Visual Features for Vehicle Localization and Ego-Motion Estimation

Oliver Pink, Frank Moosmann and Alexander Bachmann
Institut für Mess- und Regelungstechnik
Universität Karlsruhe (TH), 76128 Karlsruhe, Germany
{pink,moosmann,bachmann}@mrt.uka.de

***Abstract*—This paper introduces a novel method for vehicle pose estimation and motion tracking using visual features. The method combines ideas from research on visual odometry with a feature map that is automatically generated from aerial images into a *Visual Navigation System*. Given an initial pose estimate, e.g. from a GPS receiver, the system is capable of robustly tracking the vehicle pose in geographical coordinates over time, using image data as the only input.**

Experiments on real image data have shown that the precision of the position estimate with respect to the feature map typically lies within only several centimeters. This makes the algorithm interesting for a wide range of applications like navigation, path planning or lane keeping.

I. Introduction

A precise digital representation of the road network is crucial for autonomous navigation. At present, coverage of such high-precision digital maps is very low and mainly focused on some major cities, as the generation of these maps is costly and time-consuming.

Highly detailed aerial images, which are publicly available for virtually any region of the world, present a good alternative to manual map generation. In this paper, we present a novel method for vehicle localization by matching data from an on-board stereo camera rig to a digital feature map which is automatically created from aerial imagery. Except for one GPS measurement for position initialization, the method does not rely on other sensor data than on-board and aerial imagery.

As a main advantage over existing visual odometry and visual SLAM approaches, our method delivers a pose estimate not only relative to a local coordinate frame, but in geographical coordinates. Furthermore, problems of error integration over time and loop-closure problems are avoided by the use of the pre-built digital map.

A. Related Work

In recent years, research on the simultaneous localization and mapping (SLAM) problem has been brought from indoor applications to large-scale outdoor scenes [18], which makes them interesting for driver assistance applications. While originally SLAM methods based on range measurements from a laser range finder, recent work has also focused on developing camera-based approaches to SLAM [7][15][22].

For SLAM, it is desirable to obtain a robust estimate of the ego-motion, which is commonly realized with on-board inertial sensors. Current research on a field often referred to as visual odometry has shown the possibility of doing motion-estimation from an on-board camera only [20][19][12].

However, visual odometry suffers from the same problems than inertial-sensor based odometry, i.e. the vehicle pose error integrates over time. Similarly, a typical problem of SLAM approaches is the loop closure when an already mapped point is visited again. Furthermore, building the map from scratch is not suitable for long-distance path planning in large scale environments and building an entire road network from sensor data is very time-consuming.

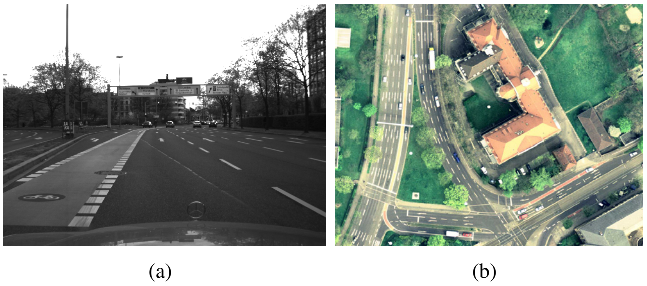
B. Objective Formulation

To overcome the limitations of existing SLAM approaches for large-scale outdoor scenes, we introduce a pre-built feature map of the environment instead of building up the map from on-board sensor data only. This feature map is generated from widely available geographically referenced aerial imagery and is matched to visual features from the on-board camera platform.

The basic idea of localization by matching on-board camera features to a global map has already been demonstrated in [21]. In this work, we will combine this idea with additional information from a visual odometry system for increased robustness and precision of the pose estimate.

Similar to conventional *Inertial Navigation Systems* which fuse motion estimates from inertial sensors with an absolute GPS pose estimate [2], the vehicle pose estimate from map matching will be fused with the vehicle motion estimates from visual odometry by a Kalman filter. Since our approach relies only on visual data from aerial and on-board cameras - except for one GPS measurement for pose initialization - we dub it a *Visual Navigation System*.

Figure 1 shows examples of the aerial images and on-board camera images that will be matched for vehicle localization.



(a) (b)

Fig. 1. Example vehicle camera image (a) and aerial image (b)

The remainder of this paper will detail the components of the *Visual Navigation System* system. Section II-A describes the generation of the digital map from aerial imagery. In

978-1-4244-3504-3/09/$25.00 ©2009 IEEE

section II-B, feature detection from on-board cameras for map matching and visual odometry is described. The actual vehicle pose estimation in section III is explained in two steps: matching the on-board features to the pre-built map is discussed in section III-A. The Kalman filter for data fusion with the underlying model for vehicular motion is introduced in section III-B. Experimental results of the overall system are given in section IV and section V concludes.

