Moving on to Dynamic Environments: Visual Odometry using Feature Classification

Bernd Kitt, Frank Moosmann, and Christoph Stiller

Abstract— **Visually estimating a robot's own motion has been an active field of research within the last years. Though impressive results have been reported, some application areas still exhibit huge challenges. Especially for car-like robots in urban environments even the most robust estimation techniques fail due to a vast portion of independently moving objects. Hence, we move one step further and propose a method that combines ego-motion estimation with low-level object detection. We specifically design the method to be general and applicable in real-time. Pre-classifying interest points is a key step, which rejects matches on possibly moving objects and reduces the computational load of further steps. Employing an Iterated Sigma Point Kalman Filter in combination with a RANSAC based outlier rejection scheme yields a robust frame-to-frame motion estimation even in the case when many independently moving objects cover the image. Extensive experiments show the robustness of the proposed approach in highly dynamic environments with speeds up to 20m/s.**

I. Introduction

Estimating a rover's motion is an important prerequisite for reliably executing a wide range of tasks like mapping, obstacle detection, and autonomous driving. In the past, this localization task was often based on wheel speed sensors or inertia sensors.

In recent years, the computational power even on standard PC hardware increased dramatically. Furthermore, cameras became cheaper and yield rich information about the environment of the vehicle. As a consequence, many algorithms have been developed using vision based localization. Compared to wheel speed sensors, vision based localization is more precise especially in slippery terrain. In comparison to inertia sensors the local drift rates are mostly lower. As for all incremental approaches, long-term drift can be mitigated by fusing GPS information or by applying loop-closure on (visual) place recognition.

Excellent algorithms have been introduced for laboratory-like, well structured environments or rough terrain with low speed (of approximately 1m/s). In urban environments with many independently moving objects, however, visually estimating a vehicle's ego-motion remains a problem (see Fig. 1). This is especially the case when the vehicle drives at moderate speed and no constraints can be put on the environment and the motion.

We here propose a method that can deal with these challenges. As sole input, video streams from a normal stereo camera rig are used. The only assumption we make is a known camera geometry, where the calibration of the stereo camera rig might even vary over time. As output, we obtain accurate motion estimates even if a large part of the image is covered by moving objects.

The remainder of this paper is organized as follows: the next section briefly describes work already done in the field of vision based motion estimation. In section III the basic motion estimation approach is introduced, which is extended by a basic outlier rejection scheme in section IV and adapted to dynamic environments in section V. We close the paper with experimental results of the proposed approach, a short conclusion and an outlook on future work.

II. Related Work

In the last decades, many algorithms for visual ego-motion estimation have been developed which can roughly be divided into two main categories: approaches using only one camera (e.g., [4], [27]) and approaches using stereo camera rigs. We concentrate on the latter, as they mostly yield better results than the monocular approaches [2] and because they do not suffer from scale ambiguities.

Further subdivision is possible into methods using feature tracking over a whole sequence of images (e.g., [24], [8], [14]) and methods matching features only between consecutive images (e.g., [13], [25], [26]). Due to the computational complexity of bundle adjustment we here focus on the latter.

To yield robust estimates, some approaches make use of assumptions about the surroundings and the vehicle movements. In [4] it is assumed that the vehicle moves on a plane which restricts the motion parameter space to two linear and one angular component. Scaramuzza et al. [21] use nonholonomic constraints of wheeled vehicles in order to reduce the parameter space.

Other methods combine visual ego-motion estimation with other sensors to improve estimation results and to reduce drift – an inherent problem of incremental motion estimation. While Dornhege et al. [8] additionally make use of inertial measurements, Agrawal et al. [1] use GPS and wheel speed

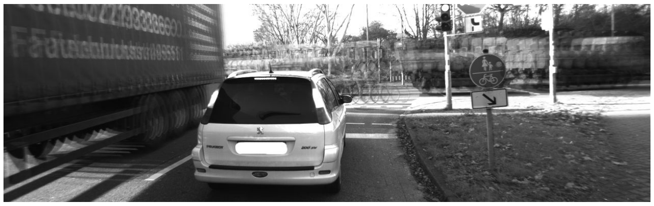


Fig. 1: Moving objects make egomotion estimation difficult.

