An experiment of a 3D real-time robust visual odometry for intelligent vehicles

Sergio Alberto Rodriguez Florez, Vincent Fremont, Philippe Bonnifait

An Experiment of a 3D Real-Time Robust Visual Odometry for Intelligent Vehicles

Sergio A. Rodríguez F.[1,2], Vincent Frémont[1,2], Philippe Bonnifait[1,2]
[1]*Université de Technologie de Compiègne (UTC)*, [2]*CNRS Heudiasyc UMR 6599, France*

***Abstract*—Vision systems are nowadays very promising for many on-board vehicles perception functionalities, like obstacles detection/recognition and ego-localization. In this paper, we present a 3D visual odometric method that uses a stereo-vision system to estimate the 3D ego-motion of a vehicle in outdoor road conditions. In order to run in real-time, the studied technique is sparse meaning that it makes use of feature points that are tracked during several frames. A robust scheme is also employed to reject outliers that are detected on moving objects of the environment. Moreover, efforts have been spent on the real-time implementation of the method. In this article, we describe the key stages of the method: features extraction and tracking, quadrifocal constraints, optimization solver and robustification. Real experiments are reported to compare the performance of this approach with GPS data and 2D-wheel-based odometry.**

***Index Terms*—3D Visual odometry, Quadrifocal tensor constraint, Optical flow.**

I. Introduction

Stereo vision systems are affordable nowadays and provide high information bandwidth. They can serve as the basis for many Intelligent Vehicle (IV) applications involving detection and recognition tasks (for instance, roadsigns, pedestrians, obstacles...). Cameras can be also suitable for 3D odometry since they can provide estimates of the complete 6 DOF of the mobile platform starting from a known pose (position and attitude) [1], [2]. This approach can be complementary to the use of more usual techniques relying on Inertial Measurement Units (IMU) and Wheel Speed Sensors (WSS) subject to wheel slippage [3]. In a multi-sensor context, it can also help in increasing the accuracy and the integrity of the perception system.

In this work, the main question we address is what is the performance of a real-time stereo visual odometric system under quasi-urban road applications? Visual odometry relies on the assumption of a static environment. So, a first goal of this research is to evaluate the impact of a dynamic scene corresponding to a road driving situation where cars, pedestrians, shadows, and lighting changes occur frequently. A second issue deals with real-time implementation. Indeed, real-time visual odometry needs to find a positive balance between the following criteria:

- Dense (all the images pixels) or sparse (features points) strategy. In this last case, how many of them?
- How to manage the key images?
- What is the parametrization and the solver of the underlying optimization problem?
- How to robustify the processing in order to handle the dynamics of scene?

In the sequel, we present the model considered for the stereo vision system (section II). Then the proposed approach which combines the advantages of several simple classical methods in one optimization criterion (stereo vision, optical flow, multi-view geometric constraints) is summarized in section III. The algorithm and the real-time system implementation are then presented (section IV). We report a real experiment and we compare the performance of the optical odometry to a proprioceptive odometry that exploits speed and yaw rate measurements from the CAN bus of the vehicle. The results are also compared to GPS data (see section V).

Figure 1. The stereo vision system installed on board the experimental vehicle. In the left-down section, a zoomed view of the stereo vision system

II. Stereo Vision system

The model considered for carrying out our tests, is a stereo vision system composed by two projective camera models rigidly joined, aligned in the x-axis direction and separated by baseline b as illustrated in Figure 2. This aligned configuration of the cameras composing the stereo vision system aim to speed up the stereo association process which will be detailed in section (III-A). For obtaining this configuration in real conditions, it is necessary to use classical stereo image rectification methods [4], [5], [6].

