

Urban Localization with Camera and Inertial Measurement Unit

Henning Lategahn, Markus Schreiber, Julius Ziegler, Christoph Stiller
Institute of Measurement and Control
Karlsruhe Institute of Technology
Karlsruhe, Germany
{henning.lategahn, stiller}@kit.edu, {schreiber,ziegler}@fzi.de

Abstract—Next generation driver assistance systems require precise self localization. Common approaches using global navigation satellite systems (GNSSs) suffer from multipath and shadowing effects often rendering this solution insufficient. In urban environments this problem becomes even more pronounced.

Herein we present a system for six degrees of freedom (DOF) ego localization using a mono camera and an inertial measurement unit (IMU). The camera image is processed to yield a rough position estimate using a previously computed landmark map. Thereafter IMU measurements are fused with the position estimate for a refined localization update. Moreover, we present the mapping pipeline required for the creation of landmark maps.

Finally, we present experiments on real world data. The accuracy of the system is evaluated by computing two independent ego positions of the same trajectory from two distinct cameras and investigating these estimates for consistency. A mean localization accuracy of 10 cm is achieved on a 10 km sequence in an inner city scenario.

I. INTRODUCTION

Future driver assistance systems ranging from next generation navigation systems to fully automatic driving require an accurate estimation of the ego position. Often, a centimeter level accuracy is sought. However, commonly used solutions of coupling a high precision GPS with IMUs may only reach these accuracies in open sky environments. In street canyon like scenarios shadowing effects and multipath propagation render these approaches infeasible. Moreover, this solution is prohibitively expensive for mass production.

Recently, methods that use a previously acquired map of some kind for ego localization are emerging [14], [16], [2], [18], [13]. First a detailed map of the environment is built from sensor data and stored for future use. Thereafter, online measurements are filtered by an estimator which outputs the ego position relative to the map. Localization is thereby independent of atmospheric disturbances and may even work in areas where satellite reception is completely blocked.

Despite its appealing properties these methods still require sensors like laser scanners [14], [16] which are both costly and bulky preventing a wide spread use.

Herein we present a method which uses a mono camera and an IMU to yield an ego position estimate relative to a previously recorded landmark map. Thus we refrain from using laser scanners and merely use visual information for localization. Our algorithm runs in real time on modest computing hardware and has been extensively tested on

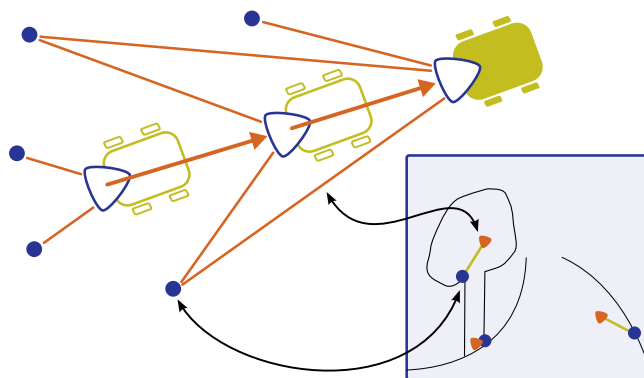


Fig. 1: A vehicle with a backward facing mono camera is shown. Landmarks (circles) are matched into the current image (shown on the right bottom) and back projection errors are minimized yielding a rough single shot pose estimate. IMU readings (arrows) are blended with past single shot estimates yielding the final result.

our test vehicle. Due to a lack of accurate ground truth trajectories we evaluate the accuracy of our method by comparing two independent estimates of the same trajectory using two independent cameras. The degree of consistency of these two distinct estimates serves as an approximation of the overall localization accuracy. We demonstrate an average localization accuracy of 10 cm in an inner city and partially rural scenario.

Section II reviews related work. Thereafter the mapping pipeline used for computing landmark maps is elucidated in Section III before introducing the localization estimator in IV. An experimental validation is presented in Section V and conclusions are drawn in VI.

II. RELATED WORK

The presented work is related to Simultaneous Localization and Mapping (SLAM) [4], [3], [17], [21], [8], [10], [15], [12] and Localization in general [14], [16], [2], [18], [13], [20].

