

Real Time On Board Stereo Camera Pose through Image Registration*

Fadi Dornaika

French National Geographical Institute (IGN)
94165 Saint-Mandé, France

fadi.dornaika@ign.fr

Angel D. Sappa

Computer Vision Center, UAB
08193 Bellaterra, Barcelona, Spain

sappa@cvc.uab.es

Abstract—This work addresses the real time estimation of on-board stereo head's position and orientation. Unlike existing works which rely on feature extraction either in the image domain or in 3D space, our approach directly estimates the unknown parameters from the stream of stereo pairs' brightness. The pose parameters are estimated by registering the right and left images, which is solved using two optimization techniques: the Differential Evolution algorithm and the Levenberg-Marquardt algorithm. We provide experiments and evaluations of performance.

Keywords: on-board stereo camera pose, image registration, differential evolution algorithm, non-linear optimization

I. INTRODUCTION

In recent years, several techniques to on-board vision pose estimation have been proposed [1], [2], [3]. Vision system pose estimation is required for any advanced driver assistance application. The real-time estimation of on-board vision system pose—position and orientation—is a challenging task since i) the sensor undergoes motions due to the vehicle dynamics and the road imperfections, and ii) the viewed scene is unknown and continuously changing.

Of particular interest is the estimation of on-board camera position and orientation related to the 3D road plane. Note that since the 3D plane parameters are expressed in the camera coordinate system, the camera position and orientation are equivalent to the 3D plane parameters. Algorithms for fast road plane estimation are very useful for driver assistance applications as well as for augmented reality applications. The ability to use continuously updated plane parameters (camera pose) will considerably make the tasks of obstacles and objects detection more efficient [4], [5]. However, dealing with an urban scenario is more difficult than dealing with highways scenario since the prior knowledge as well as visual features are not always available in these scenes [6].

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. Although prior knowledge has been extensively used to tackle the driver assistance problem, it may lead to wrong results. Hence, considering a constant camera 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.

*This work was partially supported by the Government of Spain under MEC project TRA2007-62526/AUT and research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018). The second author was supported by The Ramón y Cajal Program.

In [7], authors use a single mounted camera. An extended Kalman filter has been used in order to infer a state vector including the vehicle rigid motion (six degrees of freedom) and the camera pose where the measurements are given by the 8-parameter planar motion field and the readings of the velocity and yaw rate sensors.

In the literature, many application-oriented stereo systems have been proposed. For instance, the edge based v -disparity approach proposed in [3], for an automatic estimation of horizon lines and later on used for applications such as obstacle or pedestrian detection (e.g., [8], [9]), 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. Recently, this v -disparity approach has been extended to a u - v -disparity concept in [10]. In this work, dense disparity maps are used instead of only relying on edge based disparity maps.

In [11], we have proposed an approach for on-line stereo camera pose estimation. Although the proposed technique does not require the extraction of visual features in the images, it is based on dense depth maps and on the extraction of a dominant 3D plane that is assumed to be the road plane. This technique has been tested on different urban environments. The proposed algorithm took, on average, 350 ms per frame. As can be seen, existing works adopt the following main stream. First, features are extracted either in the image space (optical flow, edges, ridges, interest points) or in the 3D Euclidean space (assuming the 3D data are built online). Second, an estimation technique is then invoked in order to recover the unknown parameters.

In this paper, we propose a novel paradigm for on-board camera pose tracking through the use of image registration. We solve the featureless registration by using two optimization techniques: the Differential Evolution algorithm (a stochastic search) and the Levenberg-Marquardt algorithm (a directed search). Moreover, we propose two tracking schemes based on these optimizations. The advantage of our proposed paradigm is twofold. First, it can run in real-time. Second, it provides good results even when the road surface does not have reliable features such as road and lane markings, i.e., the approach can be used for both highways and urban scenarios. The rest of the paper is organized as follows. Section II describes the problem we are focusing on as well as some backgrounds. Section III presents the proposed approach in details. Section IV gives some experimental results and method comparisons. Section V concludes the paper.

II. PROBLEM FORMULATION AND BACKGROUND

A. Experimental setup

A commercial stereo vision system (Bumblebee from Point Grey¹) was used. It consists of two Sony ICX084 color 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. Figure 1(a) shows an illustration of the on-board stereo vision system.