A. Projective Camera Model

A classical pinhole model is considered for the sensor representation. Let be K, the intrinsic calibration matrix of each camera in the stereo system (see Figure 2)

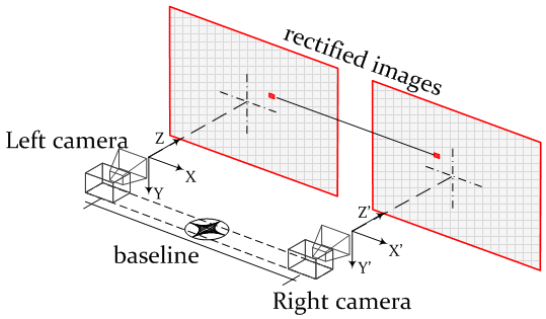

Figure 2. Stereo vision model

$$K = \begin{pmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1)$$

where is f the focal length of the camera in pixels units and $[u_0,\ v_0]^T$ the image coordinates of the principal point. The principal point is considered to be the centre of the image.

III. 3D Visual Odometry Method

A. Features Extraction and Stereo Matching

The first step of the implemented method consists in extracting n features by image of the stereo image pair denoted as $p^*_{l_n}$and $p^{*'}_{r_n}$at t, $\forall\ ^l n <= n$ and $^r n <= n$. These features points corresponds to the pixel coordinates of maximum convolution response between image and predefined masks. We use Speeded-Up Robust Features [7] for repeatability, robustness and stability on the stereo matching and tracking process. Once the features are extracted, a sparse stereo association process [8] is performed. The sparse stereo matching consists in associating features which validate the following constraints in the both directions (left image features with respect to the right ones and *vice versa*):

- The two features must validate the epipolar constraint [9]
- An upper and lower disparity threshold
- The ZNCC (Zero-Mean Normal Cross Correlation) correlation [10] value, estimated in a window of $N \times N$ (typically $N = 9$ to 11) pixels around the features which must be sufficiently higher to avoid ambiguous matching. It is obtained as follows:

$$ZNCC = \frac{\sum_{i,j=1}^{N,N} \left[I_l(i,j) - \bar{I}_l\right]\left[I_r(i,j) - \bar{I}_r\right]}{\sqrt{\sum_{i,j=1}^{N,N} \left[I_l(i,j) - \bar{I}_l\right]^2 \sum_{i,j=1}^{N,N} \left[I_r(i,j) - \bar{I}_r\right]^2}} \quad (2)$$

where I_l and I_r are left and right grayscale image features to be matched and $\bar{I}_l$,$\bar{I}_r$ are respectively the left and right mean intensity values of the images I_l and I_r.

B. Feature Tracking

The tracking stage has the purpose of measuring the 2D image displacement of the stereo matched features between two sampling times i.e. t and $t+1$. The advantage of using a tracking algorithm is to accelerate and to simplify the association process of the features in an interval of time. This is possible by avoiding a feature extraction step by each stereo image sample. For this, a classical Lucas-Kanade tracking (for short LK tracking) method [11] lets us measure the image position of the features in $t+1$ by minimizing the following error function [12]:

$$\epsilon(\mathbf{v}) = \sum_{i=x-N/2}^{x+N/2} \sum_{j=y-N/2}^{x+N/2} \left(I_t(i,j) - I_{t+1}(i+v_x, j+v_y)\right)^2 \quad (3)$$

where $\mathbf{v} = [v_x\ v_y]^T$ is the optical flow vector, I_t is the grayscale image feature to be tracked centered at (x,y) and I_{t+1} is the grayscale image feature which position is estimated as the minimum of the function. Thus, LK tracking is performed independently in the left and the right image, without any stereo and scene rigidity constraint. Then a sparse stereo association process (also known as cross-consistency check) [13], [14] is done with the tracked features obtained. Finally, we keep only the tracked features that are associated in $t+1$ which were also associated in the same way in t.

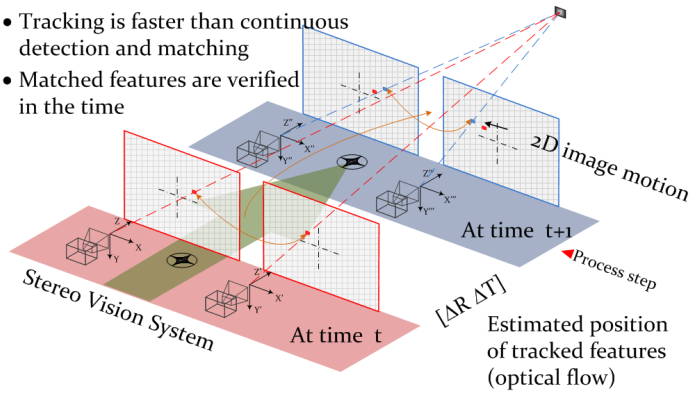

Figure 3. 2D Feature tracking (LK)