Institute of Measurement and Control, Karlsruhe Institute of Technology, Germany. {frank.moosmann, bernd.kitt}@kit.edu
Video and code available on www.cvlibs.net

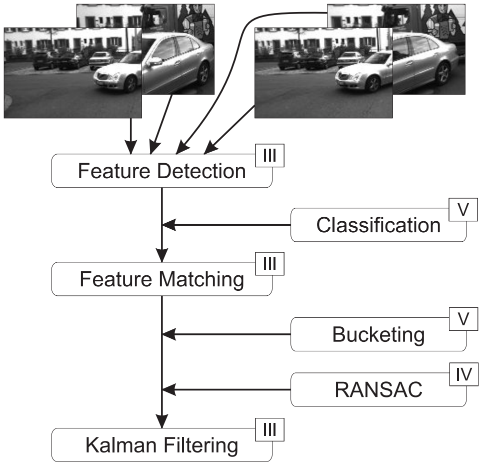


Fig. 2: Sketch of the proposed algorithm. The different components are explained in the specified sections.

sensors to improve their algorithms. Obviously, the use of GPS information limits drift dramatically due to the global nature of this system and could therefore also extend the method proposed in this work.

Good localization results have also been achieved using visual SLAM techniques (e.g. [7], [20]) which simultaneously estimate a map of the environment jointly with the trajectory of the observer in this map. Beside the computational complexity of these approaches, most of them perform well only in well structured environments with smooth camera motions at low speed.

Common to all previously mentioned work is the limitation to static environments or environments with only few moving objects. A rare exception is [9]. They interleave visual odometry with pedestrian detection and tracking, obtaining impressive results in crowded pedestrian zones. However, there are a few drawbacks to mention: object detection can currently cope only with pedestrians, computations are currently far-off from real-time, and results were shown for low speed only.

This work presents a light-weight motion estimation solely based on visual inputs. It extents [16] to yield accurate results even in environments with high traffic and at higher speed. The approach makes no assumptions about the motion or the surroundings and estimates all six degrees of freedom (6-DOF). Because of our focus on real-time applicability, we focus on frame-to-frame motion estimation.

III. Visual Ego-Motion Estimation

Fig. 2 illustrates the different steps of the proposed algorithm. The main branch describes the algorithm detailed in this section which performs well for static scenes. To apply the algorithm to dynamic environments we propose different improvements described in sections IV and V.

A. Trifocal Constraints for Visual Ego-Motion Estimation

In our approach we apply the $3 \times 3 \times 3$ *trifocal tensor* $\mathcal{T}$ which describes the relationship between three images of the same static scene. It encapsulates the projective geometry between different viewpoints and is independent from the scene structure [12]. We make use of the trifocal tensor's ability to map two corresponding feature points $\boldsymbol{x}_A \leftrightarrow \boldsymbol{x}_B$ in images A and B into image C. This mapping is expressed via the point-line-point transfer of the trifocal tensor:

$$\boldsymbol{x}_C^k = \boldsymbol{x}_A^k \cdot \boldsymbol{l}_{B,j} \cdot \mathcal{T}_i^{jk} \tag{1}$$

Here $\boldsymbol{l}_B$ denotes an arbitrary image line through feature point $\boldsymbol{x}_B$. Given the extrinsic and intrinsic camera calibration and the movement of the stereo camera rig, there is a mapping of corresponding feature points $\boldsymbol{x}_{R,k} \leftrightarrow \boldsymbol{x}_{L,k}$ captured at time step k into the current frames $k+1$ via $\boldsymbol{x}_{f,k+1} = h_f\left(\mathcal{T}_f, \boldsymbol{x}_{R,k}, \boldsymbol{x}_{L,k}\right)$ with $f \in \{R, L\}$. Here $\mathcal{T}_f$ denotes the trifocal tensor which relates the previous camera images with one of the current camera images. The reader is referred to [16] for more details.

