Efficient On-Board Stereo Vision Pose Estimation*

Angel D. Sappa[1], Fadi Dornaika[2], David Gerónimo[1], and Antonio López[1]

[1] Computer Vision Center, Edifici O Campus UAB
08193 Bellaterra, Barcelona, Spain
{asappa, dgeronimo, antonio}@cvc.uab.es
[2] Institut Géographique National
94165 Saint Mandé, France
fadi.dornaika@ign.fr

Abstract. This paper presents an efficient technique for real time estimation of on-board stereo vision system pose. The whole process is performed in the Euclidean space and consists of two stages. Initially, a compact representation of the original 3D data points is computed. Then, a RANSAC based least squares approach is used for fitting a plane to the 3D road points. Fast RANSAC fitting is obtained by selecting points according to a probability distribution function that takes into account the density of points at a given depth. Finally, stereo camera position and orientation—pose—is computed relative to the road plane. The proposed technique is intended to be used on driver assistance systems for applications such as obstacle or pedestrian detection. A real time performance is reached. Experimental results on several environments and comparisons with a previous work are presented.

1 Introduction

Several vision based advanced driver assistance systems (ADAS) have been proposed in the literature during recent years (e.g., [1], [2], [3]). They can be broadly classified into two different categories: *monocular* or *stereo*. Each one of them has its own advantages and disadvantages making it difficult to decide which is the best approach for a general purpose driver assistance system.

In general, monocular vision systems avoid problems related to 3D Euclidean geometry by using the prior knowledge of the environment as an extra source of information. However, it may lead to wrong results. For instance, considering a constant camera's position and orientation is not a valid assumption to be used in urban scenarios, since both of them are easily affected by road imperfections or artifacts (e.g., rough road, speed bumpers), car's accelerations, uphill/downhill driving, among others. Facing up to this problem [4] introduces a technique for

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

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1183–1190, 2007.

estimating vehicle's yaw, pitch and roll. Since a single camera is used, it is based on the assumption that some parts of the road have a constant width (e.g., lane markings). Similarly, [5] proposes to estimate camera's orientation by assuming that the vehicle is driven along two parallel lane markings. Unfortunately, none of these two approaches can be generalized to be used in urban scenarios, since in general lanes are not as well defined as those of highways.

The main advantage of monocular systems is their high capture rate, which is at the same time the weakest point of current stereo systems. On the other hand, the main advantage of stereo vision systems lies in the richness of 3D information, which allows to face up problems that cannot be tackled with monocular systems without having a prior knowledge of the scene. In other words, drawbacks of stereo vision systems, like the low capture rate, are related to the current technology while drawbacks of monocular vision systems are due to their monocular nature. Therefore, taking into account the fast evolution of technology it is assumed that most of the stereo systems drawbacks, which are related to the current technology, will be surpassed soon.

In this context, [6] presents an algorithm for on-board camera extrinsic parameter estimation. Although robust, the major drawback of that technique is the high CPU time required to process the whole set of data points. In the current paper a two stage technique is presented; it introduces improvements over that original approach ([6]) by using a compact set of points. An efficient RANSAC based least squares fitting approach estimates the parameters of a plane fitting to that set of points. Finally, camera's position and orientation are directly computed, referred to that plane. 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.

The remainder of this paper is organized as follows. Section 2 presents the proposed technique. Experimental results on urban scenes are presented in Section 3 together with comparisons with previous approaches. Finally, conclusions are given in Section 4.

2 Proposed Technique

Let $D(r, c)$ be a depth map with R rows and C columns (the image size), where each array element $(r, c)(r \in [0, (R-1)]$ and $c \in [0, (C-1)])$ is a 3-vector that represents a scene point of coordinates (x, y, z) in the sensor coordinate system. Figure 1 depicts the sensor coordinate system of the stereo camera that is attached to the vehicle's windshield. Due to the orientation alignment between the sensor coordinate system and the vehicle, one can assume that vertical variations between consecutive frames—due to road imperfections, car accelerations, changes in the road slope, etc.—will mainly produce changes in camera's height and pitch angle. In other words, yaw and roll angles are not so affected by those variations. In practice, all three angles can change, however in this study we are only interested in pitch angle variations. The proposed approach consists of two stages, which are presented below.

Fig. 1. On-board stereo vision sensor with its corresponding coordinate system (right camera coordinate system is used as reference).

2.1 3D Data Point Projection and Cell Selection

The aim at this first stage is to find a compact subset of points, ζ, containing most of the road's points; similar to our previous proposal [6]. Additionally, noisy data points should be reduced as much as possible in order to avoid both a 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 2D space.