II. Image Processing

This section deals with the necessary processing of both on-board and aerial images. The images will be matched using some kind of feature that has to be extracted from both views. To distinguish between low-level image features like corners or edges and features for matching, the latter will be denoted landmarks in the following.

The landmarks have to be visible from both views and ideally should not have too large dimensions or perspective distortion. Our approach will make use of lane markings as landmarks, since they are clearly visible from both views and since their detection is well-discussed in literature [8][13][14]. However, the method is applicable to any other kind of landmark. Introducing other classes of landmarks may even improve the results further.

The landmark extraction from aerial images is detailed in section II-A. Both the landmark extraction and the determination of the vehicle motion from on-board camera images is described in section II-B.

A. Aerial Images

The landmark extraction from aerial images is performed by classification of each pixel whether it belongs to a lane marking or not. This is done by a Support Vector Machine classifier [9], which is trained by manually selecting some positive and negative samples from the aerial images.

The manual training and the comparably long computation times are tolerable, since landmark extraction from aerial images can be performed off-line, and only a very small training set is required to classify large map areas.

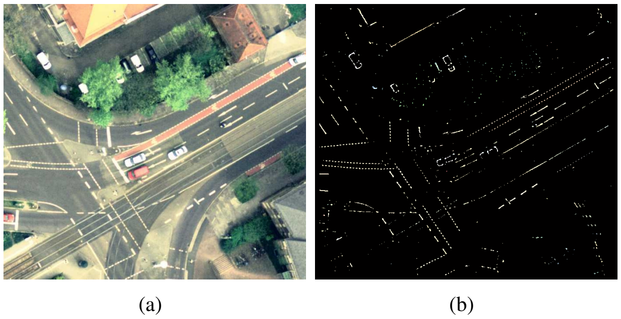
As the aerial imagery contains no height information, the classification result is a planar map, i.e. all detected features are assumed to lie in one ground plane.

Figure 2 shows a detail of an aerial image and the corresponding classification result.

The lane marking pixels are finally clustered to lane markings according to their distances and the centroid of each lane marking is stored in the feature map. Some additional properties like length and orientation of each marking are also stored for visualization purposes but will not be used for localization.

B. Vehicle Camera

The images from the on-board camera platform will serve two purposes. First, lane markings have to be detected and their 3D position has to be estimated for map matching. Second, the 3D vehicle motion has to be estimated from the 2D motion of image features.


(a)

(b)

Fig. 2. Original aerial image (a) and classification result (b).

For the latter, the image displacement, i.e. the 2D motion of image points between consecutive frames, is computed for a salient set of image points. These image points are determined using an interest point detector like e.g. the Harris corner detector [11].

Given the displacement, the motion components in 3D space are computed using depth information from a stereo image pair. In the following, it is assumed that the optical flow is determined from the right camera image and the disparity Δ is determined with respect to the right camera image. Furthermore, the camera platform is assumed to be fully calibrated and all image coordinates are assumed to be given in normalized coordinates, i.e. with focal lengths $f = 1$ and the image center located at $\mathbf{c} = (0, 0)^T$.

For a given 3D scene point $\mathbf{X} = (X, Y, Z)^T$, the corresponding position $\mathbf{x}$ in the image plane is

$$\mathbf{x} = \begin{pmatrix} y \\ z \end{pmatrix} = \Pi(\mathbf{X}) = \frac{f}{X} \begin{pmatrix} Y \\ Z \end{pmatrix}. \quad (1)$$

The depth information of the interest points can be recovered by measuring the horizontal separation of two corresponding points in a rectified stereo image pair.

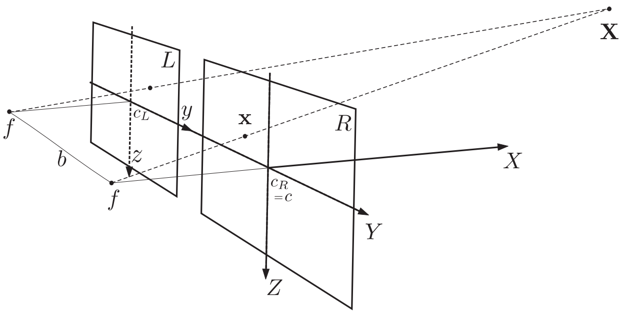


Fig. 3. The model of the stereo rig. The coordinate system of the right camera coincides with the global coordinate system. For any 3D scene point $\mathbf{X}$, its projection $\Pi(\mathbf{X})$ into the image plane can be described by eq. (1).

Given the image coordinates y_L in the left and y_R in the right image along an epipolar line, the disparity is related to the depth X of a scene point by

$$X \ = \ \frac{b}{\Delta}, \qquad (2)$$

where $\Delta = (y_L - y_R)$ is the disparity and b is the base length of the stereo rig.

Figure 3 shows the model of the stereo rig.