B. Kalman-Filtering

In a first step, we detect corner like image features in both stereo image pairs. Different kinds of feature detectors and descriptors are possible (e.g. [11], [17], [3]). After feature detection, matches between the four images are established to get feature correspondences $\boldsymbol{x}_{R,k} \leftrightarrow \boldsymbol{x}_{L,k} \leftrightarrow \boldsymbol{x}_{R,k+1} \leftrightarrow \boldsymbol{x}_{L,k+1}$. These feature matches serve as measurements for the Kalman-Filter which is briefly described in the following.

To include knowledge about the dynamic behavior of the vehicle, we use a Kalman Filter to estimate the instantaneous state of the system. The discrete-time space filter equations are given by

$$\boldsymbol{y}_{k+1} = f\left(\boldsymbol{y}_k\right) + \boldsymbol{w}_k \tag{2}$$

$$\boldsymbol{z}_k = h\left(\boldsymbol{y}_k\right) + \boldsymbol{v}_k \tag{3}$$

where $\boldsymbol{y}_k = \left(v_{X,k}, v_{Y,k}, v_{Z,k}, \omega_{X,k}, \omega_{Y,k}, \omega_{Z,k}\right)^\mathsf{T}$ is the state of the system at time step k, $f\left(.\right)$ is the in general non-linear system equation, and $\boldsymbol{w}_k \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{Q}_k\right)$ is the system noise characterized by the 6×6 covariance matrix $\boldsymbol{Q}_k$. We here assume constant velocity between consecutive time steps, so the system equation simplifies to the linear equation $\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \boldsymbol{w}_k$. This assumption is nearly fulfilled if the camera provides images with a fairly high frame-rate. Even if this assumption is violated (e.g. in the case of acceleration, deceleration or turns), the following update step guarantees reliable motion estimation.

In equation (3), the non-linear function $h\left(.\right)$ relates the system state to the $4N$-dimensional measurement vector $\boldsymbol{z}_k = \left[u_{R,k,1}, \ldots, v_{L,k,N}\right]^\mathsf{T}$ and $\boldsymbol{v}_k \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{R}_k\right)$ represents the measurement noise in form of a $4N \times 4N$ diagonal matrix $\boldsymbol{R}_k$. N thereby denotes the number of feature correspondences between the 4 images. The measurement function is explained in more detail in [16].

Different extensions of Kalman Filters have been derived for the application to non-linear systems. In such cases a linearization around the current state is often performed using a first order Taylor-approximation. This yields the *Extended*

Kalman Filter (EKF) and the *Iterated Extended Kalman Filter (IEKF)*. In our case of highly non-linear equations the results of *Extended Kalman Filters* are mostly poor. The reason for this is that the used Taylor-approximation is only a first order approximation. A better choice in such cases is the usage of Kalman Filters based on the *Unscented Transform (UT)* [23]. Because the unscented transform incorporates information about higher order moments in the estimation process, estimates usually improve. Examples for filters propagating mean and covariance based on sigma points are the *Unscented Kalman Filter (UKF)* [15] or the *Iterated Sigma Point Kalman Filter (ISPKF)* [22]. We use the latter one in our approach. Besides the reduction in linearization error, the ISPKF has another benefit compared to EKF based filtering. In our experiments, the convergence of the ISPKF is approximately 60 times faster than the convergence of the IEKF, without the need for analytical derivatives (see [16] for a detailed analysis).

IV. RANSAC-based Outlier Rejection

The approach described in section III is highly sensitive to outliers. These stem from wrong feature matches or from moving objects. To make the algorithm more robust, the proposed approach is wrapped into the RANSAC algorithm: iteratively, a subset of the feature correspondences is randomly chosen and egomotion is estimated based on the current subset. The number of used subsets/iterations is given by

$$n = \frac{log\,(1-p)}{log\,(1-(1-\epsilon)^s)}. \tag{4}$$

Here, s is the minimum number of data points needed for estimation, p is the probability that at least one sample contains solely inliers and ϵ defines the assumed percentage of outliers in the data set [5]. Because of the low number of data points ($s = 3$) necessary for motion estimation, the number of samples is low even with a serious number of outliers. The percentage of outliers can even further be reduced using the classification proposed in the next section, which dramatically reduces the number of samples and accelerates the algorithm.