SLAM methods try to solve the problem of a robot that maps an unknown area while at the same time localizing within this map. Much progress has been put forward to solving this elusive problem. Early works use extended Kalman filters and variants thereof [4], [3]. The filter state is comprised of the positions of a set of landmarks and the current

and optionally all past robot poses. Sensor readings can be fully predicted from such a state and compared to the true measurements yielding the required filter innovation.

Despite its theoretical soundness this branch of methods quickly hit an insurmountable complexity issue. Scalability could be improved by considering only subproblems at each time step and postponing a global update as long as possible. Pinies et al. present one such submapping method in [17].

At the same time a long known solution from photogrammetry experienced a renaissance: bundle adjustment. A state consisting of all robot poses and landmarks is sought that best explains the entire set of sensor measurements by means of nonlinear least squares estimate. Sibley and collaborators present a relative approach in [21].

Nowadays, a landmark-free approach is mostly followed for city scale problems of this sort. The state to be estimated consists of poses and estimated displacements (motion) between them. Once loop closures are introduced into the set of constraints the entire system becomes overdetermined and no state with zero residual can be found anymore. Non-linear least squares solvers using the Levenberg-Marquardt or Gauss-Newton method seek a minimum of the squared residual error. A 2D variant is presented in [8] and was later extended to 3D in [10]. Most pose graph approaches can be traced back to Lu’s and Milios seminal work [15].

Our work decouples mapping from localization and handles each step independently despite showing some influence of the aforementioned methods.

Recently, precomputed maps have been used for localization purposes. The obtained localization is always relative to a previously recorded map and global position accuracy therefore depends on the map accuracy. However, map relative accuracy is often the desired goal for e.g. path planning and the like.

A rotating laser scanner has been used by Levinson and co-workers as presented in [14]. The current laser scan is matched to a 3D point cloud map while also considering remittance of the laser beam as an extra measure. Moosmann and colleagues also use a scan matching approach in [16] reporting an improved localization accuracy over high precision GPS. Our method uses images of a mono camera for localization. The landmark map consists of much fewer landmarks and the sensor is less expensive.

Cameras have also been used by Badino in [1], [2]. Imagery was recorded for an urban area and holistic image features describing each single pose of the mapping trajectory are extracted from the images and stored as the map. During online localization current image features are matched to the map and position estimates are smoothed by fusing odometry information. The method is dubbed topometric localization. The final position estimate will always correspond to exactly one pose of the mapping trajectory.

Point features are computed from aerial imagery in [18] to create a sparse landmark map. Pink et al. then process an online mono camera image taken from street level, extract features and match these to the previously computed map for ego localization. The availability of an up-to-date and well

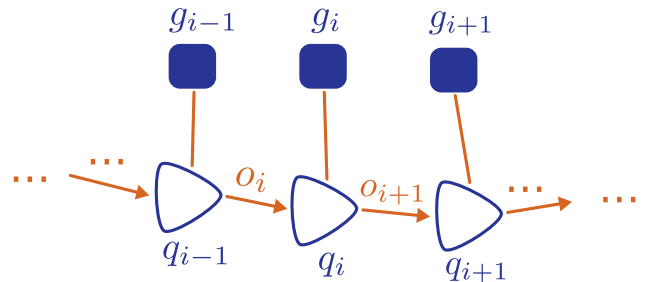


Fig. 2: First step of mapping pipeline: Triangles denote camera poses (backward facing in this case) and squares represent GPS readings. Solid variables are fixed during optimization. Edges between variables constitute constraints. Consecutive poses are e.g. constrained by visual odometry o_i .