Assuming a rigid scene, the motion of all scene points is described by the same translational velocity $\mathbf{v} = (v_x, v_y, v_z)^T$ and rotational velocity $\mathbf{\Omega} = (\omega_x, \omega_y, \omega_z)^T$. With the Longuet-Higgins equations [16], the relation between the 6 DoF velocity and the 2D displacement can be described adequately. Defining the displacement of an image point as its motion across the image plane $(\dot{y}, \dot{z})^T = (u, v)^T$, the Longuet-Higgins equations for the case of translational and rotational rigid motion can be written as

$$\begin{aligned} \dot{y} &= \frac{\dot{Y}}{X} - \frac{Y}{X^2}\dot{X} \\ &= \left(-\frac{v_y}{X} - \omega_z + \omega_x z\right) - y\left(\frac{v_x}{X} - \omega_y z + \omega_z \cdot y\right) \ , \end{aligned} \qquad (3)$$

$$\begin{aligned} \dot{z} &= \frac{\dot{Z}}{X} - \frac{Z}{X^2}\dot{X} \\ &= \left(-\frac{v_z}{X} - \omega_x y + \omega_z\right) - z\left(-\frac{v_x}{X} - \omega_y z + \omega_z y\right) \ . \end{aligned} \qquad (4)$$

Substitution of equation (2) into equation (4) and separation of the translational and the rotational motion components delivers the 2D image displacement (u_i, v_i) of the i-th interest point for a given camera motion

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = \mathbf{H}_i \cdot (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)^T \qquad (5)$$

with

$$\mathbf{H}_i = \begin{bmatrix} -\frac{y_i \cdot \Delta_i}{b} & -\frac{\Delta_i}{b} & 0 & z_i & y_i \cdot z_i & -1 - y_i^2 \\ \frac{z_i \cdot \Delta_i}{b} & 0 & -\frac{\Delta_i}{b} & -y_i & 1 + z_i^2 & -y_i \cdot z_i \end{bmatrix} .$$

Stacking equation (5) for N interest points with displacement $(u_i, v_i)^T$ delivers a linear observation model with $2N$ equations

$$\begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_N \\ v_N \end{pmatrix} + \hat{\mathbf{e}} = \begin{pmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \\ \vdots \\ \mathbf{H}_N \end{pmatrix} \cdot \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} , \qquad (6)$$

which can be solved using straightforward least squares estimation

$$\begin{pmatrix} \hat{\mathbf{v}} \\ \hat{\mathbf{\Omega}} \end{pmatrix} = \left(\mathbf{H}^T \mathbf{H}\right)^{-1} \mathbf{H}^T \hat{\mathbf{u}} \ . \qquad (7)$$

However, the least squares estimate is very sensitive to outliers, which, in this case, are mainly due to violations of the rigid scene assumption. Other causes are e.g. errors in the disparity or displacement calculation. Especially for small disparities $\Delta \rightarrow 0$, the observation matrix H becomes unobservable. Therefore, interest points with a disparity below a certain threshold are discarded.

Furthermore, instead of using all interest points for the least squares estimate, a maximum set of N inliers is determined using the RANSAC estimator [10], which has proved to yield good results even with a large number of outliers present.

For the displacement estimation, the hierarchical Lukas-Kanade-algorithm ([17], [4]) is used. Disparity is computed by area based block matching ([6]). For the subsequent Kalman filtering (section III-B), the residuals of both estimation procedures are used to determine the measurement noise.

Figure 4 shows an example result for optical flow estimation and figure 5 shows an example disparity matching result. Only interest points with sufficiently low residuals and a disparity above the threshold are shown. Only the green interest points are used for least squares motion estimation, while the red points were rejected from the RANSAC algorithm.

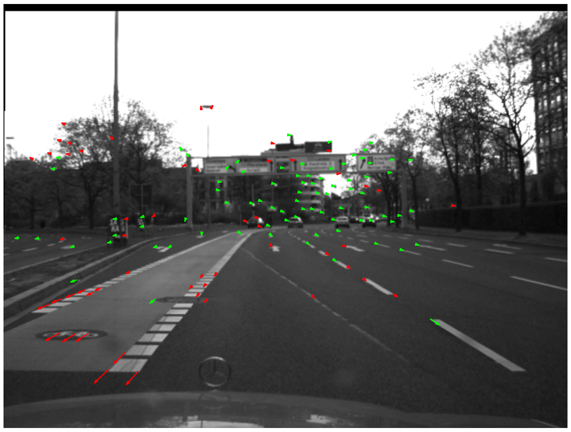


Fig. 4. Optical flow vectors for the detected feature points. Green points are used for ego-motion estimation.

The visual landmarks for map matching are yet to be determined. For this purpose, we will make use of the same combination of Harris interest point detection with disparity that was already used for motion estimation:

In the lower part of the camera image, the majority of the Harris Corner responses lies on corners of the lane markings. With equations (1) and (2), the 3D scene coordinates X for a given interest point with image coordinates $\mathbf{x} = (y, z)^T$ and disparity Δ are denoted by

$$\mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \frac{b}{\Delta} \cdot \begin{pmatrix} 1 \\ y \\ z \end{pmatrix} \ . \qquad (8)$$

Figure 6(a) shows an example corner detection result for the lower part of the image. Unlike in figure 4, corners that were rejected for motion estimation due to a high optical flow residual are also displayed.