C. Quadrifocal Parametrization

The 3D trajectory of a stereo vision system can be estimated between two succeeding stereo image pairs (i.e. ego-motion) as the estimated relative motion of the tracked features. This is only possible if the tracked features lying in a rigid scene (i.e. scene with non moving objects). This constraint of rigidity can be parametrized by using a quadrifocal tensor [15] as stated by Comport et al. in [1]. By simplicity, the quadrifocal tensor can be decomposed into two trifocal tensors, which let us transfer features from the view at time t into the view at $t+1$ under the rigidity constraint [16]. This function is known as the stereo warping operator:

$$\begin{bmatrix} \hat{p} \\ \hat{p}' \end{bmatrix} = \begin{bmatrix} p^* l'_j\, {}^l\mathcal{T}_i^{jk} \\ p'^* l_j\, {}^r\mathcal{T}_i^{jk} \end{bmatrix} \quad (4)$$

where

l'_j and l_j are the lines passing through p'^* and p^* perpendicular to the epipolar line respectively. ${}^l\mathcal{T}_i^{jk}$, is the trifocal tensor composed by the stereo image pair at time t and the

left image at time $t+1$ (i.e. cameras 1, 2, 3 in Figure 4). The second tensor, ${}^{r}\mathcal{T}_i^{jk}$, is composed by the stereo image pair at time t and the right image at time $t+1$ (see cameras 2, 1, 4 in Figure 4) .

As presented in [9], the computation of the trifocal tensor is given by:

$$\mathcal{T}_i^{jk} = a_i^j b_4^k - a_4^j b_i^k \tag{5}$$

where

$P = [I|0]$, $P' = [a_j^i]$ and $P'' = [b_j^i]$ are the canonical 3×4 camera matrices. These camera matrices are computed with respect to a camera reference. The reference for the trifocal tensor ${}^{l}\mathcal{T}_i^{jk}$, is the camera 1 and for ${}^{r}\mathcal{T}_i^{jk}$ is the camera 2 illustrated in Figure 4.

In Eq. (5), we can observe that trifocal computation is dependent on relative camera positions. Knowing that the stereo vision system is rigidly fixed, the only unknown parameters for transferring the feature points extracted at time t into $t+1$ is the rigid transformation (i.e. $\mathbf{R}$, rotation and $\mathbf{t}$, translation) defined previously as the ego-motion.

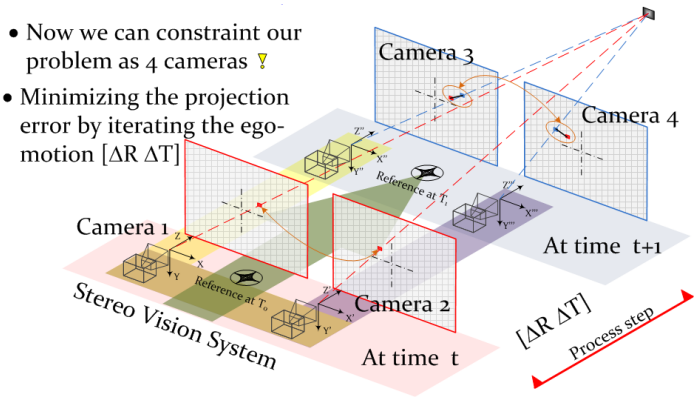

Figure 4. Trifocal constraints

D. Ego-motion estimation

Based on the warping function (see. Eq. 4) and the tracked features (optical flow), the ego-motion estimation can be resolved as an optimization problem. Therefore, we propose to estimate the 6 degrees of freedom the 3D trajectory of the stereo vision system by minimizing a non-linear objective function which represents the error between the measured motion obtained by the tracked features and the estimated motion obtained by the warped features. This error function is minimized by using a Levenberg-Marquard Algorithm (LM)[17], [18], [19].

$$\epsilon = \sum_{i=1}^{k} \mathbf{W} \left[||p_i - \hat{p}_i|| + ||p'_i - \hat{p'}_i|| \right] \tag{6}$$

where p_i and p'_i are the left and right tracked features at time $t+1$ respectively. $\hat{p}_i$ and $\hat{p'}_i$ are the left and right features at time t warped by the estimated motion (see Eq. 5). $\mathbf{W}$ is the weighting matrix estimated by a M-estimator function (in our experiments *Beaton and Tukey* function) [20], [21].