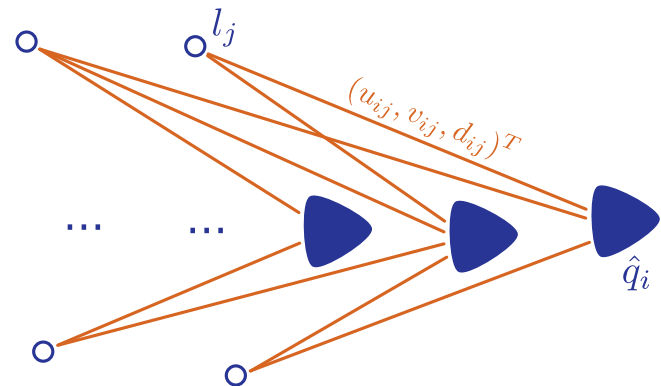


Fig. 3: Second step of mapping pipeline: Triangles denote camera poses which are kept fixed in this step and hollow circles denote landmark positions which are estimated. Poses and landmarks are interrelated by measuring their pixel position and stereo disparity $(u_{ij}, v_{ij}, d_{ij})^T$.

georeferenced aerial image however may become a problem. We have presented a preliminary version of our method in [13]. However the localization was computed from a single image without IMU integration. This method achieves a high accuracy in feature rich areas but robustness can be greatly improved by integrating IMU measurements. Furthermore a more robust image descriptor is used in this work. The mapping framework presented herein is more efficient and scales to almost arbitrary map sizes.

III. MAPPING

The creation of the landmark map is a crucial step in our processing pipeline and we present it next. Throughout the rest of the paper we assume a backward facing stereo camera setup. Moreover, we assume all sensors to be calibrated. In particular we assume the coordinate transform from camera to GPS receiver to be fully known [6]. For the sake of brevity we present all calculations in the camera coordinate system. For the mapping task, we record stereo data and high precision GPS for the area we wish to map. This data is thereafter processed by our tool chain in two steps. First, we line up our GPS readings and visual odometry to balance

a good georeference with good 3D landmark reconstruction ability. Secondly, we match image points across the entire sequence, reconstruct 3D landmarks from these tracklets and store the landmarks with associated image descriptors to disk. Both steps are explained in more depth next.

The recorded GPS trajectory is denoted by g_1, \dots, g_N with g_i representing the GPS reading for pose q_i . Poses are represented by 4×4 homogeneous matrices consisting of a 3×3 rotation matrix and a three dimensional translation vector. A motion estimate between any two consecutive poses q_{i-1} and q_i is computed from visual odometry [5] and denoted by o_i . Hence, one expects q_i to be the motion update o_i applied to q_{i-1} . At the same time however, the GPS readings g_{i-1} and g_i shall correspond to these two poses as good as possible as well. Thus, our system of constraints is overdetermined. We resort to nonlinear least squares (NLS) estimate for resolution.

NLS estimates are extensively used throughout this paper. To ease the readability we present the underlying sums of squared error terms whose minimizing argument is sought by factor graphs [9]. The graph of the NLS problem as stated above is depicted in Figure 2. The nodes of the graph are triangles denoting camera poses and squares representing the GPS readings. Nodes of the graph are the variables of the system of equations. An edge between nodes constitute a constraint each. Measurements may be associated with edges and edge labels are shown in this case (e.g. the visual odometry motion estimates o_i). Solid nodes always denote a fixed variable (one that is constant during optimization) whereas hollow ones are alterable (the argument of the sum of squared errors).

The graph of Figure 2 can be summarized by the error function

$$E_{\text{map1}}(q_1, \dots, q_N) = \sum_{i=2}^N \|(q_i \ominus q_{i-1}) \ominus o_i\|^2 + \sum_{i=1}^N \|g_i \ominus q_i\|^2 \quad (1)$$

where \ominus is an appropriate subtraction of the over parameterized poses (see e.g. [10]). The Norm $\|\cdot\|$ is a Mahalanobis norm whose Covariance matrices are neglected here for better readability. Note that the sums of equation 1 extend exactly over all edges of the graph. The minimizing argument of E_{map1} can be found by standard NLS machinery [19]

$$\hat{q}_1, \dots, \hat{q}_N = \arg \min_{q_1, \dots, q_N} \{E_{\text{map1}}(q_1, \dots, q_N)\} \quad (2)$$

and we utilize the g2o library [10] in this case. The pose estimates \hat{q}_i are kept fixed after optimization and are used during the second mapping step.