Typically, up to four corners belong to one lane marking, thus the corners have to be clustered accordingly. This is

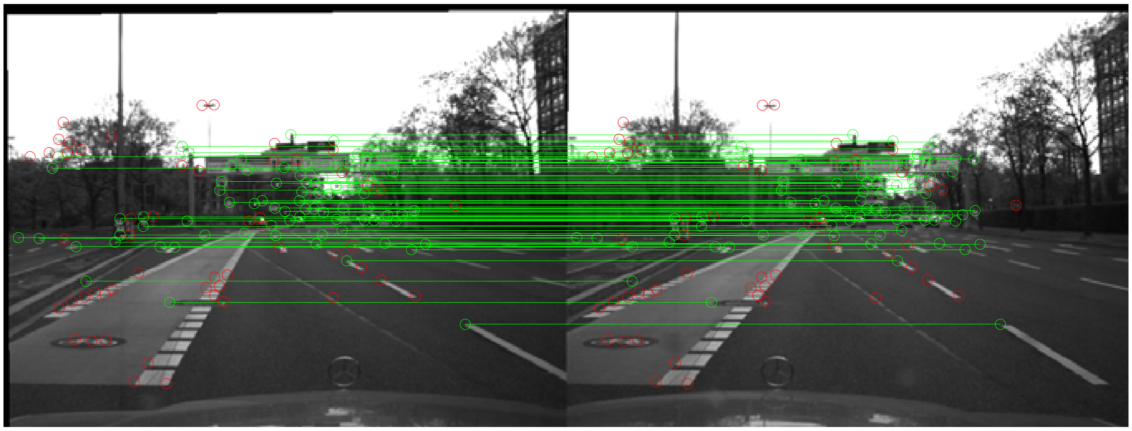
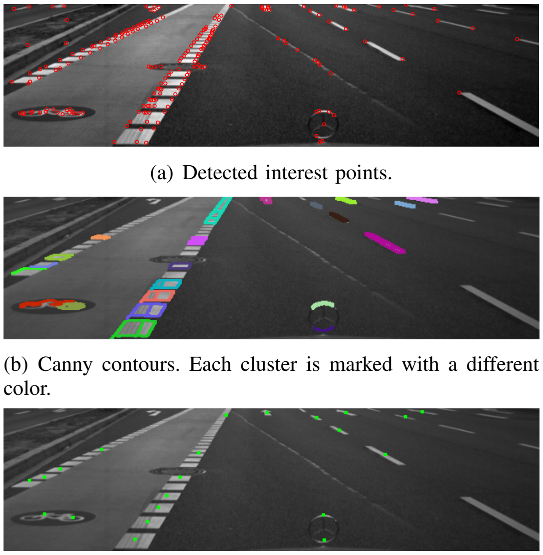


Fig. 5. Correspondences between left and right camera image. Green points are used for ego-motion estimation.



(a) Detected interest points.

(b) Canny contours. Each cluster is marked with a different color.

(c) Centroids of detected landmarks.

Fig. 6. Example landmark detection results.

accomplished by a Canny-like contour-tracing and computing the lane marking centroid from all clustered edge coordinates. Figure 6(b) shows an example result of the clustered edges and figure 6(c) shows the resulting centroids which will be used for matching.

III. Vehicle Pose Estimation

To obtain an estimate of the vehicle pose, the 3D coordinates of the lane markings from the camera view have to be matched to the lane markings in the digital map. This is a standard point pattern matching problem, for which many solutions were proposed in the recent years. These solutions mainly differ on their complexity and their robustness to outliers. We decided to use the iterative closest point algorithm [3] as a very fast and simple method for point pattern matching.

A major drawback of this method is that it only minimizes a local objective function, and therefore is very likely to stick to nearby local minima. It is therefore necessary to have a good initial position estimate. For vehicle pose tracking, the pose from the previous time step is in most cases sufficient, however for initialization, a single GPS measurement is needed.

Other methods like the ones proposed by van Wamelen et al. [23] or Caetano et al. [5] do not have these restrictions, however they require a large amount of memory and computing time to find the global optimum. Therefore, these methods are not suitable for real-time vehicle localization.

After having obtained an estimate for the current vehicle pose, the result is fused with the previous vehicle pose, vehicular motion constraints and with the vehicle motion estimate from visual odometry by a Kalman filter. The underlying system model is presented in section III-B. In the following, the vehicle pose estimation using the Iterative Closest Point method will be described in detail.