Original 3D data points, $D(r, c)$, are mapped onto a 2D discrete representation $P(u, v)$; where $u = \lfloor D_y(r, c) \cdot \sigma \rfloor$ and $v = \lfloor D_z(r, c) \cdot \sigma \rfloor$. 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 $(\Delta X, \Delta Y, \Delta Z)$ is the working range in 3D space—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 3D points. Figure 2*(bottom-left)* shows a 2D representation obtained after mapping the 3D cloud presented in Figure 2*(top-right)*.

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 picks one cell per column in the 2D projection (the cell with the largest number of points in that column). It avoids the use of a fixed threshold value for the whole 2D space. This is one of the differences with respect to [6], where a constant threshold value was used in the cell selection process.

Finally, in order to reduce the processing time, every selected cell is represented by the barycenter of its mapped points. The set of these barycenters define the sought subset of points, ζ. This data compression step is another difference with [6], where all mapped points into the selected cells were used for the fitting process. Using one single point per selected cell a considerable reduction in the CPU time is reached.

2.2 RANSAC Fitting with a Compact Set of 3D Points

The outcome of the previous stage is a compact subset of points, ζ, where most of them belong to the road (Figure 2*(bottom-right)*). However, since some outliers

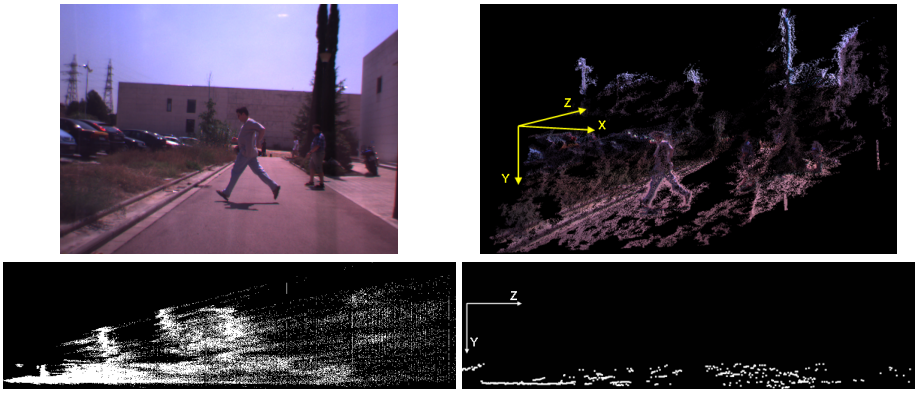

Fig. 2. (*top*) A single frame (right camera) together with the 3D data points computed with the stereo rig—notice that the image contains a large amount of holes due to occlusions and noisy regions. *(bottom-left)* YZ projection. *(bottom-right)* Cells finally selected to be used during the plane fitting stage.

are also included in that subset of points a RANSAC based [7] approach is used for computing plane parameters. Every selected cell is associated with a value that takes into account the amount of points mapped onto that position. This value will be considered as a probability density function. A cell containing a large number of mapped points, will have a high chance of being selected during the random sampling stage; at the same time RANSAC algorithm will find easier the Consensus among the whole set of point. The normalized probability density function is defined as follow:

$$f_{(i)} = \frac{n_{(i)}}{N} \tag{1}$$

where $n_{(i)}$ represents the number of points mapped onto the cell i (Figure $3(left)$) and N represents the total amount of points contained in the selected cells. Recall that we have one cell per column i. Next, a cumulative distribution function, $F_{(j)}$, is obtained as:

$$F_{(j)} = \sum_{i=0}^{j} f_{(i)} \tag{2}$$

If the values of F are randomly sampled at n points (with a uniform distribution), the application of the inverse function F^{-1} to those points leads to a set of n points that are adaptively distributed according to $f_{(i)}$. This principle is illustrated in Figure $3(right)$ where three points are randomly selected.

