

Camera Egomotion Estimation in the ADAS Context

D. Cheda, D. Ponsa and A.M. López

Abstract—Camera-based Advanced Driver Assistance Systems (ADAS) have concentrated many research efforts in the last decades. Proposals based on monocular cameras require the knowledge of the camera pose with respect to the environment, in order to reach an efficient and robust performance. A common assumption in such systems is considering the road as planar, and the camera pose with respect to it as approximately known. However, in real situations, the camera pose varies along time due to the vehicle movement, the road slope, and irregularities on the road surface. Thus, the changes in the camera position and orientation (i.e., the egomotion) are critical information that must be estimated at every frame to avoid poor performances. This work focuses on egomotion estimation from a monocular camera under the ADAS context. We review and compare egomotion methods with simulated and real ADAS-like sequences. Basing on the results of our experiments, we show which of the considered nonlinear and linear algorithms have the best performance in this domain.

I. INTRODUCTION

Currently, many research efforts are being done towards the use of cameras to develop Advanced Driver Assistance Systems (ADAS). The richness of the information provided by images (texture, color, etc.) as well as their high resolution and relevant issues such as ease of integration (even in small vehicles), low cost and low power consumption, make cameras a very appealing option for sensing the driving environment. However, the application of computer vision techniques to develop ADAS systems like active cruise control or collision avoidance is challenging, due to their stringent time-response requirements. The problem commonly comes from the computation required to process images, which is very high when generic strategies are applied. Hence, when transferring computer vision techniques to the ADAS domain, it is essential to employ prior knowledge of the driving context for the sake of efficiency. In practice, what many proposals do is taking advantage of the fact that the road is commonly planar, and that the camera pose with respect to it is approximately known. In tasks such as vehicle and pedestrian detection, this allows to discard the processing of a significant image part (e.g., the sky region), and then to inspect the remaining part taking into account perspective effects. The main drawback of this approach is that, despite the road can be assumed planar in most cases, the camera pose with respect to it is not constant over time. Indeed, due to the vehicle movement, the road slope, and irregularities on the road surface, the geometric relation between the camera and the road varies every frame. Hence, considering it constant leads to suboptimal image analysis, and therefore

D. Cheda, D. Ponsa and A.M. López are from Computer Vision Center, Autonomous University of Barcelona, 08193 Bellaterra, Barcelona, Spain. Email: {dcheda, daniel, antonio}@cvc.uab.es.

to poor system performances. A principled solution to tackle this problem is estimating the camera pose at every frame.

There exist several proposals to do it that are based solely on the information available in images, avoiding the need of installing additional sensors on the vehicle. For instance, a stereo ADAS system is proposed in [1] to estimate the change in camera pose over time by aligning the 3D data recovered from stereo-vision at consecutive time instants. In a different way, in [2] the camera pose with respect to the road is directly recovered at each instant by fitting a plane to the recovered 3D data. The normal of that plane encodes implicitly the pose of the stereo camera with respect to it. For monocular systems, recovering the camera pose is far more challenging. One possibility is to take advantage of knowledge of some structure present in the image. This is the approach followed in [3], where by fitting a 3D model of the road lane markings to the image, the camera pose is recovered. However, the requirement of observing known structures strongly limits the applicability of this strategy. A convenient way to avoid this requirement is using general methods for determining the camera pose variation along time. This is, in fact, a classical problem in computer vision, commonly referred as the camera egomotion estimation problem. Different proposals exist in the literature, and in this paper we evaluate the performance of some representative algorithms in ADAS-like sequences. This is important since the performance of egomotion algorithms vary significantly depending on the structure of the observed scene. An alternative to general egomotion estimation methods is applying structure from motion algorithms, as in [4]. However, in our case we want to estimate the egomotion avoiding an explicit reconstruction of the observed scene.

The paper is organized as follows. Section II formalizes the egomotion estimation problem, and describes the main strategies proposed to solve it. The experimental work to assess their performance is described in Section III, where the results obtained are also discussed. Finally, the paper presents the conclusions of our work.