After the Kalman Filter converges, we compute all inliers using the Euclidean reprojection error. A feature is considered as an inlier, if the Euclidean reprojection error is lower than a certain threshold. A final estimation step with all inliers of the best sample is performed to give the final egomotion estimate for the current frame. The RANSAC based outlier rejection scheme already yields robust egomotion estimates even in the presence of few independently moving objects (see Fig. 3).

V. Advanced Outlier Rejection

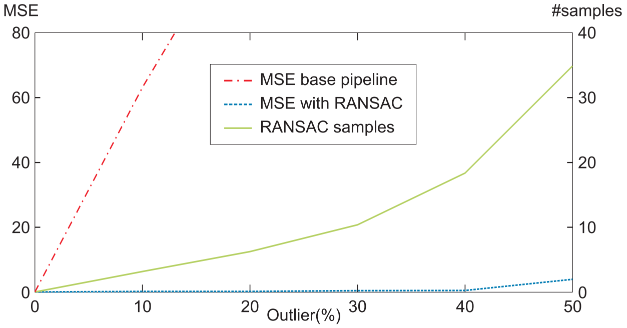


Fig. 3: Influence of non-systematic outliers.

The approach described in section III and IV is able to deliver accurate results even if wrong feature matches or few moving objects are present (cf. Fig. 3). However, an increasing number of outliers exponentially increases the required RANSAC samples and thus runtime. Additionally, if there are other regions of *consistent* motion present in the image, RANSAC might focus thereon. This leads to wrong estimates in heavy traffic scenes (see Fig. 4 and 5). The sole solution to this problem is to interleave ego motion estimation with object class detection in order to filter out (potentially) moving objects[1]. In contrast to [9], we here favor a light-weight approach applicable for real-time usage and all object-classes. In the following we present our approach which works as a preprocessing step for section III by filtering out matching candidates based on the appearance of local image regions.

A. Image Patch Classification

The first step of section III is to extract potential matching candidates by means of an interest point detector. We now seek to construct a classifier that is capable of deciding for each candidate whether it represents a potentially moving object (e.g. car, truck, pedestrian) or not.

By assigning to each candidate (u, v) a constant scale parameter s, we obtain a square *image patch*. Using some descriptor function, this patch is transformed into a *feature vector* and classified by some binary classification method. As our approach is dedicated for real-time usage, the combination of descriptor and classification function has to be not only accurate but also fast.

1) Descriptors: Within the last decade lots of descriptor functions have been developed, ranging from simply taking pixel values [18] over applying a full (Haar) wavelet transform [19] up to more sophisticated gradient-based descriptors like e.g. SIFT [17], SURF [3], or HOG [6]. As computational efficiency is a major selection criterion, we only focus on the speediest ones – pixel-wise grey values, its wavelet transform, SURF, and, as a baseline, SIFT. We give an experimental comparison in section VI.

2) Classifiers: After transforming a patch into a feature vector, it can then be classified as either positive (non-moving) or negative (potentially-moving). Many binary classification techniques exist with support vector machines (SVM) currently being the most popular for vision problems due to their robustness. Unfortunately, for each feature vector being classified, several kernel-calls are necessary which makes a SVM relatively slow. A fast and nevertheless

[1] Even advanced filtering techniques will fail when many neighboring cars move with the same motion.

accurate alternative are decision trees and its derivatives, e.g. boosting and randomized decision forests. We chose to use an ensemble of extremely randomized decision trees [10] as they are fast and the use of multiple trees makes the method robust. In fact, we experimentally verified that for the given problem they even outperform a SVM.

By counting the votes from all trees, the classifier effectively returns a single confidence value which is finally used to decide upon the class. Varying this threshold influences both, the classifier's precision (i.e. the percentage of correct positives) and recall (i.e. the percentage of positives identified). Using cross-validation, we select a threshold with high precision (such that nearly all negatives are rejected) but still acceptable recall-rate (so there are still enough positives left).