The problem we are focusing on can be stated as follows. Given a stream of stereo pairs provided by the on-board stereo head we like to recover the parameters of the road plane for every captured stereo pair. Since we do not use any feature that is associated with road structure, the computed plane parameters will completely define the pose of the on-board vision sensor. This pose is represented by the 3D plane parameters, that is, the height d and the plane normal $\mathbf{u} = (u_x, u_y, u_z)^T$ from which two independent angles can be inferred (See Figure 1(b)). In the sequel, the pitch angle will refer to the angle between the camera optical axis and the road plane; and the roll angle will refer to the angle between the camera horizontal axis and the road plane (See Figure 1(b)). Due to the reasons mentioned above, these parameters are not constant and should be estimated online for every time instant. Note that the three angles (pitch, yaw, and roll) associated with the stereo head orientation can vary. However, only the pitch and roll angles can be estimated from the 3D plane parameters.

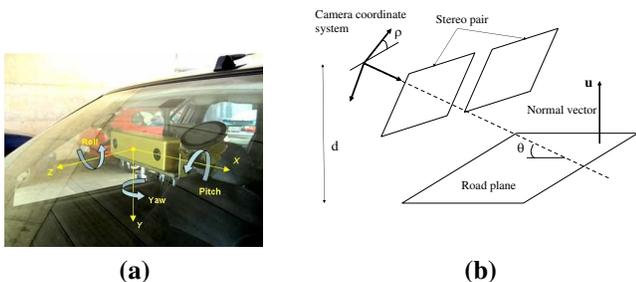


Fig. 1. (a) On-board stereo vision sensor. (b) The time-varying road plane parameters d and \mathbf{u} . θ denotes the pitch angle and ρ the roll angle.

B. Image transfer function

Before going into the details of the proposed approach, this section will describe the geometric relation between road pixels belonging to the same stereo pair—the left and right images. It is well-known [12] that the image coordinates of the projections of 3D points belonging to the same plane onto two different images are related by a 2D projective transform having 8 independent parameters—*homography*. In our setup, the right and left images are horizontally

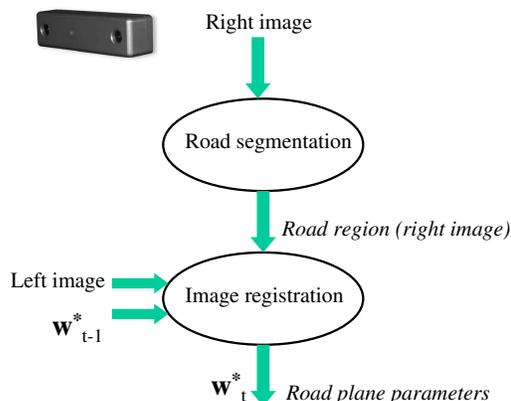


Fig. 2. The proposed approach consists of two stages: A rough road segmentation followed by image registration.

rectified². Let $p_r(x_r, y_r)$ and $p_l(x_l, y_l)$ be the right and left projection of an arbitrary 3D point P belonging to the plane (d, u_x, u_y, u_z) . In the case of a rectified stereo pair where the left and right cameras have the same intrinsic parameters, the right and left coordinates of corresponding pixels belonging to the road plane are related by the following linear transform (the homography reduces to a linear mapping)

$$x_l = h_1 x_r + h_2 y_r + h_3 \quad (1)$$

$$y_l = y_r \quad (2)$$

where h_1 , h_2 , and h_3 are function of the intrinsic and extrinsic parameters of the stereo head and of the plane parameters. For our setup (rectified images with the same intrinsic parameters), those coefficients are given by:

$$h_1 = 1 + b \frac{u_x}{d} \quad (3)$$

$$h_2 = b \frac{u_y}{d} \quad (4)$$

$$h_3 = -b u_0 \frac{u_x}{d} - b v_0 \frac{u_y}{d} + \alpha b \frac{u_z}{d} \quad (5)$$

where b is the baseline of the stereo head, α is the focal length in pixels, and (u_0, v_0) is the image center (principal point). Let \mathbf{w} be the 3-vector encapsulating the 3D plane parameters, that is, $\mathbf{w} = \frac{\mathbf{u}}{d}$.

$$\mathbf{w} = (w_x, w_y, w_z)^T = \left(\frac{u_x}{d}, \frac{u_y}{d}, \frac{u_z}{d} \right)^T \quad (6)$$

Note that the vector \mathbf{w} fully describes the current road plane parameters. The problem can be stated as follows. Given the current stereo pair estimate the corresponding 3D road plane parameters d and \mathbf{u} or equivalently the vector \mathbf{w} .