Taking in to account that the ego-motion estimation can be noisy and drifted by errors coming from stereo matching, tracking and by conditions when the scene is not completely rigid like urban environments. Thus, a robust M-estimator function has been implemented on an iteratively re-weighted least square (IRLS) [20] loop for coping with this errors. The robust function let us estimate for each iteration the corresponded weight for all the observations. This estimation is robust under the hypothesis that a larger quantity (more than 50%) of features points are lying on static objects and only an small quantity of them corresponds to matching and tracking error and moving objects.

IV. 3D Visual Odometry Algorithm

Algorithm 1 Real-time Visual Odometry Technique

1: **while** acquiring **do**
2: ▶ Acquire a new stereo view pair
3: **if** tracking **then**
4: ▶ LK tracking of reference view features $\{p, p'\}_r$ where r is the number of tracked feature pairs
5: **if** Minimum tracked point pairs threshold$< r$ **then**
6: ▶ Initialization of the LM algorithm with $x_0 = [\mathbf{\Delta R_0}\ \mathbf{\Delta t_0}]^{\mathbf{T}}, \forall\ x_0\ \in\ \mathbb{R}^6$
7: **repeat**
8: ▶ Estimate $x = [\mathbf{\Delta R}\ \mathbf{\Delta t}]^{\mathbf{T}}, \forall\ x\ \in\ \mathbb{R}^6$ by using LM algorithm $x\ =\ \min(\epsilon)$, see Eq.(6)
9: ▶Compute $\mathbf{W}$ using LM minimization residuals (Re-weigthing process)
10: ▶ $\delta x \Leftarrow ||x_i - x_{i-1}||$
11: **until** ($\delta x >$ Threshold) AND (IterMaxThreshold) AND (Outlier Percent $< 50\%$) //IRLS Loop
12: **else**
13: ▶ Re-initialises algorithm, go to step 3
14: **end if**
15: **end if**
16: **if** initialization **then**
17: ▶ Extraction of Reference View Features in left p^* and right $p^{*'}$ images {i.e. SURF features}
18: ▶ Stereo matching process $\{p^*, p^{*'}\}_n$
19: **end if**
20: Swap all t and $t+1$ variables
21: **end while**

V. Experimental Results

We present the evaluation tests carried out in order to estimate the accuracy and the integrity of the presented method in simulated and real conditions. The method was implemented in C/C++ using Intel OpenCV platform [22] and the *levmar* implementation [23] for non linear minimization. The performance illustrated has been obtained in a Intel Core 2 CPU 2.1 GHz running under Windows XP OS.

A. Synthetic data

Along the development of this work, it was not very easy to find an error function which let us to calculate a good motion estimation by ensuring an affordable time of convergence for real time execution. For this, we carried out a test using simulated data in similar quasi-urban conditions (for instance, acquisition frequency, image resolution, 3D motion and dynamic objects).

The considerations which were taken into account in the simulation model correspond to a stereo pair image resolution of 320×240 pixels at an acquisition frequency of 15fps. In order to generate the synthetic stereo images (see Figure 5, upper stereo image pair), the 3D camera motion was estimated as a function of the vehicle velocity curve and the camera acquisition frequency. Thus, the maximum 3D ego-motion applied was of 0.92 cm and a minimum of 0.27 cm simulating speeds of 50 Km/h and 15 Km/h respectively.

