vol 15, no. 3, June 2014, pp. 1246-1257

Vision Only Localization

Henning Lategahn and Christoph Stiller, *Senior Member, IEEE*

***Abstract*—Autonomous and intelligent vehicles will undoubtedly depend on an accurate ego localization solution. Global navigation satellite systems (GNSS) suffer from multipath propagation rendering this solution insufficient.**

Herein we present a real time system for six degrees of freedom (DOF) ego localization that uses only a single monocular camera. The camera image is harnessed to yield an ego pose relative to a previously computed visual map. We describe a process to automatically extract the ingredients of this map from stereoscopic image sequences. These include a mapping trajectory relative to the first pose, global scene signatures and local landmark descriptors. The localization algorithm then consists of a topological localization step that completely obviates the need for any global positioning sensors like GNSS. A metric refinement step that recovers an accurate metric pose is subsequently applied. Metric localization recovers the ego pose in a factor graph optimization process based on local landmarks.

We demonstrate a centimeter level accuracy by a set of experiments in an urban environment. To this end, two localization estimates are computed for two independent cameras mounted on the same vehicle. These two independent trajectories are thereafter compared for consistency. Finally, we present qualitative experiments of an augmented reality (AR) system that depends on the aforementioned localization solution. Several screen shots of the AR system are shown confirming centimeter level accuracy and sub degree angular precision.

***Index Terms*—camera, localization, GPS, landmark, bundle adjustment, nonlinear least squares, SLAM**

I. Introduction

THE next generation of intelligent transportation systems will heavily depend on an accurate self localization in a multitude of situations. Future navigation systems may show infrastructural information by an augmented reality (AR) system. Furthermore, it is largely agreed that a high precision self positioning system is a crucial prerequisite for fully and semi-automatic driving. Finally, a plethora of comfort and safety functions can be imagined once ego localization is coupled with maps.

It has been shown that automated driving can be simplified severely with static maps [17]. Their information may include centerlines of each lane, admissible maneuvers at intersections for each lane, position and validity of traffic lights, precedence and traffic rules. Hence, rule-compliant behavior generation may vastly be encoded in the map rather than inferred trough complex scene understanding [2] or artificial intelligence methods. The vehicle relative position of static obstacles can be retrieved easily from the ego position at any time. Thereby the on board environment perception can be moved to an offline computation hence exonerating electronic control units from computationally demanding tasks. Furthermore, the "sensing range" of such static map objects is literally unbound. All of the aforementioned methods share a common dependency on a centimeter level accurate self localization. Common approaches of applying global navigation satellite systems (GNSS) [37] to this problem are infeasible. Integrated navigation systems (INS) consisting of state of the art dual frequency GPS receivers often coupled with high precision inertial measurement units (IMUs) are prohibitively expensive. Moreover, these INS reach the sought precision unreliably and only under good geometric constellations in open-sky environments. Multipath propagation and shadowing effects common in inner city and street canyon like scenarios often render this approach impossible. Low-cost single frequency receivers found in mass production vehicles suffer from these problems even more. Their accuracy is at least one order of magnitude worse than those of INS [29]. Recent advances in localization by spinning laser scanners seem algorithmically