III. APPROACH

Since the goal is to estimate the road plane parameters \mathbf{w} for every stereo pair (equivalently the 3D pose of the stereo

²The use of non-rectified images will not have any impact on the theory of our developed method. However, the image transfer function will be given by a general homography.

¹[www.ptgrey.com]

head), the whole process is invoked for every stereo pair. Figure 2 illustrates the tracking of the stereo head pose over time. The inputs to the algorithm are the current stereo pair as well as the estimated road plane parameters associated with the previous frame. The algorithm is split into two consecutive stages. First, a rough road region segmentation is performed for the right image. Let \mathcal{R} denotes this region—a set of pixels. Second, recovering the plane parameters from the raw brightness of a given stereo pair will rely on the following fact: *if the parameter vector \mathbf{w} corresponds to the actual plane parameters—the distance d and the normal \mathbf{u} —then the registration error between corresponding road pixels in the right and left images over the region \mathcal{R} should correspond to a minimum.* In our work, the registration error is set to the Sum of Squared Differences (SSD) between the right image and the corresponding left image computed over the road region \mathcal{R} . The registration error is given by³:

$$e(\mathbf{w}) = \sum_{(x_r, y_r) \in \mathcal{R}} (I_r(x_r, y_r) - I_l(h_1 x_r + h_2 y_r + h_3, y_r))^2 \quad (7)$$

The corresponding left pixels are computed according to the linear transform given by (1) and (2). The computed $x_l = h_1 x_r + h_2 y_r + h_3$ is a non-integer value. Therefore, the grey-level, $I_l(x_l, y_l)$, is set to a linear interpolation of the grey-level of two neighboring pixels—the ones whose horizontal coordinates bracket x_l .

A. Road segmentation

In this section, we briefly describe how the road region \mathcal{R} is detected in the right images. Road segmentation is the focus of many research works [13], [14]. In our study, the sought segmentation should meet two requirements: (i) it should be as fast as possible, and (ii) it should be as generic as possible (both urban roads and highways). Thus our segmentation scheme will be a color-based approach which works on the hue and saturation components. The segmentation stage is split into two phases. The first phase is only invoked every T frames for updating the statistical color model and for obtaining a real-time performance. The second phase exploits the road color consistency over short time. Figure 3 shows the segmentation results obtained with the proposed scheme. Detected road pixels are shown in white within the ROI of two different frames.



Fig. 3. Rapid road segmentation associated with two frames.

³This functional error is justified by the fact that the two images are acquired by two identical cameras having the same orientation.

B. Image registration

The optimal current road parameters are given by

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} e(\mathbf{w}) \quad (8)$$

$$e(\mathbf{w}) = \sum_{(x_r, y_r) \in \mathcal{R}} (I_r(x_r, y_r) - I_l(h_1 x_r + h_2 y_r + h_3, y_r))^2$$

where $e(\mathbf{w})$ is a non-linear function of the parameters $\mathbf{w} = (w_x, w_y, w_z)^T$. In the sequel, we describe two minimization techniques: i) the Differential Evolution minimization, and ii) the Levenberg-Marquardt minimization. The first one is a stochastic search method and the second one is a directed search method. Moreover, we present two tracking schemes.

1) *Differential Evolution minimization*: The Differential Evolution algorithm (DE) is a practical approach to global numerical optimization that is easy to implement, reliable and fast [15]. We use the DE algorithm [16], [17] in order to minimize the error (8). This is carried out using generations of solutions—population. The population of the first generation is randomly chosen around a rough solution. We point out that even the exact solution for the first frame is not known, the search range for the camera height as well as for the plane normal can be easily known. For example, in our experiments, the camera height and the normal vector are assumed to be around $1m$ and $(0, 1, 0)^T$, respectively.

The optimization adopted by the DE algorithm is based on a population of N solution candidates $\mathbf{w}_{n,i}$ ($n = 1, \dots, N$) at iteration (generation) i where each candidate has three components. Initially, the solution candidates are randomly generated within the provided intervals of the search space. The population improves by generating new solutions iteratively for each candidate [17]. The solution is set to the best candidate among the last generation candidates.