The fitting process computes plane parameters by means of an efficient RANSAC based least squares approach. Although an automatic threshold could be computed for inliers/outliers detection, following robust estimation of standard deviation of residual errors [8], we finally decided to define a fixed value in order to reduce CPU time. Notice that robust estimation of standard deviation involves computationally expensive algorithms such as sorting function. Hence, a

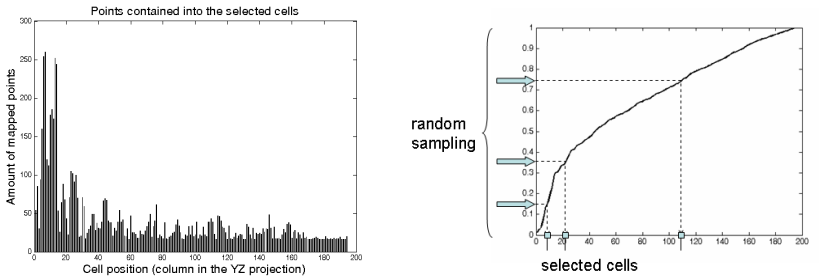

Fig. 3. *(left)* Bar diagram showing the amount of points mapped into the selected cells—recall that only one cell per column is picked up. *(right)* Cumulative distribution function computed from the amount of points mapped into every single cell.

predefined threshold value for inliers/outliers detection has been defined (a band of ±10 cm was enough for taking into account both 3D data point accuracy and road planarity). The proposed approach works as follows:

Random sampling. Repeat the following three steps K times, in our experiments K was set to 100:

1. Draw a random subsample of three different 3D points (P_1, P_2, P_3)— barycenters —where every point is drawn according to the probability density function $f_{(i)}$ using the above process (Figure 3(*right*)).
2. For this subsample, indexed by $k(k = 1, ..., K)$, compute the plane parameters[1] (a, b, c). Since P_i are barycenter points, they could define a set of collinear points; therefore, to prevent this occurs, their coordinates are set up as follow:$P_{1(x_{min}, y_{(1)}, z_{(1)})}$, $P_{2(x_{min}+(x_{max}-x_{min})/2,\ y_{(2)},\ z_{(2)})}$, $P_{3(x_{max},\ y_{(3)},\ z_{(3)})}$, where x_{min} and x_{max} correspond to the minimum and maximum x coordinate of the original whole set of points, respectively.
3. For this solution $(a, b, c)_k$, compute the number of inliers among the entire set of 3D points contained in ζ, using ±10cm 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 $(a, b, c)_i$ by using its corresponding inliers. To this end, the least squares fitting approach [9], which minimizes the square residual error $(1 - ax - by - cz)^2$ is used.
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 severe occlusion or other external factor appears.

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

Fig. 4. Vanishing lines computed according to the current camera pose—camera height and pitch angle

Finally, camera's height (h) and orientation (Θ), referred to the fitted plane (a, b, c), are easily computed. Camera's height is given by: $h = 1/\sqrt{a^2 + b^2 + c^2}$. Camera's 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 vanishing line (e.g., [10], [11]). The vanishing 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 vanishing line, is obtained as $v_{(i)} = v_{(0)} + fy_{(i)}/z_{(i)} = v_{(0)} + f/z_{(i)}b - fc/b$; 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 $z_{(i)} = 10000$).

3 Experimental Results and Comparisons

The proposed technique has been tested on different urban environments and compared with [6]. A 3.2 GHz Pentium IV PC with a non-optimized C++ code was used. The proposed algorithm took, on average, 90 ms per frame including both 3D points computation and on-board pose estimation. Notice that this is about four times faster than our previous approach [6], while the same results are obtained.

Figure 4 shows two different frames with their corresponding vanishing lines computed with the proposed technique. The computed camera height and pitch angle, as a function of time, for this sequence are presented in Figure 5. Both values are referred to the current fitted plane. This sequence contains a gentle downhill, vehicle's accelerations and two speed bumps. As can be seen, neglecting these variations will affect further processing (e.g., car or pedestrian detection, collision avoidance, etc.).

Finally, Figure 6 presents results obtained after processing a 12 second video sequence corresponding to a short flat road followed by an uphill (10 fps are depicted). Notice how the pitch angle changes during the sequence according

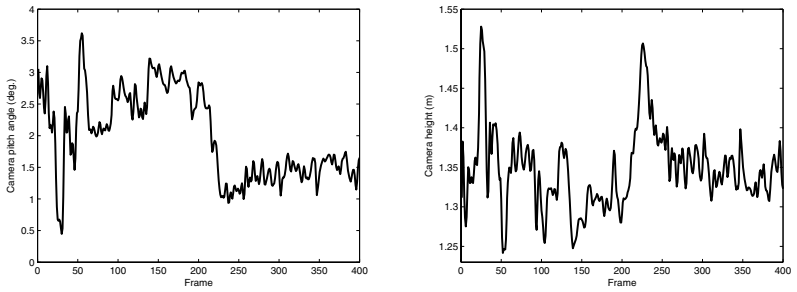

Fig. 5. *(left)* Camera pitch angle for the video sequence of Figure 4 (only 2 fps are plotted). *(right)* The corresponding camera distance to the fitted plane at every frame.

