MONOCULAR EGOMOTION ESTIMATION BASED ON IMAGE MATCHING

Diego Cheda, Daniel Ponsa and Antonio Manuel López

Computer Vision Center, Universitat Autònoma de Barcelona, Edifici O, Campus UAB, 08193, Bellaterra, Barcelona, Spain

Keywords: Advanced driver assistance systems, Egomotion estimation, Monocular camera system, Template tracking.

Abstract: In this paper, we propose a novel method for computing the egomotion of a monocular camera mounted on a vehicle based on the matching of distant regions in consecutive frames. Our approach takes advantage of the fact that the image projection of a plane can provide information about the camera motion. Instead of tracking points between frames, we track distant regions in the scene because they behave as an infinity plane. As a consequence of tracking this infinity plane, we obtain an image geometric transformation (more precisely, an infinity homography) relating two consecutive frames. This transformation is actually capturing the camera rotation, since the effect produced by the translation can be neglected at long distances. Then, we can compute the camera rotation as the result of the previously estimated infinity homography. After that, rotation can be canceled from images, just leading to a translation explaining the motion between two frames. Experiments on real image sequences show that our approach reaches higher accuracy w.r.t. state-of-the-art methods.

1 INTRODUCTION

The estimation of changes in the vehicle position and orientation along time (i.e., its egomotion) is a key component of many advanced driver assistance systems (ADAS) like adaptive cruise control, collision avoidance, lane-departure warning, etc. Despite that several sensors are available for obtaining egomotion measurements (e.g., inertial sensors), many research efforts tend toward the use of cameras due to their ease of integration, low-cost and power consumption. In that sense, different proposals exist for determining the camera pose based just on images, avoiding the installation of additional sensors on the vehicle.

Our work concerns monocular egomotion estimation using a camera mounted rigidly in a vehicle. Although the problem is more challenging, we have considered that monocular systems will be present in vehicles solving other applications, and our aim is extending functionalities without adding more cost.

Regarding visual egomotion methods, in general, they treat all image points in the same way w.r.t. their distance from the camera. However, those points belonging to distant scene objects from the camera behave as lying on an infinity plane. The image projection of this plane is only affected by camera rotation. Taking advantage of that, we propose to track the image projection of distant regions to estimate the camera rotation. Specifically, in the following, distant regions are the ones whose optical flow due to camera translation between two consecutive frames is smaller than one pixel. As a consequence of tracking this plane, we obtain a transformation relating both frames, which captures the camera rotation, since the effect produced by the translation is assumed as negligible at long distances. Then, camera rotation is extracted from the computed transformation. Once rotation is known, we register both frames, canceling its effects and leading to a pure translation motion between frames. From this image flow, translation is estimated by solving a linear equation system.

Our approach has several advantages w.r.t. most general egomotion methods. First, we avoid the feature extraction and matching, which is valuable since small errors in the estimated image flow usually bring to large perturbations in the motion estimation. Second, in contrast to most of the feature-based methods, we decouple rotation from translation estimation, leading to accurate estimations because translation estimation errors do not affect the rotation computation. Third, our method is not affected by ambiguities produced by camera motion (Fermüller and Aloimonos, 1998), since distant regions are mainly affected just by rotation. Finally, our method is also robust to outliers because errors in the distant regions segmentation have a low impact on the regions tracking.

Cheda D., Ponsa D. and López A. (2012).
MONOCULAR EGOMOTION ESTIMATION BASED ON IMAGE MATCHING.
In *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods*, pages 425-430
DOI: 10.5220/0003776904250430

Obviously, the results of our algorithm would be affected when acquired images do not show distant regions due to obstructions in the field of view (e.g., a truck or a wall in front of the vehicle). These situations could be detected and properly treated.

The paper is organized as follows. In Sec. 2, we review related works. Next, we introduce our approach based on tracking distant regions. Finally, we describe and discuss the experimental results.