B. Adaptive Bucketing

After classification, correspondences are established between the four images as described in section III. Afterwards, a technique called bucketing [28] is applied on one of the four images to further restrict correspondences. As illustrated in Fig. 4, the image is divided into rectangles called *buckets*. Based on the classification, a bucket is set active, if a certain percentage of the contained features were classified as positives. A constant amount of correspondences is then selected by taking an equal number from each active bucket.

This so-called *adaptive bucketing* kills two birds with one stone: First, it is able to further reject some of the few false positives based on neighboring classification. Second, it guarantees equal distribution of the correspondences across the image.

The latter is needed due to several reasons: An equal distribution across the image results in features being also well distributed along the roll-axis of the vehicle. This turns out to be important for a good estimation of the linear and angular velocities. The distribution of image features along the roll-axis ensures that far as well as near features are used for the estimation process. Features with a large distance to the camera are little affected by linear camera motions; hence they give less information about the translatory motion of the vehicle but are an important cue for the angular velocities. Vice versa, near features give rich information about linear motions [4]. Additionally, bucketing reduces the drift rate of the approach. Experiments with simulated data suggest that high drift rates follow from biased scene points. This effect is mitigated by the use of bucketing.

Overall, advanced outlier rejection not only improves the quality of motion estimates in highly dynamic scenes, it also reduces overall computational costs. While classification of image patches is linear in their number, effective RANSAC with Kalman-Filtering grows more than quadratically (due to matrix inversion). Depending on the current scene, the additional time spent on classification can even speed up the complete method.

VI. Experimental Results

A. Classification of Image Regions

We took 12 single images from urban scenes and hand-labeled each pixel as either *potentially moving* or *stationary*. We then evaluated several features by 3-fold cross-validation on the labeled dataset. The features compare as in table I.

Method	Avg. Error	Avg. Area Under ROC
SURF	0.30	0.75
SIFT	0.19	0.88
Pixels	0.18	0.90
Wavelets	0.17	0.91

TABLE I: Comparison between different features for classification of key-points as potentially moving.

As processing time is a major design criterion, we chose to use the wavelet features. For subsequent motion estimation, we re-trained the classifier using all labeled image features.

The complete outlier rejection step is illustrated in Fig. 4. After classifying all interest points, adaptive bucketing is applied to further reduce false positives and to assure equal distribution across the image.

B. Ego-motion Estimation

The basic ego-motion estimation, as described in section III, was already analyzed in detail in [16]. Hence, we here focus on the performance gain induced by the advanced outlier rejection. Therefore, we use different challenging sequences captured in urban environments with high traffic. The recordings of a high-precision integrated navigation system (INS) serve as reference for comparison reasons. We use two measures for evaluation: the mean squared error (MSE) is given by

$$MSE = \frac{1}{\#frames} \cdot \sum_{frames} ||\boldsymbol{x}_{EST} - \boldsymbol{x}_{INS}||^2$$

and evaluates the average Euclidean distance between the estimated position and the reference position. Furthermore we show the mean positioning error (MPE), which is the mean pose error related to the length of the trajectory l_{INS} given by the INS:

$$MPE = \frac{\sqrt{MSE}}{l_{INS}}$$

Seq.	# Frames	MSE (base)	MSE (adv)	MPE (base)	MPE (adv)
1	300	$1.33m^2$	$0.14m^2$	4.62%	1.53%
2	329	$3.33m^2$	$3.06m^2$	5.38%	5.15%
3	436	$19.67m^2$	$12.48m^2$	6.87%	5.47%
4	130	$36.97m^2$	$29.40m^2$	3.50%	3.12%
5	789	$8.68m^2$	$8.41m^2$	5.09%	5.01%
6	447	$406.29m^2$	$81.82m^2$	6.57%	2.95%
7	1380	$81.17m^2$	$30.87m^2$	5.97%	3.68%

TABLE II: Comparison between using only RANSAC (base) and the advanced outlier rejection (adv) on high-traffic scenes. Example frames are depicted in Fig. 1, 4, 6.