II. PROBLEM STATEMENT

The egomotion problem concerns the estimation of the 3D rigid motion (rotation and translation) of a camera from an image sequence acquired by it. That is, determining the variation of the six degrees of freedom (DOF) corresponding to the camera extrinsic parameters. Based on the kind of information used, we distinguish three strategies for egomotion estimation: discrete, continuous, and direct methods. Discrete methods rely on information extracted from a sparse set of feature correspondences between two distinct camera

viewpoints, both related by a rotation \mathbf{R} and translation \mathbf{t} [5]. This information is described through the so-called epipolar constraint [6], which is encoded into the well-known essential matrix \mathbf{E} [7]. Specifically, if $\mathbf{q} = [q_x, q_y, 1]$ and $\mathbf{q}' = [q'_x, q'_y, 1]$ are the homogeneous coordinates of 2D matched points between two different views, then the following relationship holds

$$\mathbf{q}'^T \mathbf{E} \mathbf{q} = 0, \quad (1)$$

where $\mathbf{E} = \mathbf{R} [\mathbf{t}]_{\times}$ is the essential matrix relating such views. With $[\mathbf{t}]_{\times}$ we denote the skew symmetric matrix of \mathbf{t} . Although both \mathbf{R} and \mathbf{t} have 3 DOF, the essential matrix \mathbf{E} has only 5 DOF, since there is a scale ambiguity in the magnitude of the translation \mathbf{t} . \mathbf{R} and \mathbf{t} are computed by decomposing \mathbf{E} once it is estimated, existing different proposals in the bibliography to do that. When the camera calibration is unknown, a relation analogous to (1) is described through the fundamental matrix [8].

The epipolar constraint does not take into account any prior information about the relation of the two views put into correspondence. However, as it happens in the case of a video sequence, if these two views are the result of camera motion during a small time interval, the change of the camera pose can be considered as theoretically infinitesimal. From that, a continuous version of the epipolar constraint can be formulated.

Given a static 3D point $\mathbf{p} = [p_x, p_y, p_z]$ in the scene, the relative movement of the camera with respect to it can be described by its linear velocity $\dot{\mathbf{t}} = [\dot{t}_x, \dot{t}_y, \dot{t}_z]$ and its angular velocity vector $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]$. If we denote the projection of \mathbf{p} in the image plane by \mathbf{q} (in homogeneous coordinates), the change of this projection due to the camera motion $(\dot{\mathbf{t}}, \boldsymbol{\omega})$ corresponds to $\dot{\mathbf{q}} = [\dot{q}_x, \dot{q}_y, 0]$, whose components are determined by the following expression¹

$$\begin{bmatrix} \dot{q}_x \\ \dot{q}_y \end{bmatrix} = \begin{bmatrix} \frac{\dot{t}_x - \dot{t}_z q_x}{p_z} + (\omega_y - \omega_z q_y - \omega_x q_x q_y + \omega_y q_x^2) \\ \frac{\dot{t}_y - \dot{t}_z q_y}{p_z} + (-\omega_x + \omega_z q_x + \omega_y q_x q_y - \omega_x q_y^2) \end{bmatrix}. \quad (2)$$

The aim of continuous methods [9] is hence determining $(\dot{\mathbf{t}}, \boldsymbol{\omega})$ from the relation between view points and their image flow. Since the image flow field is an ideal concept, it is approximated either by densely computing the optical flow in consecutive frames (i.e., the apparent motion perceived from the changes in image intensity) or, by tracking some sparse interest points [10].

In order to estimate motion parameters, continuous methods apply different manipulations on (2) to annihilate its dependency on depth (i.e., p_z). The following bilinear constraint on $\dot{\mathbf{t}}$ and $\boldsymbol{\omega}$ can be achieved by algebraic operations on (2)

$$(\dot{\mathbf{t}} \times \mathbf{q})^T (\dot{\mathbf{q}} - (\boldsymbol{\omega} \times \mathbf{q})) = 0. \quad (3)$$

Note that (3) does not depend on the position of the point in space, but only on its projection and the motion parameters. This bilinear constraint is used by different linear and

nonlinear methods to compute egomotion from optical flow. By simple algebraic manipulations, (3) is rewritten as

$$\dot{\mathbf{q}}^T [\dot{\mathbf{t}}]_{\times} \mathbf{q} + \mathbf{q}^T [\boldsymbol{\omega}]_{\times} [\dot{\mathbf{t}}]_{\times} \mathbf{q} = 0. \quad (4)$$

This is the so-called differential epipolar constraint [11], due to structural similarity with the constraint (1) of the discrete case.