During the second step we aim for a set of 3D landmarks and their visual description computed from the recorded imagery. To this end we detect salient image points and associate these across all images. We refer to a set of pixel positions belonging to the same point in 3D as a tracklet. Every landmark that is stored in the final map is computed

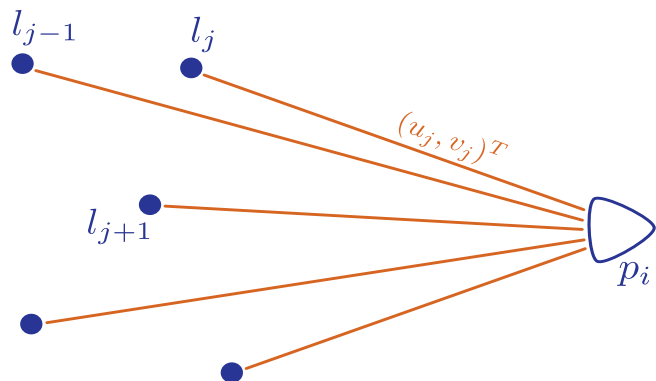


Fig. 4: First step of localization (single shot estimate): Landmarks are detected in the current camera image and their pixel position $(u_i, v_i)^T$ is determined. The pose p_i is varied such that the landmark back projection error is minimal.

from exactly one tracklet. Hence we obtain one pixel position and disparity (from stereoscopy) for a landmark j and a camera pose i and summarize it in the measurement vector $z_{ij} = (u_{ij}, v_{ij}, d_{ij})^T$. The 3D landmark position l_j can then be estimated by minimizing the sum of squared back projection errors. More concisely, we define the landmark error function to be

$$E_{\text{map2}}(l_1, \dots, l_M) = \sum_{ij} \|\pi(l_j, p_i) - z_{ij}\|^2 \quad (3)$$

with a camera projection function $\pi(p, l)$ that computes a pixel position and disparity from pose p and landmark l [7]. The minimizing argument of (3) is finally taken as the landmark position estimate. A simple heuristic removes landmarks that seem inappropriate from the map. The back projection error and track length of each tracklet is basically threshold for this purpose. We also try to keep landmarks close to the camera since these play an important role for longitudinal localization.

The NLS problem (3) can again be displayed as a graph. Figure 3 shows pose estimates \hat{q}_i which are fixed and landmark positions l_j (shown as circles) which are optimized. These are mutually interrelated by their respective measurements $(u_{ij}, v_{ij}, d_{ij})^T$.

Lastly, landmark positions are stored with their respective image descriptors forming the map. For every landmark one descriptor is stored for every pose that observed the landmark (see [13]). For a high robustness to illumination changes between mapping and localization we use our novel DIRD descriptor presented in [11].

IV. LOCALIZATION

The localization algorithm presented herein is a two step approach and yields a six degrees of freedom ego pose estimate. First, we query the map for landmarks close to the current ego pose, associate these landmarks with the current (mono) image and translate these landmark measurements into a rough ego position estimate which we refer to as single shot estimate. Second, a set of previous single shot estimates

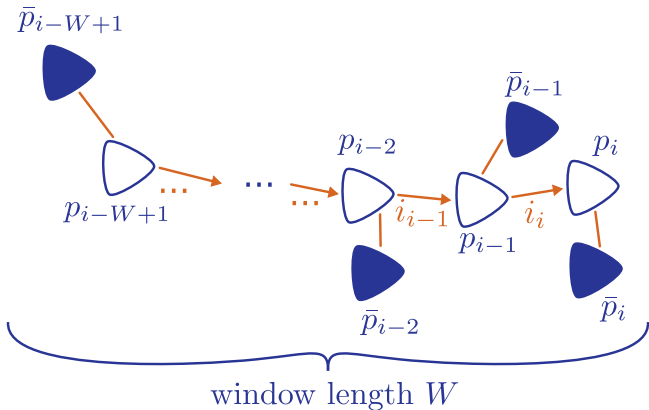


Fig. 5: Second step of localization (pose adjustment): Single shot estimates \bar{p}_i are balanced with IMU readings i_i to yield the final pose estimate. \bar{p}_i s serve as a prior in this step.

are fused with motion information from an IMU to yield the final ego pose. We will refer to this step as pose adjustment. Each step is elucidated detailedly next.