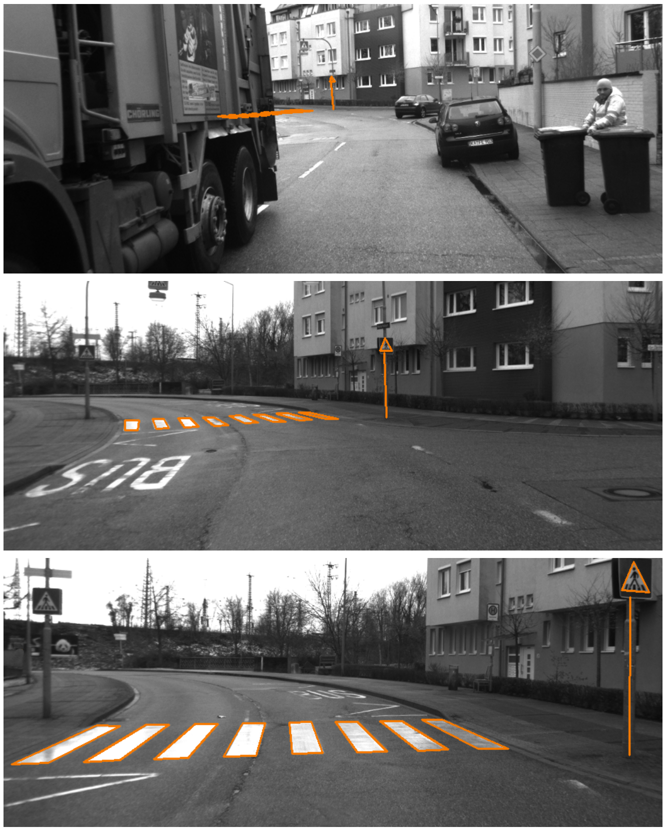

Fig. 1. The localization method allows to overlay infrastructural information of the map onto the current camera image after localizing the camera. The resulting augmented reality system demonstrates centimeter level accuracy of the proposed vision only localization method. A pedestrian crossing is reliably shown even when occluded by the truck. The "sensing distance" is unbound and does not increase computational load.

The authors are with the Institute of Measurement and Control Systems, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany (e-mail: henning.lategahn@kit.edu, christoph.stiller@kit.edu).

appealing [24], [28]. However, these scanners are expensive and bulky hence hampering their wide spread use in series transportation systems.

Herein we present a system for high precision ego localization using only a single camera. The system does not depend on any external hardware like GPS receivers, IMUs, odometers or the like. The vehicle is localized relative to a previously computed visual map. The map is computed fully automatically from a mapping trajectory with recorded stereo. The mapping process does not depend on external hardware and is computed from stereopsis alone. This map can be manually enhanced by additional infrastructural objects relevant to the vehicle. Pedestrian crossings, lane markings, traffic lights and signs are a few non exhaustive examples. At each time step these additional objects in the immediate vicinity of the car are made available to the driver or vehicle internal systems. Examples of AR enhanced camera images are shown in the experiments Section VI and in Figure 1.

The localization system operates in two stages. First the vehicle is localized topologically in the graph of map poses for initialization purposes. Thereafter a metric localization is computed. The metric ego pose estimate has six degrees of freedom (DOF) and reaches a linear accuracy of a few centimeters and sub degree angular precision. This accuracy is sufficient for the above mentioned functions and systems. We present extensive experiments evaluations to asses the presented method. The full system runs in real time (10Hz) on a modest laptop computer.

Section II reviews related work. Thereafter the mapping process is highlighted in Section IV before introducing the mathematical notations in Section III. The localization is elucidated in Section V. Experiments are presented in Section VI and conclusions are drawn in Section VII.

II. Related Work

The work we present herein is strongly related to simultaneous localization and mapping (SLAM) [5], [6], [9], [10], [14], [19], [21], [26], [30], [35] from which we draw many inspirations, localization in general and map relative localization [7], [11], [22]–[25], [28], [31], [33] in particular. Beyond that, it is related to place recognition [3], [4], [8], [20], [27], [38] sometimes referred to as loop closure detection in the SLAM literature.

SLAM is the long known problem of localizing a robot within a map while computing that map at the same time. Localizing is enabled by known maps while map generation depends on a localization solution. Most often maps are represented by a collection of landmarks which are sensed by some sensor. Bayesian filters like extended Kalman filters (EKFs) aim at estimating a state comprised of all landmarks and the current ego position [5], [10]. Sensor readings can be fully predicted from one such state vector and compared to the actual sensor output to yield the filter innovation.

Despite its theoretical soundness the filter approaches as stated above suffer from well known Draconian limitations in scalability preventing large scale real time systems. Discovering a special structure of the state covariance matrix which exhibits strong correlations only between landmarks that have been sensed jointly eventually led to submapping approaches. Only a small fraction of the state vector and covariance is updated at each time step and a global update is postponed as long as possible (e.g. until a loop closure occurs). These methods have constant complexity most of the time. We exemplary mention the work of Pinies and co-workers [30]. Their solution is numerically equivalent to the regular EKF formulation after every global update and does not necessitate any approximations.

A long known solution from photogrammetry experienced a resurgence of interest once computing power had increased: bundle adjustment. All robot poses and all landmark positions are stacked into one joint state vector which is estimated by nonlinear least squares estimation. Levenberg-Marquardt and Gauss-Newton methods are popular choices of solvers. The measurement matrix of the linearized system of equations which is iteratively solved exhibits an extremely sparse structure. The emergence of sparse matrix solvers using variable reordering [1] finally led to the breakthrough. A relative representation of the problem dubbed relative SLAM is presented in [35].