In contrast to the above methods, direct methods [12] are based on the image brightness constraint to directly recover the motion parameters. These methods integrate the information provided by all the pixels to recover egomotion, without explicitly matching between views as an intermediate step. For this purpose, most of them presuppose a strong assumption with respect to the image brightness constancy, i.e., that there should be no changes in intensities between consecutive images, which is not valid in outdoor scenes where inter-frame brightness variations are significant and time-varying illumination changes are present. However, recently in [13], lighting changes are handled via a proposed photogeometric modeling to explain the illumination variations as an evolving surface. All parameters of such model are simultaneously estimated by a second-order optimization procedure. This model can be used to overcome limitations above mentioned.

Our study is focused just on discrete and continuous methods, which are also denoted in the bibliography as feature-based and optical flow-based methods.

A. Discrete Methods

The basis for feature-based approaches lies in the early work of Longuet-Higgins [6]. Based on the epipolar geometry, it shows how to estimate the relative camera motion between two views from at least eight point correspondences. The method estimates the essential matrix \mathbf{E} relating a pair of calibrated views by solving an overdetermined homogeneous system of linear equations using a least-squares approximation. This eight-point algorithm is linear, fast and easy to implement, but very sensitive to noise (which leads to poor results). However, Hartley [14] shows that, by simply prenormalizing (translation and scaling) point coordinates, the performance significantly improves. By adding extra constraints to the original eight-point formulation, solutions requiring just seven, six, and five point correspondences have been developed [15].

Iterative methods to estimate \mathbf{E} have also been proposed [16]. They rely on determining the essential matrix that minimizes a cost function relating the distance between points and epipolar lines. While these methods are, in general, more accurate, they are also time-consuming, requiring a good initialization and coping with outliers.

The existence of mismatches in point correspondence between views can significantly affect the performance of the previous methods. To avoid that, they have been applied in conjunction with methods for outlier rejection [17].

B. Continuous Methods

Based on the image flow field, different methods have been derived to compute $\dot{\mathbf{t}}$ and $\boldsymbol{\omega}$. From (3), Bruss and Horn [9]

¹Without loss of generality, here we assume a unit focal length.

estimate $\dot{\mathbf{t}}$ by numerical optimization techniques, subjected to be unitary since, as previously stated, translation can only be estimated up to a scale factor. Once $\dot{\mathbf{t}}$ is determined, ω is computed by solving a linear system.

Based also on the bilinear constraint, Heeger and Jepson [18] developed the so-called linear subspace method. From the optical flow sampled at N discrete points in the image, a set of constraint vectors orthogonal to the camera translation velocity $\dot{\mathbf{t}}$ is built. This allows defining an overdetermined linear system to estimate $\dot{\mathbf{t}}$ with $N - 6$ constraints. Note that for N image velocity samples, N bilinear constraints could be defined over $\dot{\mathbf{t}}$ and ω , but only a subset of linear constraints is used. Once the direction of translation is computed, ω is obtained by also solving a linear system of equations. In spite of the linearity, the main disadvantage of this method is that it does not use all the available information to estimate the motion parameters. Similar approaches are followed by Soatto et al. [19], and Zhang et al. [20].

Different algorithms based on the differential epipolar constraint in (4) have also been defined. In [21] Kanatani estimates $\dot{\mathbf{t}}$ and ω from an overdetermined system of equations, which is solved by a least-squares minimization. Since this solution is shown to be systematically biased, Kanatani then proposes to apply a “renormalization” step to subtract an estimate of the output bias from the solution. Based on a study of the differential essential matrix, Ma et al. [22] propose a method conceptually similar to [21], which recovers the 3D velocity of the camera in a more natural form by eigenvector-decomposition. The performance of these and other additional resolution schemes based on this constraint have been evaluated in [16], being the “normalized” Kanatani proposal [21] the best performing one.