The pose of the camera that is the aim of our endeavor is denoted by p_i and we strive for an (intermediate) single shot estimate \bar{p}_i of it first. Thereto the map is searched for a pose q of the mapping trajectory that is closest to the current ego pose. All landmarks originally visible from q are then matched into the current camera image as in [13]. For the very first image of each run we use a GPS position to query the map but are completely independent thereof afterwards. By slightly overusing this notation we denote the pixel position of landmark j in the current image by $(u_j, v_j)^T$. An error function of sums of squared back projection errors is then defined by

$$E_{loc1}(p_i) = \sum_{ij} \|\pi(l_j, p_i) - (u_j, v_j)^T\|^2 \quad (4)$$

whose minimizing argument \bar{p}_i is found by NLS estimate. The corresponding factor graph is shown in Figure 4. Solid nodes again denote fixed variables whereas hollow ones represent variable ones. The pose p_i of the graph is the sole argument of the error function in (4).

Since (4) is essentially a quadratic error function it is highly susceptible to any outlying measurements which cannot be fully avoided in practice. Outliers may arise from false point matches or incorrectly estimated landmark positions during mapping. To handle these outliers we classify each measurement after optimizing (4) by examining its back projection error. If the error exceeds a given threshold it is flagged as an outlier. These are removed and a cleaned version of (4) is re-optimized. However, we allow for only half the measurements to be removed at most and always keep the “best” half of them in any case.

During a second pose adjustment step a set of W past single shot estimates $\bar{p}_{i-W+1}, \dots, \bar{p}_i$ are fused with motion information i_{i-W+2}, \dots, i_i from the IMU. The single shot estimate \bar{p}_i is now taken as a prior for the poses and combined with the IMU readings. All past W poses are

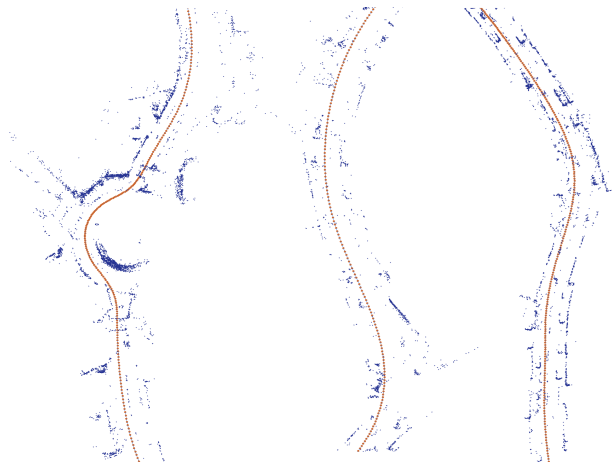


Fig. 6: Top view of three parts of the map. The trajectory of the mapping vehicle is shown in orange and landmarks are shown in violet. Landmark density varies depending on the environment. A traffic circle can be seen on the left.

estimated such that the error function

$$E_{loc2}(p_{i-W+1}, \dots, p_i) = \sum_{k=i-W+1}^i \|\bar{p}_k \ominus p_k\|^2 \quad (5) \\ + \sum_{k=i-W+2}^i \|(p_k \ominus p_{k-1}) \ominus i_k\|^2$$

is minimal. Note the great similarity to (1). The factor graph is shown in Figure 5. As before, covariance matrices of the Mahalanobis norm are omitted even though they are taken into consideration. The covariance matrices of the first sum are determined by the number of valid landmark associations of (4). Thereby unreliable single shot estimates are weakened hence increasing the overall accuracy.