Nowadays the landmark/pose notion of SLAM is replaced by a simple pose-only surrogate. The state of the map consists of all poses of the robot trajectory and the motion induced pose graph is estimated. Once loop closures are introduced, the system of pose to pose constraints becomes overdetermined. It is finally solved by standard nonlinear least squares machinery. The removal of landmark positions from the problem allows to estimate very large trajectories (e.g. [9]). Another powerful system is presented by Bosse et al. in [6] which applies the Atlas framework to create large cyclic maps from laser range finders. A fine introduction into the subject of pose graph optimization is presented in [14]. A flexible open source software library is presented in [19]. Most approaches of solving pose graphs can be traced back to the influential work of Lu and Milios [26].

The method we present herein also solves for landmarks and ego poses from measurements. However, our map is kept fixed after its creation. Thereby, we achieve massive computational savings. Nevertheless, our approach shows some resemblance to the aforementioned branch of algorithms.

Following a similar argumentation a 3D point cloud map with localization capabilities has been computed in [28]. Scans of a spinning laser are registered by a iterative closest points method with high accuracy. This point cloud map can thereafter be used for localization purposes by using the same laser again.

Clutter, moving objects and noise may cause some difficulties for laser scanners of this type. Levinson and collaborators have therefore proposed to use infrared remittance values of laser beams of the road surface only [24], [25]. The road surface can be found rather accurately in laser point clouds and are likely to be persistent over time. SLAM approaches are used to smooth the map and enforce consistency in areas of self-overlap whereas particle filters are their choice of localization estimator.

Laser scanners of the type used in both [28] and [24] are prohibitively expensive on the one hand and cause severe packaging problems on the other. Hence, their use in series production vehicles is inadmissible. The recent explosive growth of imaging technologies offers a solution and cameras are used by Pink in [31]. An aerial image of the area of interest is preprocessed and searched for lane markings and salient road surface features. Thereafter online camera images are matched to this kind of feature map by point registration algorithms. Finally, ego pose estimates are coupled with a motion model for a refined ego pose estimate. Schreiber et al. [33] exploit lane markings and cameras for map relative localization. Fang and colleagues propose a pure vision based localization for a slow moving passenger vehicle in [11]. A mono camera is attached underneath the vehicle and supported by additional illumination. Key points of the ground texture are used in a mapping process. A large optimization yields the spatial position of these points. During re-localization these points are matched and the resulting estimated is fused in an Unscented Kalman Filter (UKF). Their approach differs from ours as they use controlled lighting conditions and a very slow motion. The proposed sensor setup might be inappropriate for regular urban driving speed (motion blur). Yet another interesting approach is proposed by Courbon and collaborators in [7]. A map dubbed visual memory is computed from an initial survey trajectory using a fish eye camera. Subsequent traversals can be performed autonomously by following the initial path. The camera image is matched to the map hierarchically. First a global search is performed and thereafter refined locally using landmarks. Their hierarchical approach shows some resemblance to ours despite some pronounced differences. Our approach allows to estimate the 3D positions of the landmarks with presumably higher accuracy due to the use of stereo vision during mapping. Their method is evaluated at low speeds over a track of 750 meters whereas we perform experiments over much larger distances with regular driving speeds.

Despite the appealing properties of this research direction the required accuracy and availability of such aerial images is not always guaranteed. An alternative is introduced in our earlier works [22], [23]. The area to be mapped is traversed with high precision GPS and stereo cameras and landmark maps are computed. During online operation the map is queried and landmarks are associated with salient points of the current camera image. These landmark associations are used in the final ego pose estimators. However, these two methods strongly depend on both GPS and IMU for mapping or localization. The approach presented in this article liberates the estimator from this restrictive requirement and all dependencies on external hardware are dropped completely yielding a truly vision only system.

One issue needed to be resolved is place recognition to initialize the localizer. Place recognition is frequently addressed when detection of loop closures is important. The work horse of loop closure detection is the method of Cummins et al. [8] dubbed FAB-MAP which applies an appearance only based approach. A probabilistic model of places is learned from salient image features. Large feature vocabularies need to be trained beforehand. Their work is robust to perceptual aliasing albeit being computationally rather expensive due to its feature extraction process.

To mitigate the effects of visually describing a multitude of image features in every image Sünderhauf and co-workers [38] have resorted to a simplistic approach. The image is down sampled and partitioned into small equally sized image tiles each of which is holistically described by only one single image descriptor. Concatenating single tile features into one yields the descriptor representing the appearance of the entire image. Place recognition is thereafter straight forwardly achieved by nearest neighbor search in the space of appearances.