Calibration. Since the stereo camera is rigidly attached to the car, the differential evolution algorithm can also be used as a calibration tool by which the camera pose can be estimated off-line. To this end, the car should be at rest and should face a flat road. Whenever the car moves, the off-line calibration results can be used as a starting solution for the whole tracking process. Note that the calibration process does not need to run in real-time.

2) *Levenberg-Marquardt minimization*: Minimizing the cost function (8) can also be carried out using the Levenberg-Marquardt technique [18], [19]—a well-known non-linear minimization technique. One can notice that the Jacobian matrix only depends on the horizontal image gradient since the right and left images are rectified.

C. Tracking schemes

Since the unknown parameters (road parameters/camera pose) should be estimated for every stereo pair, we propose two tracking schemes which are illustrated in Figure 4. The first scheme (Figure 4(a)) is only based on the Differential Evolution minimization. In other words, the solution for every stereo frame is computed by invoking the whole algorithm where the first generation is generated by diffusing

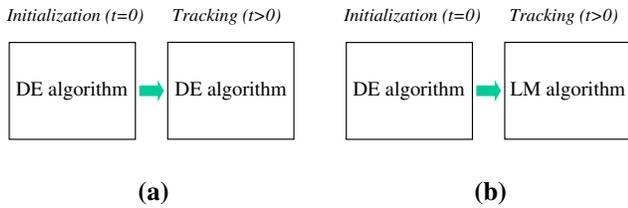


Fig. 4. Parameter tracking using two schemes. **(a)** The tracking is only based on the Differential Evolution search. **(b)** The tracking is based on the Differential Evolution search and on the Levenberg-Marquardt search.

the previous solution using a normal distribution. A uniform distribution is used for the first stereo frame.

The second scheme (Figure 4**(b)**) uses the Differential Evolution minimization for the first stereo frame only. It utilizes the Levenberg-Marquardt for the rest of the frames where the initial solution for a given frame is provided by the solution \mathbf{w}_{i-1}^* associated with the previous frame.

Although the first scheme might have better convergence properties than the second scheme, the latter one is better suited for real-time performance since the Levenberg-Marquardt algorithm is faster than the Differential Evolution search (the corresponding CPU times are illustrated in Section IV.B). In both tracking schemes, the pose parameters associated with the first stereo pair are estimated by the DE algorithm.

IV. EXPERIMENTAL RESULTS

The proposed technique has been tested on different urban environments since they correspond to the most challenging scenarios. In this section, we provide results obtained with two different videos associated with different urban road structures. Moreover, we provide a performance study using synthetic videos with ground-truth data.

A. Tracked pose parameters

The first experiment has been conducted on a sequence corresponding to an uphill driving. The stereo pairs are of resolution 320×240 . Figure 5**(a)** depicts the estimated camera height as a function of the sequence frames. Figures 5**(b)** depicts the estimated pitch angle as a function of the sequence frames. The dotted curves correspond to the first scheme that is based on the Differential Evolution minimization. The solid curves correspond to the second scheme which is based on both the Differential Evolution algorithm and the Levenberg-Marquardt algorithm. As can be seen, the estimated parameters are almost the same for the two proposed schemes. However, as we will show, the second scheme is much faster than the first scheme (the stochastic search).

a) Differential Evolution convergence: Figure 6 illustrates the behavior of the Differential Evolution algorithm associated with the first stereo pair of the above stereo sequence. This plot depicts the best registration error (SSD per pixel) obtained by every generation. The three curves correspond to three different population sizes. The first generation (iteration 0) has been built using a uniform sampling around

the solution $d = 1m$ and $\mathbf{u} = (u_x, u_y, u_z)^T = (0, 1, 0)^T$. The algorithm has converged in five iterations (generations) when the population size was 30 and in two iterations when the population size was 120. At convergence the solution was $d = 1.25m$ and $\mathbf{u} = (u_x, u_y, u_z)^T = (-0.03, 0.99, -0.02)^T$. Note that even the manually provided initial camera height has 25cm discrepancy from the current solution, the DE algorithm has rapidly converged to the actual solution. Also, we have run the Levenberg-Marquardt algorithm with the same starting solution but we get at convergence $d = 1.09m$ and $\mathbf{u} = (u_x, u_y, u_z)^T = (0.01, 0.99, -0.02)^T$.