III. EXPERIMENTAL ASSESSMENT

In this section, we report the experiments done to quantify the performance of the more relevant methods described in Sec. II using synthetic sequences as well as real sequences. Our objective is to evaluate different algorithms in ADAS-like sequences in order to determine the best performing one. First, we select a set of methods taking into account the characteristics of the dominant camera motion in our context. In normal driving conditions, a vehicle moves mainly forward, and the most important variations in the camera orientation are in the yaw and pitch angles. According to that, and considering methods available on the toolbox given in [23] and others, the egomotion algorithms compared here are selected due to the following reasons:

- Regarding the discrete epipolar constraint, the comparative study fulfilled by Nistér [15] indicates that for sideway motion the five-point algorithm leads to the best estimation with respect to linear, iterative, and robust tested methods. However, for forward motion normalized eight-point algorithm (8pts) [14] overcomes all the methods. Thus, we include just the 8pts algorithm in our study.
- As the representative of the linear algorithms using the differential essential matrix, we test the infinitesimal

TABLE I

INTRINSIC PARAMETERS OF THE USED PINHOLE AND REAL CAMERA.

Parameter	Simulated Camera	Real Camera	Unit
Focal length	810.81	820.428	pixels
Image width	640	640	pixels
Image height	640	480	pixels
Optical center	(320,320)	(305.278,239.826)	pixels

eight-point algorithm (KA) and its renormalized version (KB) of Kanatani proposal [21], since these algorithms are shown, in [16], as the best performing methods of this sort.

- We test the linear subspace method of Jepson and Heeger (J&H) [18] because it deals with the bilinear constraint in a linear fashion, showing comparable results with respect to other linear and nonlinear methods in egomotion estimation [22], [16].
- The optimization-based method of Bruss and Horn (B&H) [9] is selected because in the tests of [23] it exhibited the best performance of all (linear and nonlinear) methods.

A. Evaluation with synthetic data

To quantify the performance of selected algorithms, we adopt a methodology similar to the ones proposed by Tian et al. [23] and Ma et al. [22]. They generate random clouds of 3D points and compute their 2D projections and the image flow vectors corresponding to a particular 3D motion. Then, for each algorithm, egomotion is estimated in one thousand trials, quantifying its accuracy from the dissimilarity observed with respect to the ground truth motion. In our case, instead of random clouds of points, we generate random point configurations similar to the ones obtained from a camera mounted in a car, moving in a typical driving environment. The scene contains approximately 700 points randomly placed on the road, objects, and a plane at infinite. The road length is about 500 meters (m), and over it, we generate very few points, because, in real situations, a road does not have sufficient texture where many points can be detected. We generate the most number of points in objects of different random sizes located on both sides of the road, according to what we experimentally observed. A small number of points are generated in a plane perpendicular to the road, which is placed at infinity, emulating the distant structures observed in real sequences above the horizon line. Occlusions between different elements in the scene are managed by the z-Buffer algorithm.

Once the scene is generated, it is projected onto an image plane by using a pinhole camera model with the intrinsic parameters indicated in Table I. The image flow vectors are produced by a rigid body motion of the camera with a translation of 1 m/frame on Z axis and a rotation of 2° /frame on the X axis. Assuming a frame rate of 25 frames/s, these magnitudes correspond to a real-life situation with a car moving at 90 km/h (i.e., driving in a fast road), with some oscillations in the camera pitch angle, which commonly occurs due to the effect of the suspension system. In Fig. 1

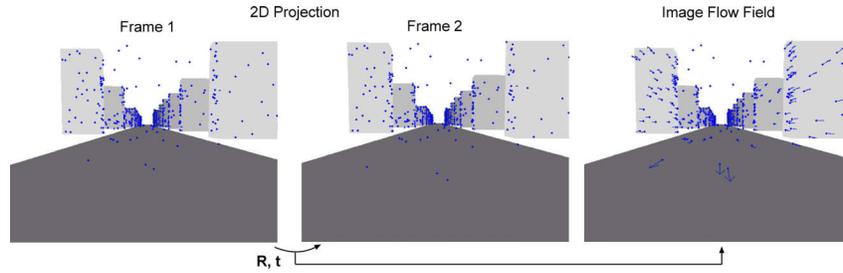


Fig. 1. 3D Scene projection and image flow resulting of the camera motion.

an example of image flow caused by a translational motion over Z axis is shown.

Zero-mean Gaussian noise of various amounts is added to each image flow vector, to simulate errors in the optical flow computation. The considered standard deviations of the Gaussian noise are 0.05, 0.15, 0.25, 0.5, and 0.7 pixels. These values can be seen as the localization accuracy of the point tracked between frames. We selected the maximum noise level as the mean of corner delocalization suffered by the Plessey corner detector [24] in several experiments done.