Badino et al. [3], [4] have followed a similar idea of describing the entire image by whole image SURF features. Their place recognizer was designed with topological localization in mind. Odometers, image and laser range finder features are fused in a histogram filter to yield the nearest pose of a previously recorded mapping trajectory during online operation. A good robustness was shown even across seasonal changes.

The aforementioned methods only seem to be the tip of the iceberg in the realm of holistically describing images for place recognition. Milford [27] has pushed the idea further by describing panorama images by only a few bits. Place matches are computed for double round trip trajectories of lengths up to 70km. The dynamic time warping of the pairwise image difference matrix appears to be a crucial ingredient. In fact, our method shows some resemblance to it.

A preliminary sketch of the dynamic programming procedure of Section V-A is briefly introduced in the appendix of our earlier work [20]. Herein, we borrow some ideas from it and extend it to an online light weight topological localization method which is used during initialization.

III. Notations and Symbols

Before delving into the details of the mapping tool chain and localization algorithm we need to slide in a short section on notations. Throughout the rest of the article we assume poses to be parameterized by 4×4 homogeneous matrices

$$p = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} \in SE(3) \tag{1}$$

with 3×3 rotation matrix $\mathbf{R}$ and 3×1 translation vector $\mathbf{t}$. Moreover, the chart $\phi(\cdot)$ maps from this over parameterized manifold $SE(3)$ into $\mathbb{R}^6$ the minimal parameterization of 3D angle and 3D translation vector [16]. The motion operator $\oplus$ which applies a motion $\delta \in \mathbb{R}^6$ to pose $p \in SE(3)$ is then defined by

$$\begin{aligned} p_2 &= p_1 \oplus \delta && (2) \\ &= p_1 \cdot \phi^{-1}(\delta) \in SE(3). && (3) \end{aligned}$$

Conversely, the subtraction of two poses yields a change by

$$\begin{aligned} \delta &= p_2 \ominus p_1 && (4) \\ &= \phi(p_1^{-1} \cdot p_2) \in \mathbb{R}^6. && (5) \end{aligned}$$

A good introduction into this subject can be found in [16] and [36].

Furthermore, both the mapping and localization algorithm uses the notion of landmarks which are natural 3D points which

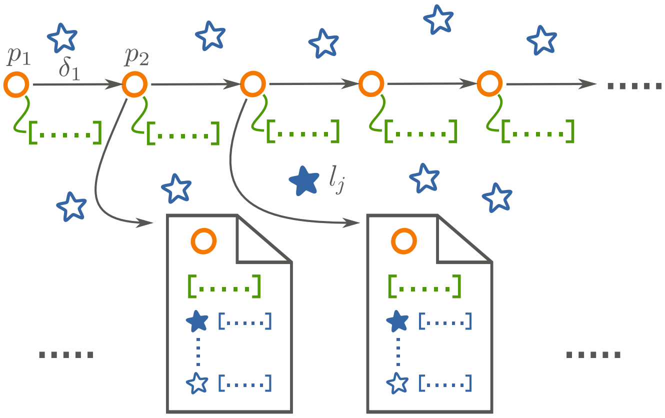

Fig. 2. A simplified summary of the mapping process is shown. First the poses $p_1,\ldots,p_N$ of the map are estimated using visual odometry estimates δ_i. Thereafter the landmark positions l_j are computed in map coordinates denoted by stars. A holistic feature describing each pose is computed. These features are denoted by brackets with dots in green. Finally, one map file is created for every pose. It contains the pose, all landmarks visible from that pose and the holistic vector. The solid star exemplary illustrates that it is stored redundantly in two files. The landmark feature vector (blue brackets with dots) for that landmark depends on the pose it is observed from.

are detected by the vision system. Landmarks are denoted by $l_j \in \mathbb{R}^3$ and are always three dimensional.

Once a landmark is observed from any pose it induces a measurement which we summarize in the variable z. It consists of pixel positions of the observing camera. The function that maps a (known) landmark l into the image plane of a (known) pose p is denoted by $\pi(l,p)$ and depends on camera parameters such as focal length, principal point and the like. We assume a pine hole camera model [15]. It follows that in the ideal case (without any noise) $z=\pi(l,p)$ would hold.

IV. Mapping

Next we present the mapping pipeline required to compute the visual map. We record stereo data for the area we wish to map. In the sequel we will refer to this traversal as the mapping trajectory.