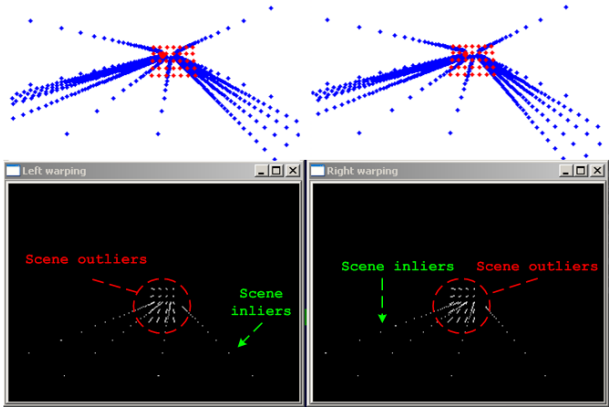

Figure 5. Simulation test with outliers

The simulation test was performed by including 20% of features coming from moving objects in the simulated environment. The results which are illustrated in Figure 5 (see lower image pair) show that the algorithm converges into a motion solution which minimizes the optical flow error generated by the static environment.

In Figure 6 (top) is illustrated the trajectory results obtained under ideal conditions in image feature points extraction and stereo matching steps. Then, some random stereo-feature mismatching and a pixel σ noise was added obtaining the trajectory shown in Figure 6 (bottom).

The error time evolution (see Figure 7) for this last trajectory let us observe a drift generated particularly in sections where there is an important rotation motion. This is because of important ego-motions need more iterations for convergence. The number of iterations is constrained because of real-time execution.

B. Real time results

Experiments using real data were carried out thanks to the experimental platform of the Heudiasyc laboratory (see

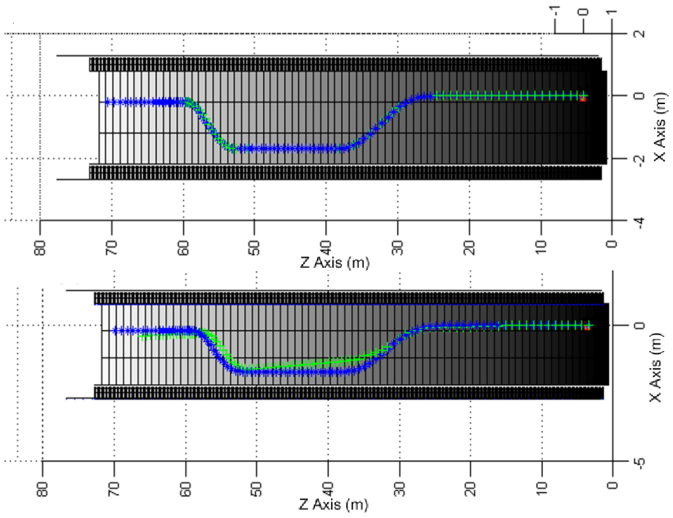

Figure 6. Trajectory obtained in simulated conditions

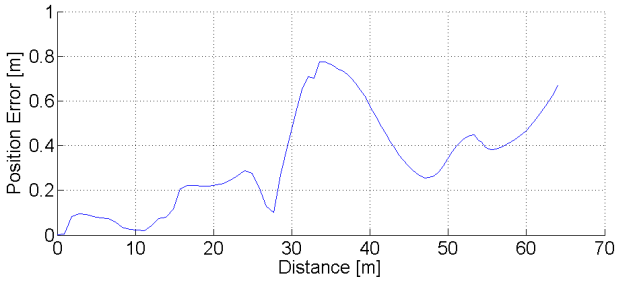

Figure 7. Error time evolution in simulation

Figure 1). This vehicle is equipped with a 47cm-baseline Videre Stereo Vision System installed at the top. This system is composed by CMOS cameras with 4.5mm lens that were set to acquire gray-scale images with a resolution of 320×240 pixels at a frequency of 15 fps. The video sequence takes place in a quasi-urban environment characterized by moving objects (i.e. vehicles, cycles, pedestrians), buildings and trees. The vehicle's velocity was less than 60 Km/h. During the experiment GPS and proprioceptive data were also acquired in parallel by the same application.