A. Iterative Point Matching

Given a set of m scene points in $\mathbb{R}^3$ from the digital map in world coordinates and n template points in $\mathbb{R}^3$ from the camera view in vehicle coordinates, the optimal transformation of the two coordinate systems has to be determined. The scene points are given as $\mathbf{S} = \mathbf{s}_1^W, \mathbf{s}_2^W, ...\mathbf{s}_m^W$ and the template points as $\mathbf{T} = \mathbf{t}_1^V, \mathbf{t}_2^V, ..., \mathbf{t}_n^V$ where the superscript W and V denote the respective vehicle or world coordinate system.

Assuming that the optimal transformation, i.e. the vehicle position $\mathbf{x}^W$ and orientation matrix $\mathbf{R}$ is known, the transformation of the scene points from world coordinates to vehicle coordinates is

$$\mathbf{s}_i^V = \mathbf{R}^{-1} \cdot \left(\mathbf{s}_i^W - \mathbf{x}^W\right) \ . \tag{9}$$

As stated before, for the iterative closest point method, the initial vehicle position has to be known up to some position and orientation error $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{R}}$. Given the initial pose estimate $\widehat{\mathbf{x}}^W$ and $\widehat{\mathbf{R}}$, the optimal transformation is

$$\mathbf{x}^W = \widehat{\mathbf{x}}^W - \widetilde{\mathbf{x}} \ , \tag{10}$$

$$\mathbf{R} = \widetilde{\mathbf{R}}^{-1}\widehat{\mathbf{R}} \ . \tag{11}$$

The estimated position of the scene points in vehicle coordinates is

$$\widehat{\mathbf{s}}_i^V = \widehat{\mathbf{R}}^{-1} \cdot \left(\mathbf{s}_i^W - \widehat{\mathbf{x}}^W\right) \ . \tag{12}$$

The goal of the ICP algorithm is now to recover the position and orientation errors $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{R}}$ given the pose estimate $\widehat{\mathbf{x}}^W$ and $\widehat{\mathbf{R}}$.

Assuming that for each template point $\mathbf{t}_i^V$, the corresponding scene point $\mathbf{m}_i^W = \mathbf{s}_j^W$ is known, the optimal rotation and translation can be determined by a translation of the centroid position and a rotation around the centroid which can be solved efficiently by a singular value decomposition (SVD) of the matrix

$$\mathbf{W} = \sum_{i=1}^{n} \mathbf{m}_i' \cdot \mathbf{t}_i'^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \ , \tag{13}$$

where $\mathbf{t}_i'$ and $\mathbf{m}_i'$ are the coordinates of the template and matched scene points relative to their respective centroid position:

$$\mathbf{t}_i' = \mathbf{t}_i^V - \sum_{k=1}^{n} \mathbf{t}_k^V \tag{14}$$

$$\mathbf{m}_i' = \widehat{\mathbf{m}}_i^V - \sum_{k=1}^{n} \widehat{\mathbf{m}}_k^V \ . \tag{15}$$

The desired rotation matrix is

$$\widetilde{\mathbf{R}} = \mathbf{U}\mathbf{V}^T \tag{16}$$

and the translation is

$$\widetilde{\mathbf{x}} = \sum_{k=1}^{n} \widehat{\mathbf{m}}_k^V - \mathbf{t}_k^V \ . \tag{17}$$

It is shown in [1] that this solution minimizes the sum of the squared residuals.

As the correspondences of the scene and template points are not known in advance, each template point is paired with the closest scene point, i.e. that minimizes the squared Mahalanobis distance

$$d^2(\mathbf{t}_i, \mathbf{s}_j) = (\mathbf{t}_i - \mathbf{s}_j)^T \cdot \mathbf{\Sigma}^{-1} \cdot (\mathbf{t}_i - \mathbf{s}_j) \ , \tag{18}$$

where $\mathbf{\Sigma}$ is the covariance matrix of the vehicle position estimate.

The point pairing and the computation of the rotation and translation errors $\widetilde{\mathbf{R}}^W$ and $\widetilde{\mathbf{x}}$ is repeated until the vehicle pose converges.

Finally, to refine the vehicle position estimate further, and to cope with outliers that are mainly due to false lane marking detection, the iterative closest point algorithm is repeated with only a smaller subset of the matches $\mathbf{m}_j$. This subset is determined by finding a maximum consensus for a rotation and translation error $\widetilde{\mathbf{R}}^W$ and $\widetilde{\mathbf{x}}$ by the RANSAC algorithm [10]. Again, the pairing and pose computation is repeated until the pose converges.

The combination of a full least squares optimization and a RANSAC optimization afterward has shown to give better convergence results in practice than a RANSAC optimization alone, while still having the robustness to outliers of the RANSAC algorithm. Especially for large initial pose deviations, only a small set of points is paired correctly. In these cases, RANSAC tends to reject the correct matches, not leading to any convergence at all. However, for a good initial pose estimate, the RANSAC stage is very well suited to reject all false correspondences.