We postpone all localization details until Section V despite shortly sketching the requirements on the visual map from the localization perspective first. During online localization 3D landmarks of the immediate vicinity of the ego vehicle are retrieved from the map and associated with the current camera image. These landmarks associations are harnessed to yield an ego pose estimate. Hence the visual map needs to contain a set of landmarks with associated visual descriptors for robust association.

The nearest pose of the mapping trajectory needs to be found during an initialization phase of the localization algorithm. To this end, we compute a single holistic feature vector for each image of the mapping trajectory. Throughout the rest we will refer to these features as holistic features (as apposed to landmark features). During localization the (spatially) nearest pose can be found by computing the nearest pose in the space of holistic feature vectors.

Furthermore, the map needs to store its data efficiently since its amount easily eclipses available primary memory capacities.

From the aforementioned preamble we derive the requirements on the map which also serves as a road map for the next paragraphs: the map needs to contain 3D landmarks with associated landmark features, each pose of the mapping trajectory needs to be augmented by one holistic feature vector for initialization purposes and finally the data structure of the map needs to be stored efficiently on secondary memory. Figure 2 shows an overview.

A pose of the map is now denoted by $p_i \in SE(3)$ with $i \in \{1,\ldots,N\}$ and N being the number of poses/images of the mapping trajectory. We spatially discretize poses to be no closer to each other than 50 centimeters. Due to the lack of any global positioning system we fix the first pose p_1 to the origin and successively estimate the pose p_{i+1} from p_i by setting $p_{i+1} = p_i \oplus \delta_i$ where $\delta_i \in \mathbb{R}^6$ is a visual odometry estimate [13]. All such estimated poses of the map are kept fixed thereafter. Note, that the inevitable drift by accumulating motion is irrelevant in our case since motion is very accurate locally and the entire map requires no global reference. For the sake of simplicity we assume loop free trajectories. Loopy trajectories, however, could be handled after loop closure detection [20] and pose graph optimization [19] which is beyond the scope of this article.

Next, we associate salient image points across all images of the mapping sequence. We refer to a set of pixel positions belonging to a single point in 3D as a tracklet. Every landmark that is finally stored in the map is computed from exactly one tracklet. It remains to show how to compute the 3D position of one landmark l_j from its tracklet. Recall that for mapping a stereo setup is used. Thus, the pixel position and disparity of l_j is available when observed from a set of poses $p_k,\ldots,p_{k+K}$. We summarize pixel positions and disparities in the measurement vectors $z_k = (u_k, v_k, d_k)^T, \ldots, z_{k+K} = (u_{k+K}, v_{k+K}, d_{k+K})^T$ for the landmark of interest. Finally, the error function

$$E_{\mathrm{lm}}(l_j) \quad = \quad \sum_{\kappa=k}^{k+K} ||\pi(l_j, p_\kappa) - z_\kappa||^2 \tag{6}$$

provides a goodness of fit of l_j with respect to the measured pixel positions and the poses $p_k,\ldots,p_{k+K}$ that are fixed. The function $\pi(l,p)$ computes pixel position and disparity for a landmark l observed from pose p [15]. Hence, the deviation of the expected pixel position from the measured pixel position is penalized. A good fit of the landmark l_j results in a low squared back projection error (6). In fact, we seek the landmark position $\hat{l}_j$ that yields the lowest possible error given the poses and the pixel observations. The 3D landmark position can be estimated by

$$\hat{l}_j \quad = \quad \arg\min_{l_j} \{E_{\mathrm{lm}}(l_j)\} \tag{7}$$

and is found by nonlinear least squares (NLS) estimate using the Gauss-Newton method [32]. To this end, (6) is linearized around an initial guess of l_j and its derivative is equated with zero and finally solved for l_j. The process of re-linearization and solving the resulting linear system is repeated until convergence as there exist no closed form analytical solution

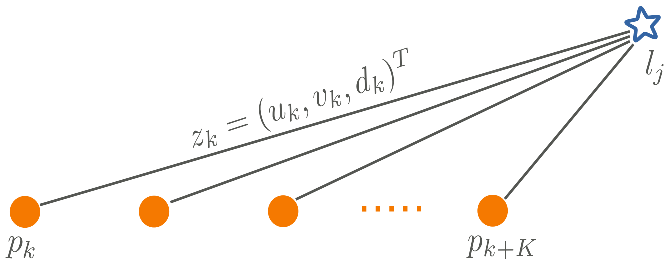

Fig. 3. The factor graph of the NLS problem (6) is shown. Landmark l_j is is observed from the depicted poses. For each pose there exists one pixel and disparity for that landmark. The NLS solver estimates the landmark position to minimize the squared back projection error. Computing an error term $\pi(l_j, p_\kappa) - z_\kappa$ for each edge and summing its squared norm corresponds exactly to (6). Hence, the shown factor graph serves as an alternative (but equivalent) representation of (6) and shall ease the understanding of the involved NLS problem.

of (7). This estimation is repeated for every tracklet.