The generic evaluation criteria proposed in [23] is adopted to absolutely quantify the algorithms accuracy. For each algorithm, we measure the mean error of the estimates in one thousand different scenes. The mean error $\mu_{\hat{\mathbf{t}}}$ in the $\hat{\mathbf{t}}$ estimation is quantified as the angle (in degrees) between the true translation direction \mathbf{t} , and the average of the translation direction $\bar{\mathbf{t}}$ of all trials, that is

$$\mu_{\hat{\mathbf{t}}} = \cos^{-1}(\bar{\mathbf{t}}^T \mathbf{t}) ,$$

since the dot product is $\bar{\mathbf{t}}^T \mathbf{t} = |\bar{\mathbf{t}}||\mathbf{t}|\cos(\alpha)$, and $|\hat{\mathbf{t}}| = |\bar{\mathbf{t}}| = 1$ because the estimated translation is only recovered up to a scale factor.

To quantify the mean rotation error, we use the difference angle between the true rotation $\boldsymbol{\omega}$ and the mean of estimated rotations $\bar{\boldsymbol{\omega}}$. For this purpose, rotation matrices \mathbf{R} and $\bar{\mathbf{R}}$ for both $\boldsymbol{\omega}$ and $\bar{\boldsymbol{\omega}}$ are built. The product between \mathbf{R}^T and $\bar{\mathbf{R}}$ is an identity matrix when both are equal. Thus, the difference between both matrices is defined as $\Delta\mathbf{R} = \mathbf{R}^T \bar{\mathbf{R}}$. $\Delta\mathbf{R}$ can be characterized by an unit axis vector and an angle $\mu_{\mathbf{R}}$. This angle is used as the mean rotation error. Since $\text{trace}(\mathbf{R}) = 1 + 2\cos(\alpha)$, then the angle is equal to

$$\mu_{\mathbf{R}} = \cos^{-1} \left(\frac{1}{2} (\text{trace}(\Delta\mathbf{R}) - 1) \right) .$$

We choose these metrics because they provide a compact error measure that facilitates the evaluation of the algorithms, i.e., a scalar error value for $\hat{\mathbf{t}}$ and $\boldsymbol{\omega}$, respectively. Another option could be the mean of each component vector of the estimated $\hat{\mathbf{t}}$ and $\boldsymbol{\omega}$, but this would make more difficult the performance analysis.

Our results in synthetic ADAS-like sequences are shown in Fig. 2. We can see that the best result is achieved by B&H probably because it entails an iterative optimization process to estimate the parameters. With respect to the remaining linear methods, KB overcomes all other linear approaches

because it deals with the bias in the translation direction. KA, J&H and 8pts are more affected by noise. The 8pts algorithm does not have good results —especially notice its poor results in rotation estimation—, since it is based on discrete epipolar constraint, which works better when the motion between the two images is relatively large, and in the ADAS context the camera motion is small. In our experiments, algorithms show a better performance than the one reported in comparative study done by Tian et al. [23]. This is because in that study motion corresponding to sideway translation was considered, and apparently motion parameters can be more robustly estimated when translation is along Z -axis, which is consistent with the sensitivity analysis done in [25].

B. Evaluation with real sequences

Once compared methods in synthetic data, the next step is checking their performance on sequences acquired by a vehicle moving in a real-world scenario.

We use image sequences from the Environment Perception and Driver Assistance dataset². These sequences have been captured with a calibrated stereo pair (see Table I for its intrinsic parameters) in driving scenarios including highways, urban and rural roads (see [26] for more details). They contain between 250 and 300 frames, and car velocity and yaw rate is provided by an inertial sensor from which an approximation of \mathbf{t} can be computed. Fig. 3 shows some frames of the sequence and the trajectory followed by the vehicle.

Since the dataset just partially provides the ground truth of the camera motion, we need to complete it by providing the camera rotation information. To this end, we use a stereo ego-motion estimation approach similar to the one proposed by Badino [1] to compute the camera motion along a sequence. The interest points are detected by using Shi and Tomasi [27] corner detector applied to the current and previous left image. These points are matched in successive frames by a KLT tracker [28], which determines their optical flow along the sequence. Stereo is computed by triangulation with the tracked points for every frame obtaining 3D points of the environment. Parameters \mathbf{t} and \mathbf{R} are obtained by matching the clouds of 3D points in consecutive frames, using a least-squares closed-form solution based on unit quaternions. Since errors in point correspondences, tracking and/or stereo can generate flow vectors representing inconsistent motion,

²From the enpeda project: <http://www.mi.auckland.ac.nz/>.