The reason for separating both of these localization steps (single shot, pose adjustment) from each other is twofold. First, both estimates are very fast to compute thereby reducing computational burden if computed separately. Secondly, it is possible to add a robustification to the second estimates much like the first. We detect likely outliers of the single shot estimates and simply remove them from (5). Unlike filter like algorithms this detection is run on every time step and an outlier classification decision can always be revised in the future. Experiments on several hours of recorded data show that these outlier classifications are indeed revised quite often. No such decision is ever final thus contributing to the overall robustness.

V. EXPERIMENTS

Next, we present experiments on real world data. First the mapping results are furnished before showing localization results

We have mapped approximately 10km of an urban and partially rural area. The mapping trajectory was chopped into manageable pieces and the mapping algorithm presented above was applied to each chunk independently.

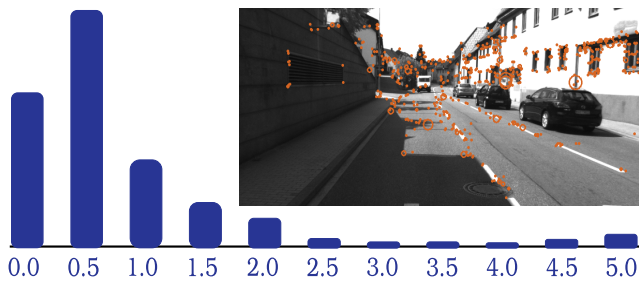


Fig. 7: A typical image of the mapping sequence. The landmarks that are successfully computed are marked by orange circles. Circle size corresponds to back projection error. A histogram of these back projection (pixel) errors is shown as well.

These sub maps overlap to avoid any landmark sparsity due to initialization or the like. Mapping is robust and takes approximately two hours for this map size without having put much effort into any parallelization. Furthermore, mapping is fully automatic and requires no manual intervention. We extracted roughly 610000 landmarks for this map size. Top views of some parts of the landmark map are shown in Figure 6. A traffic circle can be seen on the left. We observed that landmark density is particularly high in the urban areas whereas the rural parts are much sparser (middle of Figure 6). The map was created from a sequence recorded during noon on a sunny day. Figure 7 shows one frame with a histogram of squared back projection errors for that frame. The extracted landmarks are marked in the image.

To validate the robustness of the localization we have then recorded image data for the same area on a different day in the morning, at noon and in the evening. These sequences have been taken for testing. To provide a baseline for the capability of the method we have also added the mapping sequence to the set of test sequences. Thus the mapping sequence was also tested for localization from which we expect excellent results due to a perfect feature matching. In the sequel we will refer to these four test sequences as “morning”, “noon”, “evening” and “self”. The localization algorithm was run with these four test sequences. A window length of $W = 35$ (cf. (5)) has been chosen for all experiments.

For each of the four localization runs we have monitored the number of correctly associated landmarks per image. We report the numbers for the detected inlier set according to the single shot estimator (cf. Section IV). The results are depicted in Figure 8 as four violin plots. The mean number of point matches is marked by a small cross. Figure 8 shows a summary of the findings. Feature matching degraded significantly in areas of poor texture (rural) as it was expected. Matching for the “self” sequence was perfect. Thus the left plot of Figure 8 shows an upper bound with an average of 517 correctly associated landmarks per image. Moreover, matching worsens considerably from noon to evening to morning. Henceforth, illumination conditions are most difficult for the morning sequence in our case.

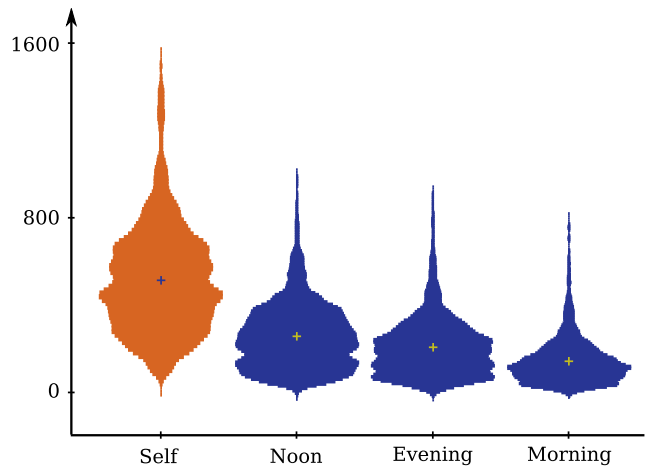


Fig. 8: Localization results: Violin plots (vertical histograms) of the number of successfully associated landmarks per image are shown. The orange plot (Self) shows results for localization with the mapping sequence to serve as an optimistic reference.