b) Horizon line: In the literature, the pose parameters—plane parameters—can be used to compute the horizon line. In our case, since the roll angle is very small, the horizon line can be represented by an horizontal line in the image. Once the 3D plane parameters d and $\mathbf{u} = (u_x, u_y, u_z)^T$ are computed, the vertical position of the horizon line will be given by

$$v_h = v_0 + \frac{\alpha d}{u_y Z_\infty} - \frac{\alpha u_z}{u_y} \approx v_0 - \frac{\alpha u_z}{u_y} \quad (9)$$

The above formula is derived by projecting a 3D point $(0, Y_p, Z_\infty)$ belonging to the road plane and then taking the vertical coordinate $v = \alpha \frac{Y_p}{Z_\infty} + v_0$. Z_∞ is a large depth value. The right-hand expression is obtained by using the fact that u_y is close to one and Z_∞ is very large. Figure 7 illustrates the computed horizon line for frames 10 and 199. The whole video illustrating the computed horizon line can be found at www.cvc.uab.es/~asappa/HorizonLine.avi.

c) Occlusions and segmentation errors: In order to study the algorithm behavior in presence of significant occlusions or significant segmentation errors, we conducted the following experiment. We used a video sequence corresponding to a flat road (see Figure 3). We run the proposed tracking technique described in Section III.B twice (the second tracking scheme (DE-LM)). The first was a straightforward run. In the second run, the right images were modified to simulate a significant registration error. To this end we set the vertical half of a set of 20 right images to a fixed color. The left images were not modified.

Figure 8 compares the pose parameters obtained in the two runs. The solid curves were obtained with the non corrupted images. The dotted curves were obtained when the right images of the same sequence are artificially corrupted. The simulated corruption starts at frame 40 and ends at frame 60. The upper part of the Figure illustrates the stereo pair 40. As can be seen, the only significant discrepancy has affected the camera height. Moreover, one can see that the algorithm resumes to provide the correct parameters once the corruption has disappeared. Figure 9 shows the registration error obtained at convergence as a function of the sequence frames.

Figure 10**(a)** illustrates the registration error (8) as a function of the camera height while the orientation is kept fixed. Figure 10**(b)** illustrates the registration error as a function of the camera pitch angle for four different camera heights. In both figures, the depicted error is the SSD per

pixel. From the slope of the error function we can see that the camera height will not be recovered with the same accuracy as the plane orientation. This will be confirmed in the accuracy evaluation section (see Section IV.C).

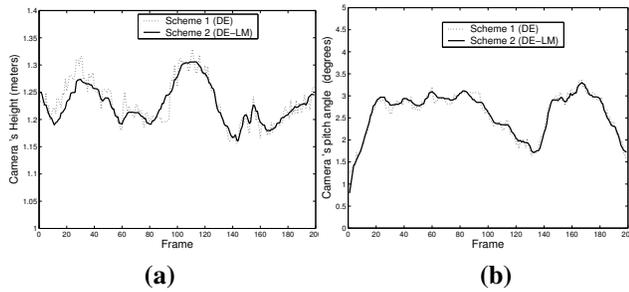


Fig. 5. Camera height and orientation computed by the proposed tracking schemes.

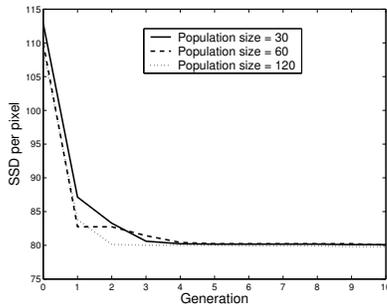


Fig. 6. The evolution of the best registration error obtained by the Differential Evolution algorithm associated with the first stereo pair.



Fig. 7. The estimated horizon line associated with frames 10 and 199. The sequence corresponds to an uphill driving.