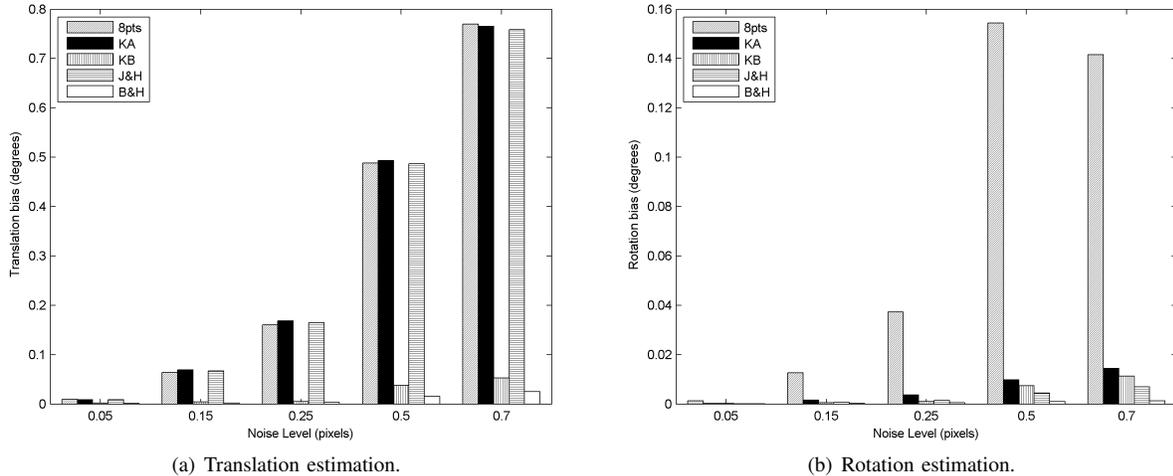


Fig. 2. Results of translation and rotation estimation with synthetic sequences.



Fig. 3. Frames of the real sequence and the trajectory followed by the vehicle which is measured with an inertial sensor.

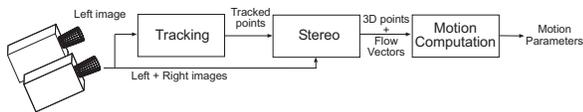


Fig. 4. Stereo egomotion framework proposed by Badino [1] and used to recover motion parameters from stereo sequence.

a robust egomotion estimate is obtained using RANSAC. Then, we take both rotation estimates made with this stereo method and translation estimates from the inertial sensor as a baseline to compare the accuracy of the considered monocular egomotion algorithms. A diagram of this stereo egomotion strategy is shown in Fig. 4.

Fig. 5 summarizes our results along a 300 frames sequence. Table II reports estimation errors using the format of “error mean \pm error standard deviation”. From our observations, B&H notably achieves the best results since it entails an iterative optimization process to estimate the parameters. This is in accord with our previous analysis under simulated environment. However, in contrast to the synthetic experiments, the remaining algorithms show very similar performance in egomotion estimation. Notice that the performance of the 8pts algorithm remains on par with the differential ones, which is probably due to the fact that noise in this sequence is proportional to the amount of performed motion (as have been proved in [29]). We also note that the noise in flows affects the performances of all algorithms

TABLE II
EGOMOTION ESTIMATION IN A REAL DRIVING SCENARIO.

Algorithms	Translation Error	Rotation Error
8pts	$6.89^\circ \pm 4.62^\circ$	$0.28^\circ \pm 0.25^\circ$
KA	$6.46^\circ \pm 4.34^\circ$	$0.24^\circ \pm 0.25^\circ$
KB	$6.86^\circ \pm 5.15^\circ$	$0.24^\circ \pm 0.26^\circ$
J&H	$6.82^\circ \pm 4.54^\circ$	$0.15^\circ \pm 0.26^\circ$
B&H	$3.28^\circ \pm 2.02^\circ$	$0.17^\circ \pm 0.23^\circ$

(especially in translation direction, which is affected by a bias), what can be improved by a subsequent step of bundle adjustment to refine the pose estimate.