Next we present experiments for assessing the localization accuracy. Our attempts in finding an accurate enough reference GPS trajectory as ground truth has been without success. Partially the required accuracy was available but in areas of high GPS blockage (the areas we are mostly interested in) it was not. Furthermore, one may bear in mind that a high precision GPS trace is of no avail for assessing the localization precision if the mapping GPS trace was inaccurate in this area. The precision we are able to achieve with GPS and IMU is at best of the same magnitude as the one we achieve with our method. Hence, GPS traces cannot be used as ground truth as GNSS inaccuracies would heavily distort our findings. Thus, we report an accuracy measure which is map relative and not necessarily global. However, this is the desired accuracy oftentimes.

We have evaluated the residual of (5) after optimization. That is after pose adjustment we examine the translation error between the single shot estimates (\bar{p}_i , first step of localization) and the final pose estimate (p_i , second step of localization). The mean norm of this translation error for all time steps are shown in Figure 9. The residual increases with fewer landmark associations. For the “self” test set the mean (norm of) translation residual is approximately 5cm and increases to roughly 10cm for the “morning” dataset.

Finally, we have estimated a trajectory for every test dataset from two independent cameras thus obtaining two trajectory estimates each. These two estimates are compared to each other. Since the cameras are calibrated the offset can be subtracted and the results are compared. Figure 10 shows the translation difference (error) between two such estimates. Localization accuracy is consistent with the residual error analysis shown in Figure 9. Localizing with the image data that was used for mapping achieves an accuracy of around 5cm and increases to about 10cm on average for the morning sequence.

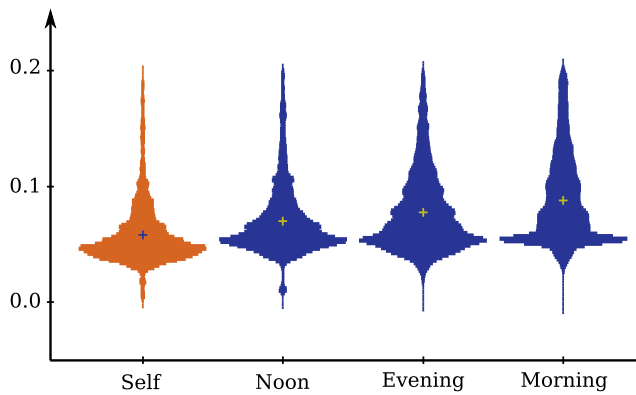


Fig. 9: Violin plots of the mean residual after pose adjustment (translation only) is shown in meters. See also Figure 8.

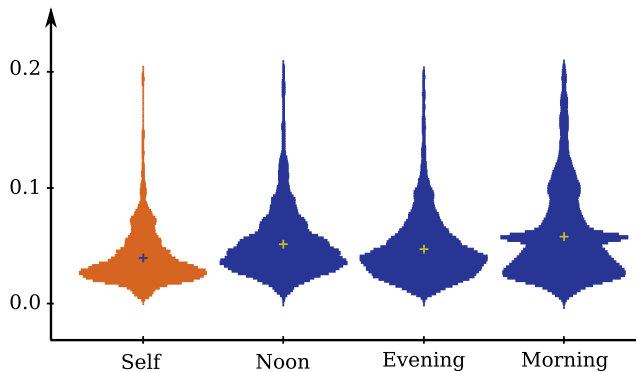


Fig. 10: Violin plots of the mean consistency between two distinct cameras of the same trajectory is shown in meters. See also Figure 8.