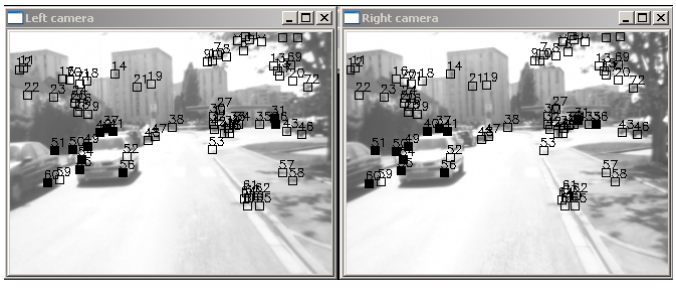

Figure 8. Features points during real-time odometry estimation, ■ : Outliers □ : Inliers

In Figure 8, the indexed stereo feature point pairs which

are classified as inliers (i.e. empty squares) for ego-motion estimation are shown. It is also worth to mention it, that almost all the stereo feature points lying in the moving vehicle are classified as outliers (i.e. filled squares) by the robust function.

As can be noticed in Figure 9, some objects like tree's leaves are an important source of mismatching. This reveals some complex scenes of the test. However, we have remarked that the rejection of moving objects performs quite well.

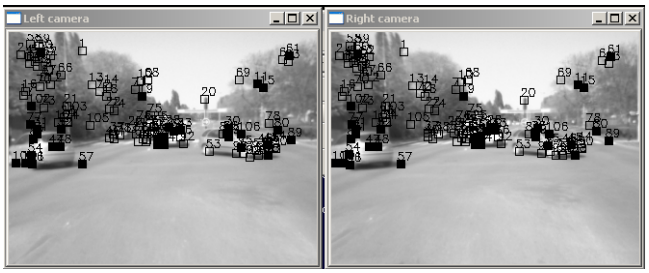

Figure 9. Feature points lying in moving objects ■ : Outliers □ : Inliers

No ground truth localization system was available during this experiment, particularly for the attitude estimates. Therefore, we report here only the comparison of the visual odometry trajectory with a Septentrio PolaRx2c GPS receiver (Figure 10 and 13) and proprioceptive sensors (IMU-WSS) 2D Odometry (Figure 10). As the key characteristic of any dead-reckoning system is its drift with respect to the traveled distance, the low accuracy (absolute error) of the GPS used here is not a matter (the GPS was used in differential SBAS mode - Satellite Based Augmentation System): the atmospheric effects that often bias the pseudo-ranges are subject to slow variation. So, the precision (relative error) is very good in open sky conditions (less than one meter typically). Unfortunately, it remains multi-path effects and satellites outages than introduce jumps. Therefore, using a GPS for comparison is meaningful keeping in mind that this approach doesn't allow estimating the real errors, only differences.

The visual odometry position was initialized using the first GPS point and the attitude using the median of the first 50 points heading. With a 790 m long trajectory, 3.9% 2D-odometry (x-z plane) and 0.25% vertical (y-axis direction) drifts were obtained by using the visual odometry. These differences correspond to the ratio of the position error over the total traveled distance. The 2D-odometry difference evolution is shown in Figure 11.

These results highlight a quite good visual odometry drift which is caused principally by scenes where an important quantity of features lying in vehicles exceed the features points lying in the static scene. This is one of the main drawback of the sparse approach.

The experiment with real data has also shown situations where very few points were tracked in the process. Please note that this situation which didn't appear in the simulated conditions. Since this phenomenon is a crucial issue in real conditions. Figure 10 shows in green crosses when the visual odometry is reinitialized for re-extracting new features points. As expected, the algorithm re-initializes frequently in turns.

