vienna, Austria, September 2005.

Rasmussen, C. (2004a). Grouping dominant orientations for ill-structured road following, *Proceedings of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 470–477, Washington, USA, June 2004.

Rasmussen, C. (2004b). Texture-based vanishing point voting for road shape estimation, *Proceedings of British Machine Vision Conference*, London, UK, September 2004.

Rousseeuw P. & Leroy, A. (1987). Robust Regression and Outlier Detection, John Wiley & Sons, New York, 1987.

Sappa, A. ; Gerónimo, D.; Dornaika, F. & López, A. (2006). Real time vehicle pose using on-board stereo vision system, *Proceedings of Int. Conf. on Image Analysis and Recognition*, LNCS Vol. 4142, Springer Verlag, pp. 205–216, Póvoa de Varzim, Portugal, September 2006.

Suttorp, T. & Bücher, T. (2006). Robust vanishing point estimation for driver assistance, *Proceedings of IEEE Intelligent Transportation Systems*, pp. 1550–1555, Toronto, Canada, September 2006.

United Nations Economic Commission for Europe. (2005). Statistics of road traffic accidents in Europe and North America.

Wang, C.; Tanahashi, H.; Hirayu, H.; Niwa, Y. & Yamamoto, K. (2001). Comparison of local plane fitting methods for range data, *Proceedings of IEEE Computer Vision and Pattern Recognition*, pp. 663–669, Hawaii, December 2001.

Zhaoxue, C. & Pengfei, S. (2004). Efficient method for camera calibration in traffic scenes, *Electronics Letters*, vol. 40, no. 6, pp. 368–369, March 2004.