The algorithm runs with 3Hz on a compact automotive computer and reaches roughly 10Hz on a better equipped computer with image processing being the main bottle neck. However, we still see much room for improvement here.

VI. CONCLUSION AND FUTURE WORK

Herein we have presented a method for robust six degrees of freedom ego localization using a visual landmark map. The creation of the map from recorded imagery and GPS traces has been presented. During online localization the map is queried, landmarks are matched with salient points of the current camera image and converted into a single shot pose estimate. These single shot estimates are combined with IMU data during a second optimization step yielding the final result. The algorithm runs robustly in real time. Comparing the estimated trajectories of a single run from two independent cameras has been the metric of choice to assess the accuracy of the presented approach. Both trajectories are consistent to within 10 cm on average. First experimental results of varying illumination conditions between mapping and localization are encouraging.

Automatically augmenting the landmark map with vehicle relevant infrastructural information like curb stones, lanes, traffic lights etc. seems an obvious next step which we are currently working on. Thereby we hope to shift these

computationally demanding tasks from an online to an offline computation. Furthermore, we believe that finally dropping the dependency on the initial GPS position needed for jump starting the localizer and the IMU for constraining the motion is a fruitful direction of future research.

REFERENCES

- [1] H. Badino, D. Huber, and T. Kanade. Visual topometric localization. In *Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, June 2011.
- [2] H. Badino, D. Huber, Y. Park, and T. Kanade. Real-time topometric localization. In *International Conference on Robotics and Automation (ICRA)*, St Paul, Minnesota, USA, May 2012.
- [3] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *Robotics & Automation Magazine, IEEE*, 13(3):108–117, 2006.
- [4] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1052–1067, 2007.
- [5] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 963–968. IEEE, 2011.
- [6] Andreas Geiger, Frank Moosmann, Omer Car, and Bernhard Schuster. Automatic camera and range sensor calibration using a single shot. In *International Conference on Robotics and Automation (ICRA)*, St. Paul, USA, May 2012.
- [7] R. Hartley and A. Zisserman. *Multiple view geometry*, volume 642. Cambridge university press Cambridge, UK, 2000.
- [8] K. Konolige, G. Grisetti, R. Kummerle, W. Burgard, B. Limketkai, and R. Vincent. Efficient sparse pose adjustment for 2d mapping. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 22–29. IEEE, 2010.
- [9] F.R. Kschischang, B.J. Frey, and H.A. Loeliger. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2):498–519, 2001.
- [10] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3607–3613. IEEE, 2011.
- [11] H. Lategahn, J. Beck, B. Kitt, and C. Stiller. How to learn an illumination robust image feature for place recognition. In *IEEE Intelligent Vehicles Symposium*, Gold Coast, Australia, 2013.
- [12] H. Lategahn, A. Geiger, and B. Kitt. Visual slam for autonomous ground vehicles. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1732–1737. IEEE, 2011.
- [13] Henning Lategahn and Christoph Stiller. City gps using stereo vision. In *IEEE International Conference on Vehicular Electronics and Safety*, Istanbul, Turkey, Juli 2012.
- [14] J. Levinson and S. Thrun. Robust vehicle localization in urban environments using probabilistic maps. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4372–4378. IEEE, 2010.
- [15] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4(4):333–349, 1997.
- [16] Frank Moosmann and Christoph Stiller. Velodyne SLAM. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 393–398, Baden-Baden, Germany, June 2011.
- [17] P. Piniés and J.D. Tardós. Large-scale slam building conditionally independent local maps: Application to monocular vision. *IEEE Transactions on Robotics*, 24(5):1094–1106, 2008.
- [18] O. Pink. Visual map matching and localization using a global feature map. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–7. IEEE, 2008.
- [19] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge University Press, 2007.
- [20] M. Schreiber, C. Knoepfel, and U. Franke. Laneloc: Lane marking based localization using highly accurate maps. In *IEEE Intelligent Vehicles Symposium*, Gold Coast, Australia, 2013.
- [21] G. Sibley, C. Mei, I. Reid, and P. Newman. Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment. *The International Journal of Robotics Research*, 2010.