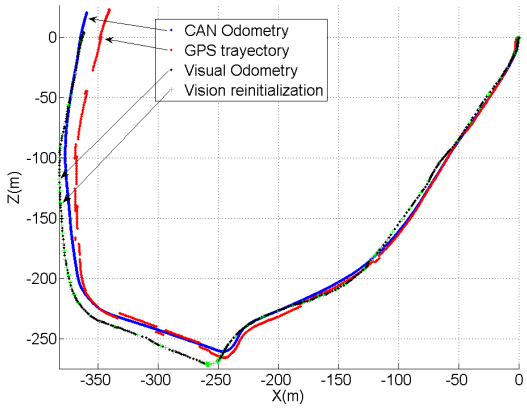

Figure 10. 2D estimated trajectory using GPS, proprioceptive sensors and visual odometry

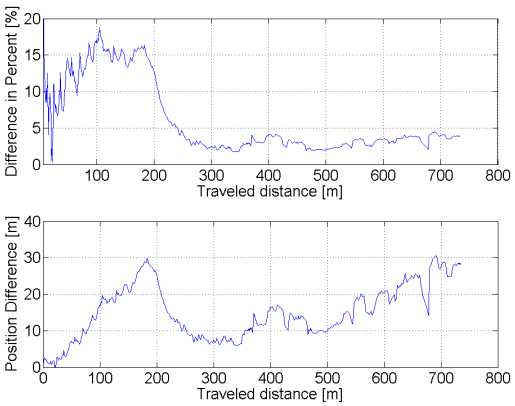

Figure 11. Difference evolution between GPS and Visual Odometry

Figure 12 shows a zoomed part of the trajectory reconstructed in Figure 10, highlighting a drift start. By observing the left stereo corresponding image, one can see that some features points present a sliding motion (left sidewalk) and poor texture surfaces (for example reflective pattern on top-right building) induce a bias on the optical flow. In addition, there is not an uniform feature dispersion in the image.

If one can conclude that visual odometry doesn't improve the 2D accuracy compared to WSS odometry on good quality roads, it should be noticed that it provides the full 6 degrees of freedom which is not possible to obtain using wheel-based odometry. Figure 13 plots the 3D estimated trajectory obtained using visual odometry and GPS. The GPS jumps are particulary visible in this plot are mainly due to the satellites changes. One can notice that the 3D trajectory obtained using the visual odometry is quite smooth. This an interesting feature for integrity monitoring since a smoothed prediction of the pose is crucial to eliminate GPS outliers. Finally, the altitude drift is very small (less than 0.3% of the traveled distance) which is a very nice feature.

Finally, Figure 14 presents a time execution histogram of the algorithm revealing that convergence time is not constant. This is caused by the non homogeneous vehicle motion and by

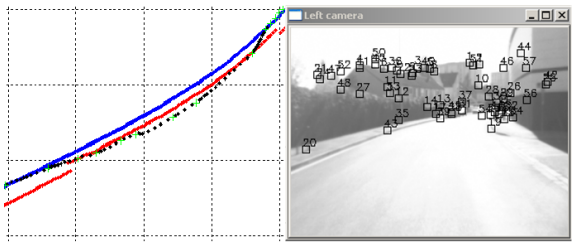

Figure 12. Observed drift caused by critical conditions

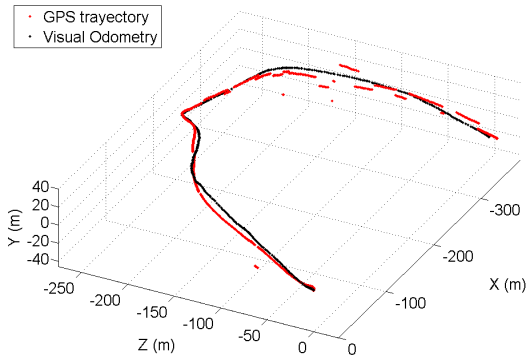

Figure 13. 3D estimated trajectory (local frame coordinates)

the variability of scene complexity (i.e. outliers/inliers ratio). However, approximately a half of the ego-motion estimations has been performed in less than 100ms which illustrates a good real-time implementation.

VI. Conclusion

A real-time visual odometry approach has been presented and experimentally studied. The core of the method combines in one non linear criterion the ego-motion estimation based on sparse optical flow and quadrifocal tensor warping. An experiment under quasi-urban conditions illustrates the good performance of the optical odometry with respect to GPS and proprioceptive ones. The obtained real-time results show a good trade-off between precision and execution time thanks to a sparse feature approach. However, this experiment revealed that an important degradation source of the visual odometry performance is due to high rotational speed mouvements like 90° turns and roundabouts. The management of these critical conditions and the reduction of the drift in a multi-sensor context constitute the main perspective of this research.