IV. CONCLUSIONS

In this work, we review several relevant camera egomotion estimation methods in the literature, and evaluate their performance in the ADAS context. This study is motivated by the relevance, for monocular ADAS systems, of knowing the camera pose with respect to the environment, in order to achieve a reliable and efficient performance. To this end, we compare egomotion methods on synthetic and real ADAS-like sequences to determine those that have better performance. As a conclusion of our synthetic experiments, we show that the best nonlinear and linear performing methods are B&H and KB, respectively. Regarding real sequence experiments, the best performance is reached by B&H while other algorithms perform less accurately and at a similar level. Unexpectedly, we observe that 8pts behaves similar

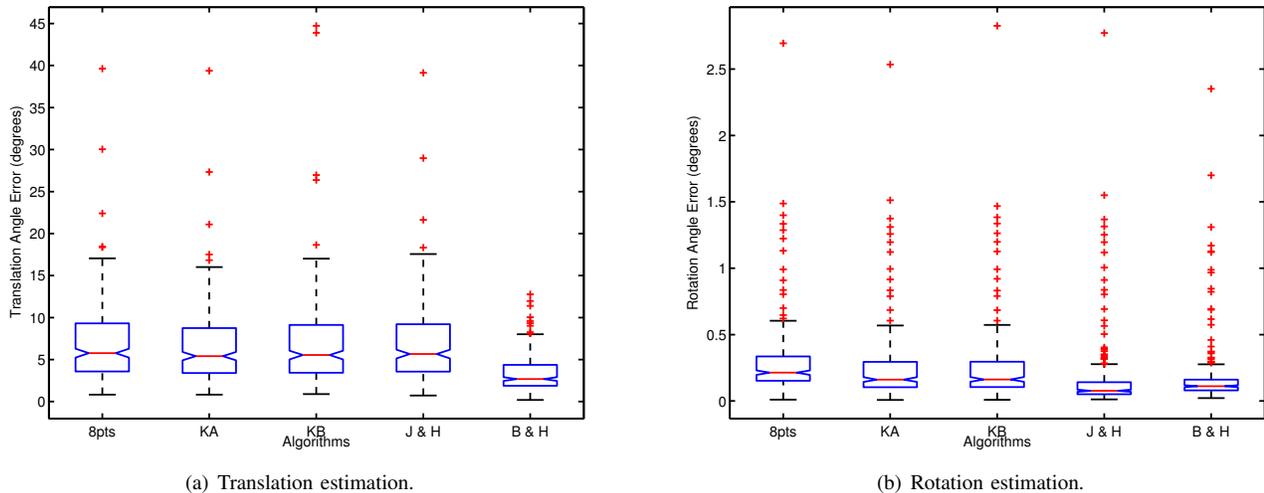


Fig. 5. Results of translation and rotation estimation in real sequences.

to continuous methods, probably because the noise in flows is proportional to the motion in the analyzed sequence. Also, we note that translation estimation is affected by a bias. Our current work focuses on a deeper study of these two last aspects.

V. ACKNOWLEDGMENTS

This work has been supported by the projects TRA2007-62526/AUT and CTP-2008ITT00001; and research programme Consolider-Ingenio 2010: MIPRCV (CSD2007-00018); and the Autonomous University of Barcelona.