B. Motion Tracking

Similar to conventional *Inertial Navigation Systems*, the *Visual Navigation System* obtains an estimate for the current vehicle pose by fusing vehicle motion estimates from visual odometry and the pose estimates from map matching in a Kalman filter structure. The underlying system assumes vehicle motion with constant velocity and yaw rate in the x,y-ground plane of the world coordinate system. Figure 7 illustrates the vehicle motion.

The vehicle pitch and roll angles as well as the camera pose are modeled as constant.

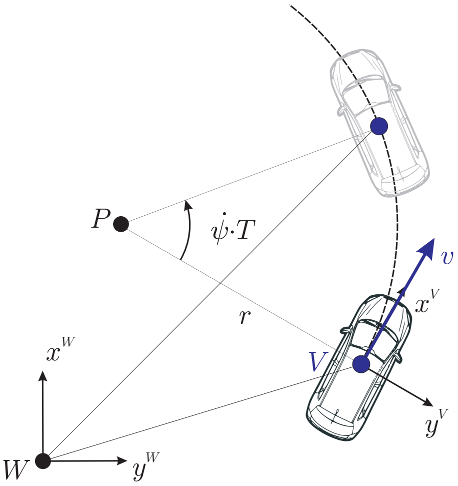


Fig. 7. Vehicle motion with constant yaw rate with instantaneous center of rotation P. The z-axis points into the image plane.

The vehicle position at time instant $k+1$ relative to the vehicle coordinate system at time instant k is given by

$$\mathbf{x}_{k+1}^V = \mathbf{x}_k^V + \left(\begin{bmatrix} 0 \\ r \\ 0 \end{bmatrix} + \mathbf{R} \cdot \begin{bmatrix} 0 \\ -r \\ 0 \end{bmatrix} \right) \ , \tag{19}$$

where $\mathbf{R}$ is the rotation matrix around the z-axis

$$\mathbf{R} = \begin{bmatrix} \cos(\dot{\psi} \cdot T) & -\sin(\dot{\psi} \cdot T) & 0 \\ \sin(\dot{\psi} \cdot T) & \cos(\dot{\psi} \cdot T) & 0 \\ 0 & 0 & 1 \end{bmatrix} \ . \tag{20}$$

Transformation to world coordinates gives the system model

$$\mathbf{x}_{k+1}^W = \mathbf{x}_k^W + \mathbf{R}_V^W \cdot \begin{bmatrix} \frac{v}{\dot{\psi}} \cdot \sin(\dot{\psi} \cdot T) \\ \frac{v}{\dot{\psi}} \cdot (1 - \cos(\dot{\psi} \cdot T)) \\ 0 \end{bmatrix} \ . \tag{21}$$

The velocity and rotation vectors $\mathbf{v}$ and $\mathbf{\Omega}$ from visual odometry (see section II-B) and the pose estimates in geographical coordinates $\mathbf{x}^W$, $\mathbf{R}$ from map matching (see section III-A) serve as measurements for the Kalman filter. The output of the Kalman filter is a vehicle pose estimate in world coordinates and a covariance matrix as quality measure for the estimate.

IV. Experimental Results

For experimental evaluation, the visual odometry results, the results from map matching and the fused pose estimates will be compared.

Figure 8 shows an overlay of the yellow lane markings from the feature map with an example sequence of vehicle camera images for the different pose estimates. All sequences were initialized with the same exact pose estimate at $t = 0$. Only every 25th frame is shown, i.e. the i-th row shows the results at $t = i$ seconds. Figure 9 shows all 50 pose estimates for the first 2 seconds relative to the initial pose[1].

Figures 8(a) and 9 clearly show the accumulation of a positioning error over time, when only integrating the motion estimates. After 2 seconds, the absolute accumulated position error after 50 frames is $0.56m$. On the other hand, the overlays from the map matching pose estimates in figures 8(b) match the real lane markings very well, except for some cases where a larger number of outliers is present. The image in the third row of 8(b) shows such an example. In figure 9, these cases can be noticed as blue dots that do not lie on the trajectory. These intermediate pose errors typically lie within $0.5m$ and can be filtered very well by the extended Kalman filter, as can be seen in 8(c). The map overlay matches the real lane marking position in all cases.

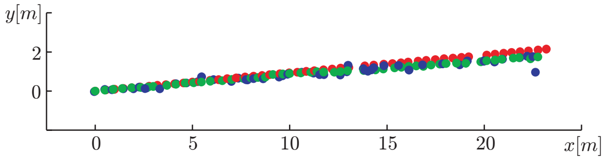


Fig. 9. Vehicle pose estimates for the first 50 frames (i.e. 2 seconds). The y-axis points to the east, the x-axis to the south. The units are meters relative to the initial vehicle pose. Red: Integrated vehicle motion estimates. Blue: Map matching pose estimates. Green: Filtered pose estimates

It is worth to notice that, according to our previous experiments in [21], map matching alone can cope with position errors of up to $2m$. In combination with visual odometry, the pose matching remains reliable even if the map matching fails for a longer sequence of frames, e.g. if no lane markings are present at all. As long as the overall accumulated pose error remains below $2m$, the first valid matching instantaneously yields a correct position estimate. A quality measure for the current pose estimate can be obtained directly from the covariance matrix of the underlying Kalman filter.