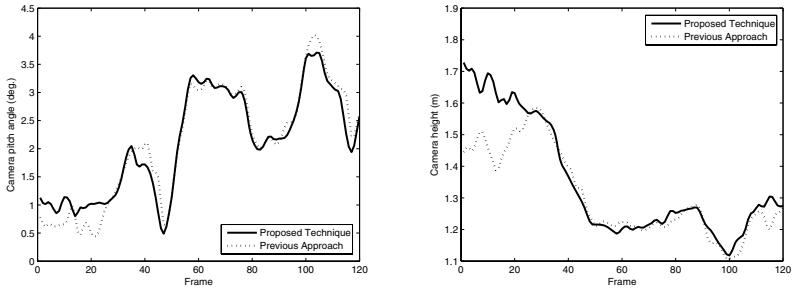

Fig. 6. Comparisons between the proposed technique and [6] with a 12 second-long video sequence: *(left)* Camera pitch angle. *(right)* The corresponding camera height to the fitted plane at the every frame.

to the current geometry. In this scene there are no speed bumps and the car keeps almost a constant speed after the initial acceleration, used for starting the car's motion. In both plots, the obtained results are presented together with the results obtained with [6]. As can be appreciated, although the obtained results have similar trend, the new proposed approach behaves better than the previous proposal in those critical situations where two different geometries converge, first 40 frames—in this case a flat road with a quite sharp uphill. Since our previous proposal uses a constant threshold value for cell selection (Section 2.1), only cells near to the sensor were considered; on the contrary, with the new approach all candidate cells are considered

4 Conclusions

An efficient technique for a real time pose estimation of on-board camera has been presented. The input data are a set of 3D points provided by the on-board stereo camera. After an initial mapping a compact set of 3D points is chosen as

candidate for fitting a plane to the road. The RANSAC technique selects points according to a probability distribution function that takes into account density of points at a given position. Although it has been tested on urban environments, it could be also useful on highway scenarios. A considerable reduction in the CPU processing time was reached by working with a reduced set of points selected according to a continuously updated probability distribution function. The latter drives to a faster convergence during the RANSAC fitting stage.

References

1. Bertozzi, M., Binelli, E., Broggi, A., Del Rose, M.: Stereo vision-based approaches for pedestrian detection. In: Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, San Diego, USA, IEEE Computer Society Press, Los Alamitos (2005)
2. Gerónimo, D., Sappa, A., López, A., Ponsa, D.: Adaptive image sampling and windows classification for on–board pedestrian detection. In: Proceedings of the International Conference on Computer Vision Systems, Bielefeld, Germany (2007)
3. Hautière, N., Tarel, J., Lavenant, J., Aubert, D.: Automatic fog detection and estimation of visibility distance through use of an onboard camera. Machine Vision and Applications 17(1), 8–20 (2006)
4. Coulombeau, P., Laurgeau, C.: Vehicle yaw, pitch, roll and 3D lane shape recovery by vision. In: Proc. IEEE Intelligent Vehicles Symposium, Versailles, France, pp. 619–625. IEEE Computer Society Press, Los Alamitos (2002)
5. Liang, Y., Tyan, H., Liao, H., Chen, S.: Stabilizing image sequences taken by the camcorder mounted on a moving vehicle. In: Proc. IEEE Int. Conf. on Intelligent Transportation Systems, Shangai, China, pp. 90–95. IEEE Computer Society Press, Los Alamitos (2003)
6. Sappa, A., Gerónimo, D., Dornaika, F., López, A.: On-board camera extrinsic parameter estimation. Electronics Letters 42(13), 745–747 (2006)
7. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Graphics and Image Processing 24(6), 381–395 (1981)
8. Rousseeuw, P., Leroy, A.: Robust Regression and Outlier Detection. John Wiley & Sons, New York (1987)
9. Wang, C., Tanahashi, H., Hirayu, H., Niwa, Y., Yamamoto, K.: Comparison of local plane fitting methods for range data. In: Proc. IEEE Computer Vision and Pattern Recognition, Hawaii, pp. 663–669. IEEE Computer Society Press, Los Alamitos (2001)
10. Zhaoxue, C., Pengfei, S.: Efficient method for camera calibration in traffic scenes. Electronics Letters 40(6), 368–369 (2004)
11. Rasmussen, C.: Grouping dominant orientations for ill-structured road following. In: Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, pp. 470–477. IEEE Computer Society Press, Washington, USA (2004)