REFERENCES

- [1] H. Badino, "A Robust Approach for Ego-Motion Estimation Using a Mobile Stereo Platform," in *Int. Workshop on Complex Motion*, 2004, pp. 198–208.
- [2] A. D. Sappa, F. Dornaika, D. Ponsa, D. Gerónimo, and A. M. López, "An Efficient Approach to Onboard Stereo Vision System Pose Estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 476–490, 2008.
- [3] D. Ponsa, J. Serrat, and A. M. López, "On-board Image-based Vehicle Detection and Tracking," *Trans. Inst. Meas. Control*, 2009.
- [4] N. Cornelis, B. Leibe, K. Cornelis, and L. V. Gool, "3D Urban Scene Modeling Integrating Recognition and Reconstruction," *Int. J. Comput. Vision*, vol. 78, no. 2-3, pp. 121–141, 2008.
- [5] P. H. S. Torr and A. Zisserman, "Feature Based Methods for Structure and Motion Estimation," in *Vision Algorithms: Theory and Practice*, no. 1883, 2000, pp. 278–295.
- [6] H. C. Longuet-Higgins, "A Computer Algorithm for Reconstructing a Scene from Two Projections," *Nature*, vol. 293, pp. 133–135, 1981.
- [7] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
- [8] O. Faugeras, Q. T. Luong, and T. Papadopolou, *The Geometry of Multiple Images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications*. MIT Press, 2001.
- [9] A. Bruns and B. K. P. Horn, "Passive Navigation," *Comput. Vision, Graphics, and Image Process.*, vol. 21, pp. 3–20, 1983.
- [10] C. McCarthy and N. Barnes, "Performance of Optical Flow Techniques for Indoor Navigation with a Mobile Robot," in *IEEE Int. Conf. on Rob. and Autom.*, 2004, pp. 5093–5098.
- [11] Y. Ma, S. Soatto, J. Kosecká, and S. Sastry, *An Invitation to 3D Vision: From Images to Geometric Models*. Springer-Verlag, 2003.
- [12] M. Irani and P. Anandan, "About Direct Methods," in *Int. Workshop on Vision Algorithms: Theory and Practice*, 2000, pp. 267–277.
- [13] G. Silveira and E. Malis, "Unified direct visual tracking of rigid and deformable surfaces under generic illumination changes in grayscale and color images," *Int. J. Comput. Vision*, vol. 89, no. 1, pp. 84–105, 2010.
- [14] R. I. Hartley, "In Defense of the Eight-Point Algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 6, pp. 580–593, 1997.
- [15] D. Nistér, "An Efficient Solution to the Five-Point Relative Pose Problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–777, 2004.
- [16] X. Armangué, H. Araújo, and J. Salvi, "A Review on Egomotion by Means of Differential Epipolar Geometry Applied to the Movement of a Mobile Robot," *Pattern Recognit.*, vol. 36, no. 12, pp. 2927–2944, 2003.
- [17] P. H. S. Torr and A. Zisserman, "MLESCAC: a New Robust Estimator with Application to Estimating Image Geometry," *Comput. Vis. Image Underst.*, vol. 78, pp. 138–156, 2000.
- [18] A. D. Jepson and D. J. Heeger, "Simple Method for Computing 3D Motion and Depth," in *Int. Conf. on Comput. Vision*, 1990, pp. 96–100.
- [19] S. Soatto and P. Perona, "Recursive 3-D Visual Motion Estimation Using Subspace Constraints," *Int. J. Comput. Vision*, vol. 22, no. 3, pp. 235–259, 1997.
- [20] Z. Zhang, P. Cui, and H. Cui, "Recovery of Egomotion from Optical Flow with Large Motion Based on Subspace Method," in *IEEE Int. Conf. on Rob. and Biomimetics*, 2006, pp. 555–560.
- [21] K. Kanatani, "3-D Interpretation of Optical Flow by Renormalization," *Int. J. Comput. Vision*, vol. 11, no. 3, pp. 267–282, 1993.
- [22] Y. Ma, J. Kosecká, and S. Sastry, "Linear Differential Algorithm for Motion Recovery: A Geometric Approach," *Int. J. Comput. Vision*, vol. 36, no. 1, pp. 71–89, 2000.
- [23] T. Tian, C. Tomasi, and D. J. Heeger, "Comparison of Approaches to Egomotion Computation," in *IEEE Conf. on Comput. Vision and Pattern Recogn.*, 1996, pp. 315–320.
- [24] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," in *Alvey Vision Conf.*, vol. 15, 1988, p. 50.
- [25] K. Daniilidis and H. Nagel, "Analytical Results on Error Sensitivity of Motion Estimation from Two Views," in *European Conf. on Comput. Vision*, 1990, pp. 199–208.
- [26] T. Vaudrey, C. Rabe, R. Klette, and J. Milburn, "Differences between Stereo and Motion Behaviour on Synthetic and Real-World Stereo Sequences," in *Image Vision Comput.*, 2008, pp. 1–6.
- [27] J. Shi and C. Tomasi, "Good Features to Track," in *IEEE Conf. on Comput. Vision and Pattern Recogn.*, 1994, pp. 593–600.
- [28] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams: a Factorization Method - Part 3 Detection and Tracking of Point Features," Tech. Rep., 1991.
- [29] W.-Y. Lin, G.-C. Tan, and L.-F. Cheong, "When Discrete Meets Differential," *Int. J. Comput. Vision*, vol. 86, no. 1, pp. 87–110, 2010.