B. Method comparison

The second experiment has been conducted on a short sequence of stereo pairs corresponding to a typical urban environment (see Figure 3). The stereo pairs are of resolution 320×240 . Here the road is almost flat and the changes in the pose parameters are mainly due to the car's accelerations and decelerations. Figures 11(a) and 11(b) depict the estimated camera height and orientation as a function of the sequence frames using two different methods. The solid curves correspond to the developed direct approach (DE-LM) and the dashed curves correspond to a 3D data based approach [11]. This approach uses a dense 3D reconstruction followed by a RANSAC-based estimation of the dominant 3D plane—the road plane. One can see that despite some discrepancies the

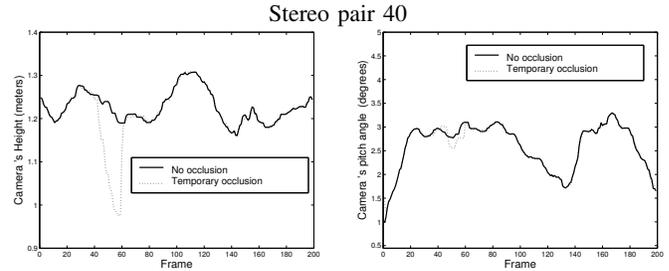


Fig. 8. Comparing the pose parameters when a significant occlusion occurs. The solid curves are obtained with the non corrupted images. The dotted curves are obtained when 20 frames of right images of the same sequence are artificially corrupted. The occlusion is simulated by setting the vertical half of the right images to a fixed color. This occlusion starts at frame 40 and ends at frame 60.

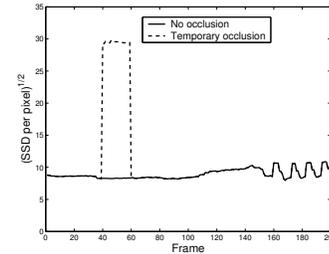
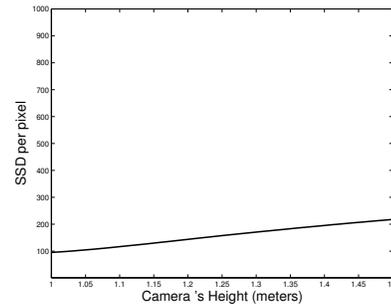
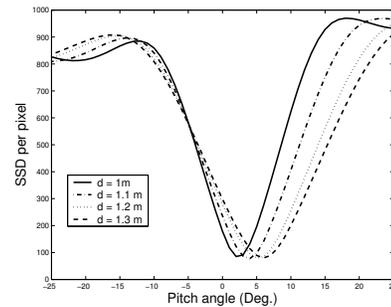


Fig. 9. The registration error obtained at convergence as a function of the sequence frame. The second tracking scheme is used.



(a)



(b)

Fig. 10. The registration error as a function of the camera pose parameters. (a) depicts the error as a function of the camera height with a fixed orientation. (b) depicts the error as a function of the camera pitch angle associated with four different camera heights.

proposed direct method is providing the same behavior of changes.

On a 3.2 GHz PC, the proposed approach processes one stereo pair in about 20 ms assuming that the ROI size is 190×90 pixels and the number of the detected road pixels is 11000 pixels (3 ms for the fast color-based segmentation and about 17 ms for the Levenberg-Marquardt minimization). One can notice that this is much faster than the 3D data based approach, which needs 350 ms. Moreover, the Levenberg-Marquardt algorithm is faster than the DE algorithm which needs 120 ms assuming that the number of iterations is 5 and the population number is 30 (the number of pixels is 11000). Obviously, devoting a very small CPU time for estimating the road parameters/camera pose is advantageous for real-time systems since the CPU power can be used for extra tasks such as pedestrian or obstacle detection.

C. Convergence study

In order to study the convergence behavior of the two optimization techniques we used a synthetic stereo sequence containing 1000 stereo frames. For each stereo frame the ground-truth pose parameters are known. Gaussian image is added to the grey-levels of all images. The standard deviation of the noise is kept fixed to 4. For every stereo frame in the sequence the starting solution was shifted from the ground-truth solution by 20 cm for the camera height and by 10 degrees for the plane normal. This shifted solution is used as the starting solution for the Levenberg-Marquardt technique and as the center of the first generation for the Differential Evolution technique. Table I depicts the average height and orientation errors obtained with the LM and DE minimizations. As can be seen, the DE minimization has better convergence properties than the LM minimization which essentially looks for a local minimum. This is reason why the two proposed tracking schemes use the DE algorithm for estimating the pose parameters associated with the first video frame (See Figure 4).

1000 stereo frames	LM	DE
Ave. height error (%)	26.6	3.5
Ave. orientation error (degrees)	10.9	0.41

TABLE I
AVERAGE CAMERA POSE ERRORS.