2 RELATED WORK

The egomotion problem concerns the estimation of the 3D rigid motion of the camera along a sequence, involving six degrees of freedom (DOF). The goal is estimating a translation vector $\dot{\mathbf{t}}$ and the angles $\boldsymbol{\omega}$ of a rotation matrix $\mathbf{R}$ from the image motion observed in subsequent frames. In this paper, we develop a vision-based method to egomotion estimation. Using a single camera, DOF is reduced to five since translation can be only recovered up to a scale factor (i.e., only translation direction is estimated, while its magnitude cannot be recovered due to lack of depth information). In the following, we review the most important related works.

Egomotion methods are classified as discrete or differential (Cheda et al., 2010) depending on whether they use point matches between views or optical flow, respectively. In contrast to these methods, direct ones compute egomotion without the need of matching between views (Zelnik-Manor and Irani, 2000).

In general, feature-based methods consider all points in the similar way w.r.t. their depth in the scene. However, some recent works try to take advantage of the relation between image motion and scene depth. Using a monocular camera, in (Burschka and Mair, 2008), a RANSAC process selects optical flow vectors that can be explained only by a particular rotation. Implicitly, this process tries to select image points located at far distances from the camera. Then, using the selected set of optical flow vectors, rotation is estimated by solving a linear equation system. Using a stereo camera, in (Obdrzalek and Matas, 2010), a voting strategy is used to egomotion estimation. Rotation is estimated through a voting schema where a vote weighted with the triangulated distance of each point to the camera is assigned for each motion vector. Once rotation is computed, the rest of motion is due to translation, which is also estimated by a voting strategy. In (Thanh et al., 2010), rotation and translation are estimated using far/near features, respectively. The point classification is done by checking their disparity over seven views of the same scene provided by a special omnidirectional sensor.

Another approach uses homographies to estimate motion parameters when the corresponding points are on a plane. In the ADAS context, some works try to track points over a ground plane such that this plane induces a homography relating two consecutive frames (Liang and Pears, 2002; Wang et al., 2005; Scaramuzza et al., 2010). To do that, a homography over two frames is robustly estimated by selecting points belonging to the ground plane. Inlier points are used to compute the camera motion by decomposing the estimated homography.

A disadvantage of using just points for egomotion estimation is that valuable information for motion estimation is unexploited. For instance, distant regions like sky, mountains, etc. are discarded because a few points can be detected over nearly uniform and low-textured surfaces. Then, distant regions are hard to be tracked by using interesting points. In contrast to the feature-based methods, direct methods avoid feature matching, and use only measurable information from images (Zelnik-Manor and Irani, 2000; Comport et al., 2010). Most of these approaches require a physical plane existing in the scene to obtain an accurate estimation of the camera motion. However, if no such planar surface exists, they cannot be applicable. The work of (Comport et al., 2010) does not require a region of interest, but it uses all image information by performing a dense stereo matching between consecutive frames.

Our proposal is a novel method within this category. Instead of requiring a real scene plane, we extract camera motion information from distant regions since they behave as an infinite plane. This strategy allows us to effectively annihilate the rotation between two consecutive frames, leading to just translation explaining the remaining camera motion. The basis of our proposal is the assumption that distant regions can be automatically segmented from images, which has been proved feasible in several application domains (see Sec. 3.3). In the next section, we define our approach.

3 PROPOSED METHOD

Now, we overview our method. First, we summarize the coplanar relation between two frames. Then, we describe our algorithm in the subsequent sections.

3.1 Transformation Relating Two Views

We consider the relation between two frames based on the property that the projection of a plane into these frames is related by a homography.

Let $\mathbf{p} = [p_x, p_y, p_z]^T$ and $\mathbf{p}' = [p'_x, p'_y, p'_z]^T$ be the coordinates of the same scene point, expressed in two different cameras coordinates systems C and C', respectively. The projection of $\mathbf{p}$ and $\mathbf{p}'$ into the image planes is $\mathbf{q} = \mathbf{K}\mathbf{p} = [q_x, q_y, q_z]^T$ and $\mathbf{q}' = \mathbf{K}\mathbf{p}' = [q'_x, q'_y, q'_z]^T$, and $\mathbf{K}$ is a calibration matrix with a focal length f and the principal point at the origin $[0,0]$.