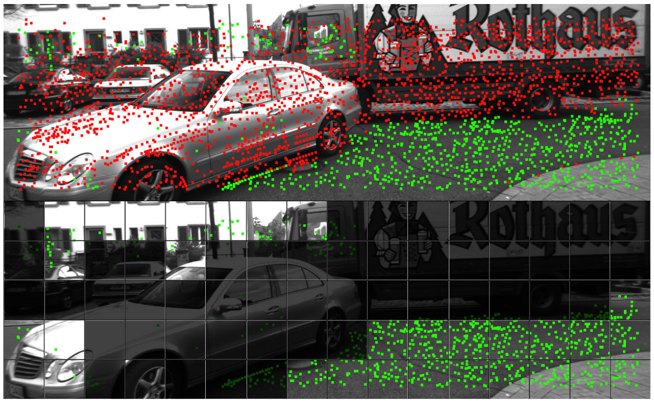


Fig. 4: Outlier rejection by image patch classification and adaptive bucketing (sequence 6).

The results for some of our experiments are given in table II. As can be seen, the MSE is significantly smaller using advanced outlier rejection. The partially large MSE values are due to the fact, that the INS drifts dramatically when the vehicle stops. See also Fig. 6.

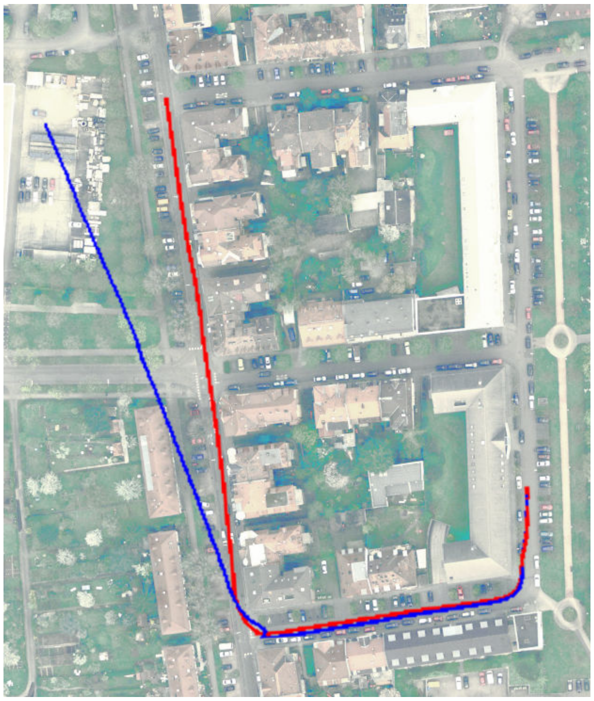


Fig. 5: Comparison between the trajectories estimated with (red) and without (blue) the advanced outlier rejection on sequence 6, superimposed to an aerial image of the street.

To show the results of the egomotion estimation, we illustrate different trajectories in Fig. 5 and 6. Fig. 5 illustrates a sequence where at the bottom left corner two moving objects covered a big part of the images. This situation is depicted in Fig. 4, where it is easy to imagine that the classification results will help the motion estimation. Indeed, the trajectory of the advanced outlier rejection is in accordance with the satellite image, whereas without advanced outlier rejection the trajectory only follows the street up to the intersection where the truck and the vehicle perturbed the motion estimation.

Other situations that illustrate the differences best are shown in Fig. 6. In both situation the own car stopped at a traffic light and heavy traffic was passing at the left and at the right. A backward motion is naturally estimated when classification is not used. With classification, the approach even outperforms our INS unit, which fuses GPS with inertial signals.

VII. Conclusion

We proposed an algorithm for reliably estimating a robot's own motion. Using stereo image streams, trifocal geometry between image triples is used together with Kalman-Filtering to obtain motion estimates. Outliers are rejected by RANSAC and a novel *advanced outlier rejection* scheme – a combination of appearance-based feature classification and equal feature selection on buckets.