V. CONCLUSION

A featureless technique for real time estimation of on-board stereo head pose has been presented. The method adopts a registration scheme that uses images' brightness. The advantages of the proposed technique are as follows. First, it does not need any specific visual feature extraction neither in the image domain nor in 3D space. Second, it is very fast compared to almost all proposed stereo-based techniques. The proposed featureless registration is carried out using two optimization techniques: the Differential Evolution algorithm (a stochastic search) and the Levenberg-Marquardt algorithm (a directed search).

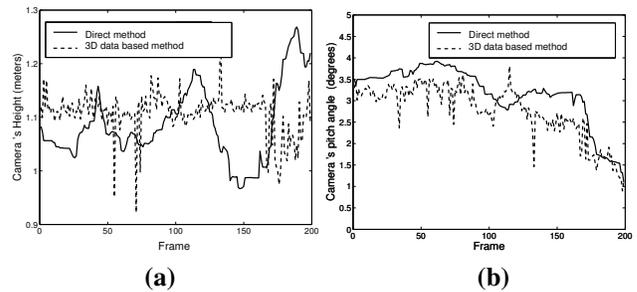


Fig. 11. Camera height and orientation using two different methods.

REFERENCES

- [1] X. Liu and K. Fujimura, "Pedestrian detection using stereo night vision," *IEEE Trans. on Vehicular Technology*, vol. 53, no. 6, pp. 1657–1665, November 2004.
- [2] G. Stein, O. Mano, and A. Shashua, "A robust method for computing vehicle ego-motion," in *IEEE Intelligent Vehicles Symposium*, 2000.
- [3] 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 Intelligent Vehicles Symposium, Versailles, France*, June 2002, pp. 646–651.
- [4] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, 2005.
- [5] G. Toulminet, M. Bertozzi, S. Mousset, A. Benschrair, and A. Broggi, "Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis," *IEEE Trans. on Image Processing*, vol. 15, no. 8, pp. 2364–2375, 2006.
- [6] U. Franke, D. Gavrila, S. Görzig, F. Lindner, F. Paetzold, and C. Wöhler, "Autonomous driving approaches downtown," *IEEE Intelligent Systems*, vol. 13, no. 6, pp. 1–14, 1999.
- [7] T. Suzuki and T. Kanade, "Measurement of vehicle motion and orientation using optical flow," in *IEEE Intelligent Vehicles Symposium*, 1999.
- [8] M. Bertozzi, E. Binelli, A. Broggi, and M. Del Rose, "Stereo vision-based approaches for pedestrian detection," in *Procs. Computer Vision and Pattern Recognition*, San Diego, USA, June 2005.
- [9] R. Labayrade and D. Aubert, "A single framework for vehicle roll, pitch, yaw estimation and obstacles detection by stereovision," in *Proc. IEEE Intelligent Vehicles Symposium, Columbus, OH, USA*, June 2003, pp. 31–36.
- [10] Z. Hu and K. Uchimura, "U-V-Disparity: An efficient algorithm for stereovision based scene analysis," in *Procs. IEEE Intelligent Vehicles Symposium*, Las Vegas, USA, June 2005, pp. 48–54.
- [11] A. Sappa, D. Gerónimo, F. Dornaika, and A. López, "On-board camera extrinsic parameter estimation," *Electronics Letters*, vol. 42, no. 13, pp. 745–747, June 2006.
- [12] O. Faugeras and Q. Luong, *The Geometry of Multiple Images*. The MIT Press, 2001.
- [13] P. Lombardi, M. Zanin, and S. Messelodi, "Switching models for vision-based on-board road detection," in *IEEE Intelligent Transportation Systems*, 2005.
- [14] P. Jansen, W. v. d. Mark, J. v. d. Heuvel, and F. Groen, "Colour based off-road environment and terrain type classification," in *IEEE Intelligent Transportation Systems*, 2005.
- [15] K. V. Price, J. A. Lampinen, and R. M. Storn, *Differential Evolution: A Practical Approach To Global Optimization*. Springer, 2005.
- [16] S. Das, A. Konar, and U. Chakraborty, "Two improved differential evolution schemes for faster global search," in *Genetic and Evolutionary Computation*, 2005.
- [17] R. Storn and K. Price, "Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.
- [18] R. Fletcher, *Practical Methods of Optimization*. New York: Wiley, 1990.
- [19] W. H. Press, S. A. Teukolsky, W. T. Wetterling, and B. P. Flannery, *Numerical Recipes, The Art of Scientific Computing*. New York: Cambridge University Press, 1992.