C and C' are related by a rigid body motion

$$\mathbf{p}' = \mathbf{R}\mathbf{p} + \mathbf{t} \ , \qquad (1)$$

where $\mathbf{t} = [t_x, t_y, t_z]^T$ is a translation vector and, assuming that the camera rotation is small, $\mathbf{R}$ is a rotation matrix parametrized by Euler angles $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$ (pitch, yaw and roll, respectively) and approximated by (Adiv, 1985)

$$\mathbf{R} = \begin{bmatrix} 1 & -\omega_z & \omega_y \\ \omega_z & 1 & -\omega_x \\ -\omega_y & \omega_x & 1 \end{bmatrix} \ .$$

Furthermore, we assume that $\mathbf{p}$ lies on a plane Π, which is defined by its normal vector $\mathbf{n} = [\alpha, \beta, \gamma]$ as

$$\frac{\mathbf{n}}{d_\Pi}\mathbf{p} = 1 \ , \qquad (2)$$

where $\mathbf{n}^T$ is a unit vector in the direction of the plane normal, and d_Π is the distance to the plane from C.

Plugging (2) into (1), we obtain the expression

$$\mathbf{p}' = \mathbf{G}\mathbf{p} \ , \qquad (3)$$

where $\mathbf{G} = (\mathbf{R} + \frac{\mathbf{nt}}{d_\Pi})$ is the homography induced by Π in 3D. In 2D, the transformation between the corresponding image points can be written as $\mathbf{q}' = \mathbf{H}\mathbf{q}$, where $\mathbf{H}$ is the homography between two views of Π defined as

$$\mathbf{H} = \mathbf{K}\left(\mathbf{R} + \frac{\mathbf{nt}}{d_\Pi}\right)\mathbf{K}^{-1} \ . \qquad (4)$$

Since $\mathbf{t}$ depends on the unknown depth d_Π, it can be computed only up to a scale factor. Notice that the contribution of camera rotation $\mathbf{R}$ to the displacement of an image point is independent of the depth. We exploit this fact to camera rotation estimation.

In similar manipulation to (Zelnik-Manor and Irani, 2000), due to $\mathbf{H}$ is computed up to a scale factor

$$\begin{bmatrix} q'_x \\ q'_y \end{bmatrix} = \frac{1}{\mathbf{H}_3\mathbf{q}} \begin{bmatrix} \mathbf{H}_1\mathbf{q} \\ \mathbf{H}_2\mathbf{q} \end{bmatrix} \ ,$$

where $\mathbf{H}_i$ is the i^{th}-row of $\mathbf{H}$, and the image flow field $[\dot{q}_x, \dot{q}_y]^T = [q'_x - q_x, q'_y - q_y]^T$ is

$$\begin{bmatrix} \dot{q}_x \\ \dot{q}_y \end{bmatrix} = \begin{bmatrix} \dfrac{(\mathbf{H}_1 - \mathbf{H}_3 q_x)\mathbf{q}}{\mathbf{H}_3\mathbf{q}} \\ \dfrac{(\mathbf{H}_2 - \mathbf{H}_3 q_y)\mathbf{q}}{\mathbf{H}_3\mathbf{q}} \end{bmatrix} \ . \qquad (5)$$

If the camera rotations are small, which is the case in the neighboring frames of a sequence taken by a moving vehicle, we can assume that $\mathbf{H}_3\mathbf{q}$ is approximately equal to 1, f is large enough, and the translation over Z-axis is small relative to the plane depth (Zelnik-Manor and Irani, 2000).