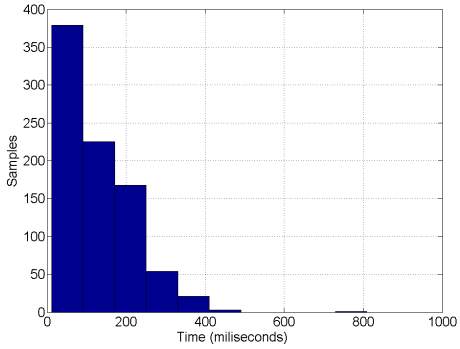

Figure 14. Computation time histogram for the complete sequence

References

[1] A. Comport, E. Malis, and P. Rives, "Accurate quadrifocal tracking for robust 3d visual odometry," *IEEE International Conference on Robotics and Automation*, pp. 40–45, April 2007.

[2] W. van der Mark, D. Fontijne, L. Dorst, and F. C. Groen, "Vehicle ego-motion estimation with geometric algebra," *Proceedings IEEE Intelligent Vehicle Symposium, Versialles*, vol. 1, pp. 18–20, 2002.

[3] M. Agrawal and K. Konolige, "Real-time localization in outdoor environments using stereo vision and inexpensive gps," *18th International Conference on Pattern Recognition, 2006. ICPR 2006.*, vol. 3, pp. 1063 – 1068, 2006. [Online]. Available: http://www.ai.sri.com/ agrawal/icpr06.pdf

[4] A. Fusiello, E. Trucco, and A. Verri, "A compact algorithm for rectification of stereo pairs," *Machine Vision and Applications Manuscript*, vol. 1, pp. 1–8, 1999.

[5] R. Y. Tsai, "A versatile camera calibration techniaue for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 4, 1987.

[6] Z. Zhang, "A flexible new technique for camera calibration," *IEEE T. Pattern. Anal.*, vol. 22, no. 11, pp. 1330–1334, 2000.

[7] H. Bay, A. Ess, T. Tuytelaars, , and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding (CVIU)*, vol. 110, pp. 346–359, 2008.

[8] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.

[9] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision. Second Edition*, C. U. Press, Ed. Cambridge, 2003.

[10] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision*, Springer, Ed. Springer, 2005.

[11] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proceedings of the 1981 DARPA Imaging Understanding Workshop*, vol. 1, pp. 121–130, 1981.

[12] J.-Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker description of the algorithm," Intel Corporation Microprocessor Research Labs, Tech. Rep., 2002.

[13] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 652–659, 2004.

[14] ——, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, 2006.

[15] A. Shashua and L. Wolf, "On the structure and properties of the quadrifocal tensor," in *ECCV (1)*, 2000, pp. 710–724. [Online]. Available: http://citeseer.ist.psu.edu/shashua00structure.html

[16] C. Ressl, "Geometry, constraints and computation of the trifocal tensor," Phd Thesis Dissertation, Technische Universität Wien, 2003, wolfgang Förstner AND Karl Kraus.

[17] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.

[18] D. Marquardt, "An algorithm for the least-squares estimation of non-linear parameters," *SIAM Journal of Applied Mathematics*, vol. 11, pp. 431–441, 1963.

[19] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3th Edition*, C. Press, Ed. Cambridge Press, 2007.

[20] C. V. Stewart, "Robust parameter estimation in computer vision," *Society for Industrial and Applied Mathematics*, vol. 41, no. 3, pp. 513–537, 1999.

[21] A. Comport, M. Pressigout, E. Marchand, and F. Chaumette, "A visual servoing control law that is robust to image outliers," *Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, 2003.

[22] G. Bradski and A. Kaehler, *Learning OpenCV, Computer Vision with OpenCV*, O'Reilly, Ed. http://sourceforge.net/projects/opencvlibrary: O'Reilly, 2008.

[23] M. Lourakis. levmar: Levenberg-marquardt nonlinear least squares algorithms in c/c++. http://www.ics.forth.gr/ lourakis/levmar/.