Finally, figure 10 shows an overlay of the original aerial image, the extracted yellow lane marking and the vehicle for the pose estimate at $t = 2s$. It clearly illustrates that the pose estimate in combination with the feature map can provide information about the environment even outside the current field of view. This information may be useful for other applications, e.g. determining drivable area or road curvature.

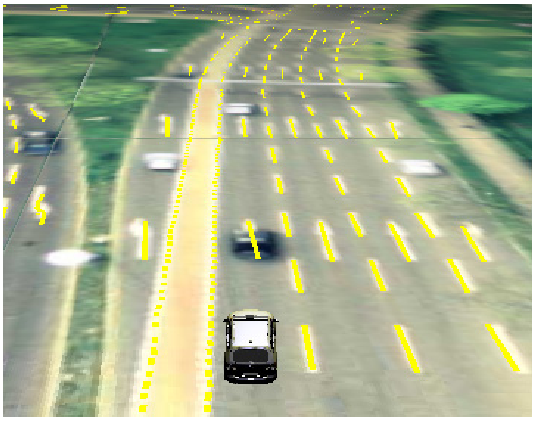


Fig. 10. Overlay of vehicle position, aerial image and lane markings (yellow) from the feature map.

V. Conclusion

We described a system that matches features from an onboard camera to a previously generated feature map to obtain a precise vehicle localization result. Robustness of the vehicle pose estimate is increased by fusing the localization result with an ego-motion estimate which is also obtained from the on-board camera platform. The resulting pose estimates are accurate within several centimeters with respect to the feature map even when a large number of false detections is present. Except for one pose estimate for initialization, the proposed *Visual Navigation System* system relies on images as the only input data.

Future work will evaluate possibilities of using globally optimal methods for map matching. However, these methods are comparably slow and are very likely not to be suitable for real-time vehicle applications. Alternatively, a globally optimal method could be used for initialization and the proposed iterative method for pose tracking. Meanwhile, initialization with GPS appears to be a reasonable alternative for most practical purposes.

VI. Acknowledgements

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

References

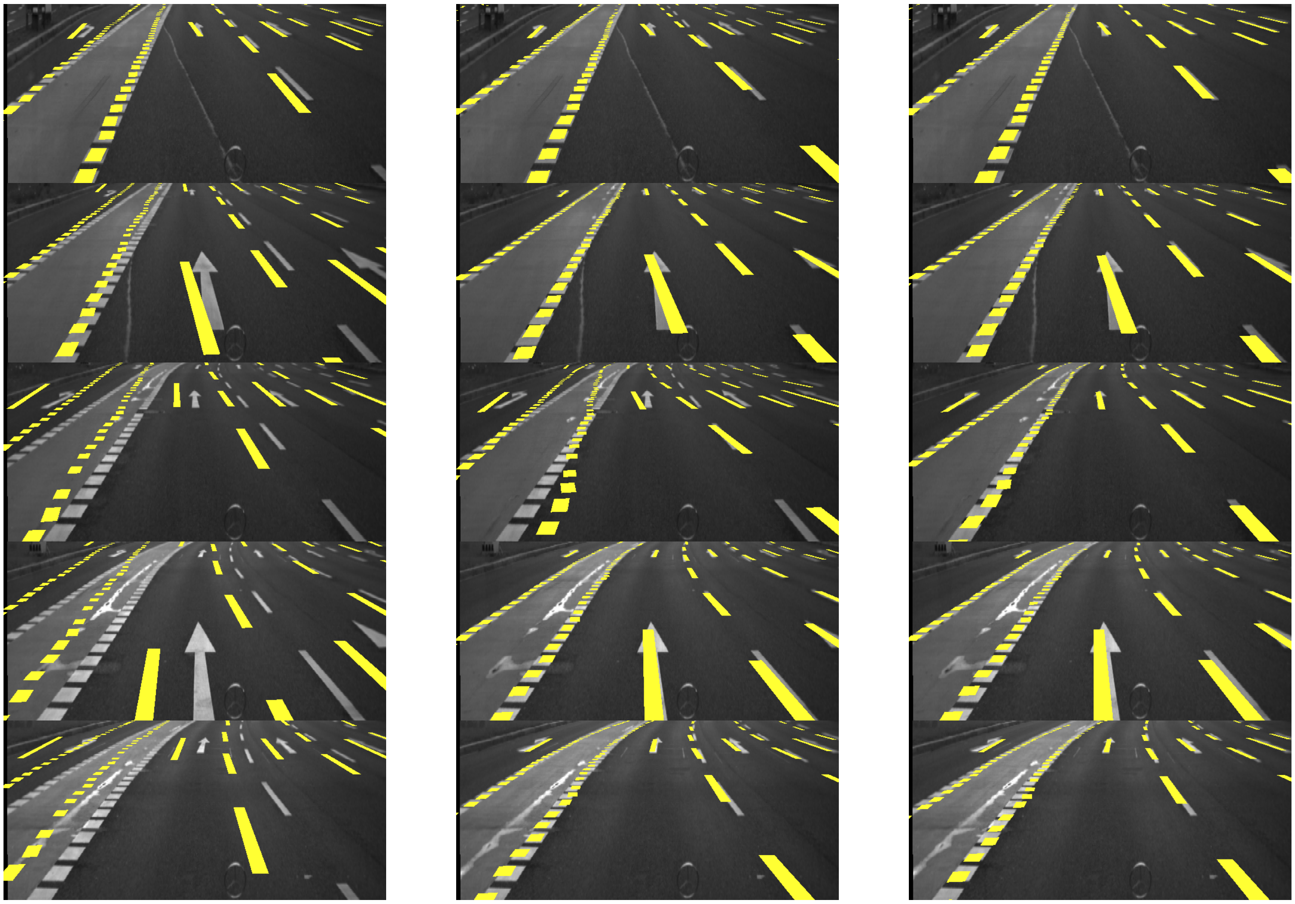
[1] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5):698–700, 1987.