Now, we assume that Π is located at infinity, which is the case for distant regions. In this case, from Eq. (3), we observe that if d_Π tends to ∞, then $\frac{\mathbf{nt}}{d_\Pi}$ tends to 0, letting $\mathbf{G} = \mathbf{R}$. This means that $\mathbf{H}$ from Eq. (4) does not depend on the translation between views, only on the rotation and camera internal parameters. Thus, the transformation between the corresponding image points can be rewritten as $\mathbf{q}' = \mathbf{H}_\infty\mathbf{q}$, where $\mathbf{H}_\infty$ is the infinite homography matrix between the two views of Π_∞ defined as $\mathbf{H}_\infty = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}$.

Given $\mathbf{H}_\infty$, and operating over Eq. (5), the image flow describing the motion between frames becomes

$$\dot{\mathbf{q}} = \mathbf{Q}\boldsymbol{\omega} \ , \qquad (6)$$

where $\mathbf{Q}$ only depends on $\mathbf{q} = [q_x, q_y, 1]^T$, with $q_z = 1$

$$\mathbf{Q} = \begin{bmatrix} -\dfrac{q_x q_y}{f} & f + \dfrac{q_x^2}{f} & -q_y \\ -f - \dfrac{q_y^2}{f} & \dfrac{q_x q_y}{f} & q_x \end{bmatrix} \ .$$

We use Eq. (6) as the base of our approach to compute camera orientation. In the next section, we overview our algorithm that takes advantage of this motion's particularity to estimate camera rotation by tracking distant regions.

3.2 Algorithm Overview

Basically, given two consecutive frames $\mathcal{I}$ at instant t and $t+1$, the algorithm proceeds as follows:

1. Distant regions are detected in $\mathcal{I}_{t+1}$ as we explain in Sec. 3.3. The result of this step is a template $\mathcal{T}$ containing the plane to be tracked.
2. $\mathcal{T}$ is aligned w.r.t. $\mathcal{I}_t$. Camera rotation parameters are extracted based on Eq. (6) (see Sec. 3.4).
3. The rotation effect between $\mathcal{I}_t$ and $\mathcal{I}_{t+1}$ is canceled, leading to just a translation $\mathbf{t}$ explaining the observed motion between both frames.
4. Finally, translation $\mathbf{t}$ is computed as in Sec. 3.5.

3.3 Distant Regions Segmentation

Traditionally, depth estimation has been addressed as a 3D reconstruction problem, focusing on multi-view methods (e.g., structure from motion). Nevertheless, studies on human vision suggest that depth perception is also supported by monocular cues such as occlusions, perspective, textures, etc. Recently, some proposals on depth estimation from a single image have been done (Saxena et al., 2009). However, for many applications, obtaining exact depths of the scene may not be necessary. In our case, only having information about the proximity/distantness of some regions can be enough for egomotion estimation. It is out of scope of the paper to address this problem. However, a discrete depth map could be estimated by segmenting the distance space in two (or more) threshold ranges. Using an appropriate set of discriminative visual features, a classifier could be trained for distinguishing close/distant regions according to a desired distance threshold. This segmentation would allow us to effectively distinguish the distant regions in an image.

3.4 Distant Regions Matching

Once distant regions for the frame I_{t+1} have been computed, we have a template $\mathcal{T}$ corresponding to a plane. Then, we align $\mathcal{T}$ w.r.t. I_t using Lucas-Kanade algorithm (Lucas and Kanade, 1981). This algorithm iteratively minimizes the difference between $\mathcal{T}$ and I_t under the following goal objective

$$\sum_{\mathbf{q}} \left(I_t(\mathcal{W}(\mathbf{q},\boldsymbol{\omega})) - \mathcal{T}(\mathbf{q}) \right)^2$$

w.r.t. $\boldsymbol{\omega}$, where $\mathcal{W}(\mathbf{q},\boldsymbol{\omega}) = \mathbf{q} + \dot{\mathbf{q}} = \mathbf{q} + \mathbf{Q}\boldsymbol{\omega}$ from Eq. (6). As the result of this process, we obtain the rotation parameters $\boldsymbol{\omega}$ to align $\mathcal{T}$ w.r.t. I_t.