Detailed experiments were carried out with our experimental vehicle in an urban setting. In contrast to most existing work, we showed that it is possible to obtain good motion estimates with cameras alone. Even with many moving objects present in the images good results were obtained, owing to the good job of the *advanced outlier rejection*.

The sole drawback of the proposed method is that the classifier has to be learned in advance. This includes that hand-labeled training examples must be present for all possible situations. Therefore we will try to adapt an online-learning approach for the feature classification that can adapt to various situations. Further work will focus on the few situations where estimates are still erroneous. Including a kinematic model of the car will already reduce sideways drifts. As with all incremental approaches, we could also improve the estimates by fusing GPS information.

Acknowledgements

The authors gratefully acknowledge the contribution of the German collaborative research center on Cognitive Automobiles (SFB/Tr28) granted by Deutsche Forschungsgemeinschaft. Aerial image courtesy of Stadt Karlsruhe, VLW Geodaten.

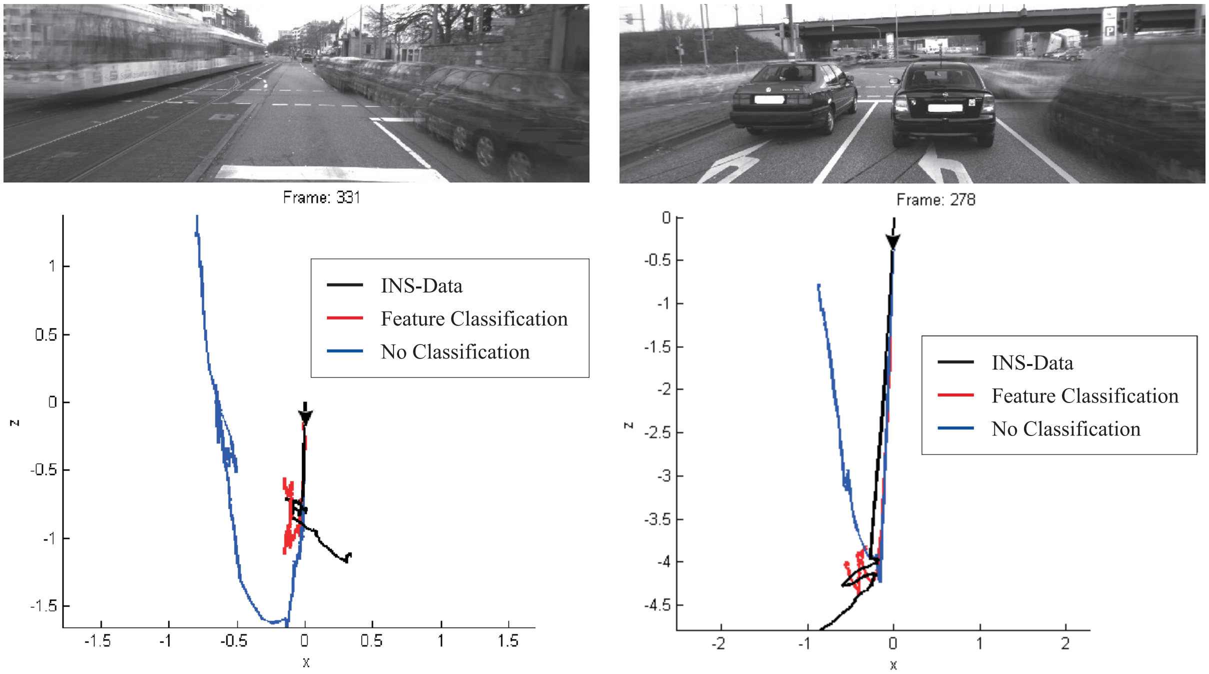
References

[1] M. Agrawal and K. Konolige, "Real-time localization in outdoor environments using stereo vision and inexpensive gps," in *Proc. of the International Conference on Pattern Recognition*, 2006, pp. 1063 – 1068.

[2] H. Badino, "A robust approach for ego-motion estimation using a mobile stereo platform," in *First International Workshop on Complex Motion*, 2004, pp. 198 – 208.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, June 2008.