[2] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li. *Estimation with applications to tracking and navigation*. Wiley-Interscience, 2001.

[3] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 14(2):239–256, 1992.

[4] J. Bouget. Pyramidal implementation of the Lucas Kanade feature tracker. Technical report, Tech. Rep. included in the OpenCV library, 2002.

[1] The coordinate system relative to the initial pose instead of geographical coordinates was chosen for reasons of comparability. Transformation to latitute/longitude using the geographically referenced aerial images is straigtforward.



(a) (b) (c)

Fig. 8. Overlay of the feature map (yellow) with vehicle camera images for different vehicle pose estimates and different time steps. The i-th row shows the pose estimate after i seconds, i.e. every 25th image is shown. Column (a): Integration of the vehicle motion estimates. Column (b): Unfiltered ICP matching results. Column (c): Filtered Pose estimate with vehicle motion and map matching result as measurement data.

[5] T. S. Caetano, T. Caelli, D. Schuurmans, and D. A. C. Barone. Graphical models and point pattern matching. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 28(10):1646 – 1663, Oct. 2006.

[6] T. Dang, C. Hoffmann, and C. Stiller. Self-calibration for active automotive stereo vision. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, Tokyo, 2006.

[7] A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 29(6):1052–1067, June 2007.

[8] C. Duchow. A novel, signal model based approach to lane detection for use in intersection assistance. In *Proceedings of the IEEE Intelligent Transport Systems Conference*, pages 1162–1167, 2006.

[9] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. Wiley, 2. edition, 2001.

[10] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, (24):381–395, 1981.

[11] C. Harris and M. Stephens. A combined corner and edge detector. In *The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[12] J. Horn, A. Bachmann, and T. Dang. Stereo vision based ego motion estimation with sensor supported subset validation. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2007*, pages 741–748. IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, June 2007.

[13] S.-S. Ieng, J.-P. Tarel, and R. Labayrade. On the design of a single lane-markings detector regardless the on-board camera's position. *Proceedings of IEEE Intelligent Vehicle Symposium 2003*, pages 564–569, 2003.

[14] Z. Kim. Realtime lane tracking of curved local road. In *Proceedings of the IEEE Intelligent Transport Systems Conference*, pages 1149–1155, 2006.

[15] T. Lemaire, C. Berger, I. Jung, and S. Lacroix. Vision-based SLAM: Stereo and monocular approaches. 74(3):343–364, September 2007.

[16] H. C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London, Series B*, 208:385–397, 1980.

[17] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *DARPA Image Understanding Workshop*, pages 121–130, 1981.

[18] R. Martinez-Cantin and J. Castellanos. Unscented slam for large-scale outdoor environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2005*, pages 3427–3432, 2-6 Aug. 2005.

[19] R. Munguia and A. Grau. Monocular SLAM for visual odometry. *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, pages 1–6, 3-5 Oct. 2007.

[20] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR) 2004*, 1:I–652–I–659 Vol.1, 27 June-2 July 2004.

[21] O. Pink. Visual map matching and localization using a global feature map. In *CVPR Workshop on Visual Localization for Mobile Platforms*, 2008.

[22] R. Sim, P. Elinas, and J. J. Little. A study of the rao-blackwellised particle filter for efficient and accurate vision-based slam. *International Journal of Computer Vision*, 74(3):303–318, 2007.

[23] P. B. van Wamelen, Z. Li, and S. S. Iyengar. A fast expected time algorithm for the 2-D point pattern matching problem. *Pattern Recognition*, 37(8):1699–1711, 2004.