3.5 Translation Estimation

Once rotation angles $\boldsymbol{\omega}$ are estimated by aligning $\mathcal{T}$ w.r.t. I_t, we can cancel the rotation effects. Without rotation motion, the difference between both frames I_t and I_{t+1} is due to the camera translation.

Assuming a plane with $\mathbf{n} = [0,0,1]^T$ (i.e., the plane is located over the camera principal axis), we can compute translation by solving the following equation system from Eq. (5)

$$\dot{\mathbf{q}} = \begin{bmatrix} f & 0 & -q_x \\ 0 & f & -q_y \end{bmatrix} \mathbf{t} \ .$$

To solve this system, we select matching points between I_t and I_{t+1}, belonging to close regions.

4 EXPERIMENTAL RESULTS

In this section, we test our approach on sequences taken with a stereo camera mounted on a vehicle driven through a city (Kitt et al., 2010)[1]. This sequence consists of more than 1400 frames, where translations and rotations ground truth (GT) are provided by measurements of an INS sensor. Additionally, we have taken advantage of a stereo depth map available on the processed sequence to optimally segment distant regions. In that way, we quantify the best performance that can be achieved with our proposal. Distant regions are defined as those composed by pixels located beyond 70 meters (m), where the effect of the vehicle translation on the optical flow is subpixel.

In the ADAS context, the dominant angle variation is on the pitch and yaw angles, due to the suspension system and the vehicle turning effects, respectively. In normal driving situations, the roll can be assumed as null. Then, Eq. (6) is reduced to two DOF.

Region matching between two consecutive frames is done by a modified version of Lucas-Kanade algorithm based on the public available Matlab code described in (Baker and Matthews, 2004). The main improvements w.r.t. that version are the inclusion of a multi-resolution pyramidal schema that allows us to consider different scales during template matching, and the use of arbitrary shape templates.

We measure the accuracy of our approach by computing the rotation estimation errors as in (Tian et al., 1996). Mean rotation error is quantified by the difference angle between the true rotation $\boldsymbol{\omega}_i$ and the estimated rotation $\bar{\boldsymbol{\omega}}_i$, at frame $i = 1,\ldots,N$. For this purpose, $\mathbf{R}_i$ and $\bar{\mathbf{R}}_i$ for both $\boldsymbol{\omega}_i$ and $\bar{\boldsymbol{\omega}}_i$ are built. The product between $\mathbf{R}_i$ and $\bar{\mathbf{R}}_i$ is an identity matrix when both are equal. Thus, the difference between both matrices is defined as $\triangle\mathbf{R}_i = \mathbf{R}_i^T \bar{\mathbf{R}}_i$. In Euler terms, $\triangle\mathbf{R}_i$ can be characterized by an axis unit vector and an angle. This angle is used as the rotation error. Since $trace(\mathbf{R}_i) = 1 + 2cos(\alpha_i)$, then the angle is equal to

$$\mu(\triangle\mathbf{R}_i) = cos^{-1}\left(\frac{1}{2}\left(trace(\triangle\mathbf{R}_i) - 1\right)\right) \ .$$

Then, we compute the mean of rotation errors (MRE) for the whole sequence, providing a scalar value for $\boldsymbol{\omega}$, and simplifying the evaluation of the algorithms.

Trajectory errors are quantified by computing the Euclidean distance between both the GT and estimated trajectory as follows

$$e(\mathbf{q}_i, \mathbf{q'}_i) = \sqrt{(q_x - q'_x)^2 + (q_z - q'_z)^2} \ ,$$

where $\mathbf{q}_i$ are the 2D coordinates of the GT trajectory and $\mathbf{q'}_i$ are the coordinates of the estimated trajectory on XZ plane. We compute the mean Euclidean distance (MED) averaging $e(\mathbf{q}_i, \mathbf{q'}_i)$ over all frames. We do not include the displacement on Y-axis because GT

[1]http://www.rainsoft.de/software/

Table 1: Comparison between our approach and general and distance-point-based egomotion methods. The mean rotation error (in degrees) and the mean Euclidean distance error (in meters) are shown for each algorithm.