[4] J. Campbell, R. Sukthankar, I. Nourbakhsh, and A. Pahwa, "A robust visual odometry and precipice detection system using consumer-grade monocular vision," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2005, pp. 3421 – 3427.



(a) Sequence 1

(b) Sequence 2

Fig. 6: Vehicle stop at a traffic light: blended camera images and bird's eye view of the trajectories estimated in comparison to the reference INS-trajectory.

[5] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-point ransac for ekf-based structure from motion," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2009.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision & Pattern Recognition*, vol. 2, June 2005, pp. 886–893.

[7] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, 2007.

[8] C. Dornhege and A. Kleiner, "Visual odometry for tracked vehicles," in *Proc. of the IEEE International Workshop on Safety, Security and Rescue Robotics*, 2006.

[9] A. Ess, B. Leibe, K. Schindler, and L. van Gool, "Robust multiperson tracking from a mobile platform," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1831–1846, 2009.

[10] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machile Learning Journal*, vol. 63, no. 1, 2006.

[11] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. of the 4th Alvey Vision Converence*, 1988, pp. 147 – 151.

[12] R. Hartley and A. Zisserman, *Multiple View Geometry in computer vision*, 2nd ed. Cambridge University Press, 2008.

[13] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, September 2008, pp. 3946 – 3952.

[14] A. E. Johnson, S. B. Goldberg, Y. Cheng, and L. H. Matthies, "Robust and efficient stereo feature tracking for visual odometry," in *Proc. of the IEEE International Conference on Robotics and Automation*, May 2008, pp. 39 – 46.

[15] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," in *Proc. of the IEEE*, vol. 92, no. 3, March 2004, pp. 401 – 422.

[16] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme," in *Proc. of the IEEE Intelligent Vehicles Symposium*, San Diego, CA, USA, June 2010, pp. 486 – 492.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91 – 110, 2004.

[18] R. Marée, P. Geurts, J. Piater, and L. Wehenkel, "Random subwindows for robust image classification," in *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 34–40.

[19] F. Moosmann, D. Larlus, and F. Jurie, "Learning saliency maps for object categorization," in *ECCV Workshop on the Representation and Use of Prior Knowledge in Vision*, 2006.

[20] L. M. Paz, P. Piniés, J. D. Tardós, and J. Neira, "Large-scale 6-dof slam with stereo-in-hand," *IEEE Trans. on Robotics*, vol. 24, no. 5, pp. 946 – 957, October 2008.

[21] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, "Real-time monocular visual odometry for on-road vehicles with 1-point ransac," in *Proc. of the IEEE Conference on Robotics and Automation*, May 2009.

[22] G. Sibley, G. Sukhatme, and L. Matthies, "The iterated sigma point kalman filter with applications to long range stereo," in *Proc. of Robotics: Science and Systems*, August 2006.

[23] D. Simon, *Optimal State Estimation*, 1st ed. Wiley, 2006.

[24] N. Sünderhauf, K. Konolige, S. Lacroix, and P. Protzel, "Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle," in *Tagungsband Autonome Mobile Systeme 2005*. Springer, 2005, pp. 157 – 163.

[25] A. Talukder, S. Goldberg, L. Matthies, and A. Ansar, "Real-time detection of moving objects in a dynamic scene from moving robotic vehicles," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, October 2003, pp. 1308 – 1313.

[26] A. Talukder and L. Matthies, "Real-time detection of moving objects from moving vehicles using dense stereo and optical flow," in *Proc. of the IEEE International Conference on Intelligent Robots and Systems*, vol. 4, September 2004, pp. 3718 – 3725.

[27] K. Yamaguchi, T. Kato, and Y. Ninomiya, "Vehicle ego-motion estimation and moving object detection using a monocular camera," in *Proc. of the International Conference on Pattern Recognition*, 2006, pp. 610 – 613.

[28] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artifical Intelligence*, vol. 78, no. 1 – 2, pp. 87 – 119, 1995.