Algorithms	Error	
	MRE (in °)	MED (in *m*)
Our	0.052	4.93
5pts	0.094	10.70
Stereo	0.067	4.16
Burschka	0.108	7.32
Distant Points	0.147	4.38

data are unreliable in this coordinate (recognized by their authors), and distort the comparison.

Next, we show the results obtained by our approach versus general and distant-based methods. Finally, we analyze the robustness of our approach when distant regions are imprecisely segmented.

4.1 Comparison Against Other Methods

We compare our results against the ones obtained by two general methods: 5pts algorithm (Nistér, 2004) and by a stereo algorithm proposed by (Kitt et al., 2010); and we also compare our results against the ones of two distant points-based methods: Burschka et al. (Burschka and Mair, 2008) and distant-point algorithms. The last algorithm, implemented by us, use distant points instead of distant regions to egomotion estimation. In this case, the camera rotation is computed by solving a linear equation system over the optical flow of points located at a distance over 70 *m*.

Results are shown in Tab. 1. The accuracy of our method outperforms the other considered methods in rotation estimation (MRE). This is because we take advantage of valuable information for motion estimation, which is unexploited by feature-based methods. For instance, we are considering regions located at sky which are discarded by feature-based methods since few points can be detected over such regions. Figure 1 shows a comparison between our estimation of yaw angle (in radians) against the GT provided by the INS sensor. Our estimation is very close to GT. Notice that the estimated yaw angle variation using our method is smoother. We guess that this is because we do a maximal use of the information available in frames to estimate the camera rotation, which lead to estimated parameters reflecting the behavior of the real camera motion in the sequence, characterized by a slow and smooth movement.

Regarding the estimated trajectory (MED), our approach has a comparable performance to the stereo and distant point algorithms, and outperforms both 5pts and Burschka et al. algorithms. The error percentage is 1.08 % of the total traveled distance, which

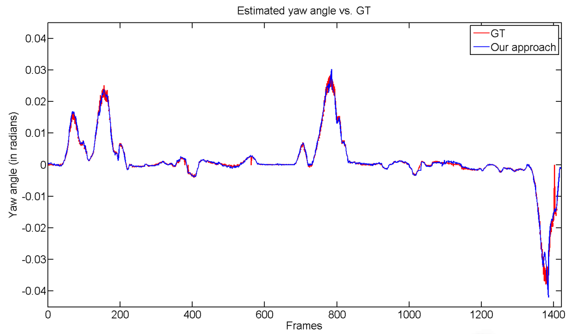

Figure 1: Comparison between estimated yaw angle versus GT (in radians) on the sequence.

Figure 2: Comparison between estimated trajectory and GT.

is a good performance. Figure 2 shows the trajectory described by our algorithm versus GT. Our result is accurate w.r.t INS sensor measurements.

Naturally, as we stated in Sec. 1, the results of our approach would be affected when most of the field of view is obstructed. In these situations, the distant region segmenter does not provide enough information to apply our method. Then, we can use another egomotion method to deal with such cases.

4.2 Robustness to Noisy Segmentation

We are based on stereo depth maps to segment the image in close/far regions. However, segmentations as described in Sec. 3.3 will present outliers due to classification errors. To test our algorithm under these situations, we add different amounts of near regions simulating the performance of a realistic segmenter. Close regions are chosen randomly and used during template matching as if they were distant ones. Additionally, we remove (in the same proportion as we add) distant regions from the considered template to simulate misclassification. Table 2 shows the results

Table 2: The mean rotation error (in degrees) is shown for our approach under different amount of outliers.

Error	Outliers		
	10%	20%	30%
MRE (in $^\circ$)	0.0570	0.0613	0.0667

of this experiment. The performance does not degrade significantly even under a large number of outliers.

5 CONCLUSIONS

In this work, we have proposed a direct monocular egomotion method based on tracking distant regions. These regions can be assumed as located at the infinite plane, inducing an infinity homography relation between two consecutive frames. By tracking that plane, we are able to estimate the camera rotation. Once rotation is computed, we cancel its effect on the images, leaving the resulting motion due to camera translation. This method is simple and performs a sufficiently stable camera parameter's estimation.

We successfully apply our algorithm to a sequence taken from a vehicle driving in an urban scenario. Rotations are accurately estimated, since distant regions provide strong indicators of camera rotation. In comparison to the state-of-the-art methods, our approach outperforms the considered methods. Moreover, from the experimental results, we conclude that our method is also robust to segmentation errors. Occasional mistakes can occur when acquired images do not show distant regions due to obstructions in the field of view.

As future work, we plan to test the proposal using a monocular segmentation algorithm to distinguish between close/far regions from single images. These segmentations will provide enough information about depth to be used in our egomotion algorithm.

ACKNOWLEDGEMENTS

This work is supported by Spanish MICINN projects TRA2011-29454-C03-01, TIN2011-29494-C03-02, Consolider Ingenio 2010: MIPRCV (CSD200700018), and Universitat Autònoma de Barcelona.

REFERENCES

Adiv, G. (1985). Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 7(4):384 –401.

Baker, S. and Matthews, I. (2004). Lucas-Kanade 20 Years On: A Unifying Framework. *Int. J. Comput. Vision*, 56(1):221 – 255.

Burschka, D. and Mair, E. (2008). Direct Pose Estimation with a Monocular Camera. In *Int. Workshop Robot Vision*, pages 440–453.

Cheda, D., Ponsa, D., and López, A. (2010). Camera Egomotion Estimation in the ADAS Context. In *IEEE Conf. Intell. Transp. Syst.*, pages 1415 –1420.

Comport, A., Malis, E., and Rives, P. (2010). Real-time Quadrifocal Visual Odometry. *Int. J. Rob. Res.*, 29:245–266.

Fermüller, C. and Aloimonos, Y. (1998). Ambiguity in structure from motion: sphere versus plane. *International Journal of Computer Vision*, 28(2):137–154.

Kitt, B., Geiger, A., and Lategahn, H. (2010). Visual Odometry based on Stereo Image Sequences with RANSAC-based Outlier Rejection Scheme. In *IEEE Intell. Veh. Symp.*, pages 486 – 492.

Liang, B. and Pears, N. (2002). Visual Navigation Using Planar Homographies. In *IEEE Int. Conf. Rob. Autom.*, volume 1, pages 205 – 210.

Lucas, B. and Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. In *Int. Joint Conf. Artif. Intell.*, pages 674–679.

Nistér, D. (2004). An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777.

Obdrzalek, S. and Matas, J. (2010). A Voting Strategy for Visual Ego-motion from Stereo. In *IEEE Intell. Veh. Symp.*, pages 382 –387.

Saxena, A., Sun, M., and Ng, A. (2009). Make3D: Learning 3D Scene Structure from a Single Still Image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):824–840.

Scaramuzza, D., Fraundorfer, F., and Pollefeys, M. (2010). Closing the Loop in Appearance-Guided Omnidirectional Visual Odometry by Using Vocabulary Trees. *Rob. Autom. Syst.*, 58:820–827.

Thanh, T., Kojima, Y., Nagahara, H., Sagawa, R., Mukaigawa, Y., Yachida, M., and Yagi, Y. (2010). Real-Time Estimation of Fast Egomotion with Feature Classification Using Compound Omnidirectional Vision Sensor. *IEICE Trans. Inf. Syst.*, 93D(1):152–166.

Tian, T., Tomasi, C., and Heeger, D. J. (1996). Comparison of Approaches to Egomotion Computation. In *IEEE Conf. Comput. Vision Pattern Recogn.*, pages 315–320.

Wang, H., Yuan, K., Zou, W., and Zhou, Q. (2005). Visual Odometry Based on Locally Planar Ground Assumption. In *IEEE Int. Conf. Inf. Acquisition*, pages 59–64.

Zelnik-Manor, L. and Irani, M. (2000). Multi-Frame Estimation of Planar Motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22:1105–1116.