NLS estimation is used extensively during localization and will play an important role in Section V. To ease the understanding of the functions whose minimizing argument is sought we represent NLS problems by factor graphs [18]. Figure 3 shows a graph consisting of nodes and edges connecting pairs of nodes. Every node of a factor graph corresponds to one variable of the NLS problem (e.g. poses, landmarks etc.). Edges connecting nodes correspond to constraints between the connecting variables and oftentimes coincide with measurements.

Figure 3 shows the poses $p_k, \ldots, p_{k+K}$ and the landmark l_j. Poses are solid which denote a fixed variables (ones that are not optimized). The landmark is shown by a hollow symbol indicating a variable which is optimized and is exactly the argument of (6). The summation of (6) extends over all edges of the graph each of which corresponds to a pixel position and disparity. In this particular case a single edge of the graph corresponds to one constrain $\pi(l_j, p_\kappa) - z_\kappa$ with z_κ being the respective pixel position and disparity. The connecting nodes of an edge are the only variables of this term.

At this point we cannot resist mentioning the general graph optimization software library g2o [19] which we use extensively for solving equations like (6). Using (7) for every tracklet/landmark yields the 3D position of every landmark. Finally, we prune some landmarks from the map that seem inappropriate for localization. Back projection errors and lengths of the tracklets are heuristically thresholded for this purpose. For robust and reliable landmark association during online localization landmark feature vectors are computed. In our case we use our illumination robust yet efficient novel DIRD [20] descriptor. The DIRD descriptor computes Haar features for four different scales for every pixel position of the image. Each Haar vector of every pixel position is thereafter normalized to unit L2 length. We have experimentally found that this intermediate normalization step largely contributes to the illumination robustness. Normalized Haar features are thereafter pooled over a predefined sparse set of neighborhood pixel positions by summation. Then, nine such pooled vectors are concatenated and finally each vector element is quantized to a byte value. Details can be found in [20]. For the example landmark l_j (see Figure 3) one descriptor is computed for every pose $p_k, \ldots, p_{k+K}$ it is observed from.

Next, we address the open issue of computing holistic feature vectors for every pose of the map. We largely follow our previous work on loop closure detection [20]. The input image is down sampled and partitioned into 4×4 equally sized tiles each of which is 48×48 pixels in size. Then, one DIRD descriptor is computed for the center part of each tile. All sixteen DIRD features of one image are concatenated to form the holistic feature vector. The final holistic vectors are of dimension 3456 where each element of the vector is single byte (8 bits).

For quick online retrieval we store all landmarks visible from a given pose p_i and their feature vectors extracted from that particular image i together in one file. Hence, every landmark is represented by a multitude of landmark feature vectors; one for each image the landmark was observed from. This wastefully appearing over parameterization, however, contributes much to a reliable association during online operation. It frees us from any struggle related to scale and/or rotation invariance. We simply match landmark features of the nearest pose. Moreover, the search for potentially matchable landmarks is easy. Only landmarks that are stored for the currently nearest mapping pose are used. The map data structure is depicted in Figure 2.

V. Localization

Next we present the localization algorithm. A single monocular camera is used and we show how to localize that camera relative to the visual map as described in Section IV. Firstly a rough overview is presented. At that point we spare the details before elaborating the technicalities in Sections V-A and V-B respectively. Figure 4 shows an overview of the algorithm.

Our localization algorithm follows a two step approach. At first the method identifies the pose of the map that is closest to the current ego position. We use a visual description of the image and query this “visual signature” against the map database. The result is the nearest pose of the map. We refer to this step as topological localization as it performs a search in the graph of map poses. Details of how to achieve great robustness in situations of visual ambiguity are elucidated detailedly in Section V-A.

Knowledge about the nearest pose allows to subsequently load 3D landmarks of the vicinity of the area the vehicle is currently in. These 3D landmarks are associated with pixel positions of the current camera image. Landmark to pixel associations are finally harnessed to derive a high precision metric ego pose estimate with six degrees of freedom. Details of this step are presented out in Section V-B. An overview sketch of the approach is illustrated in Figure 4.

A. Topological Localization

The goal of topological localization is to find the pose of the map that is nearest to the current ego position. Thereto

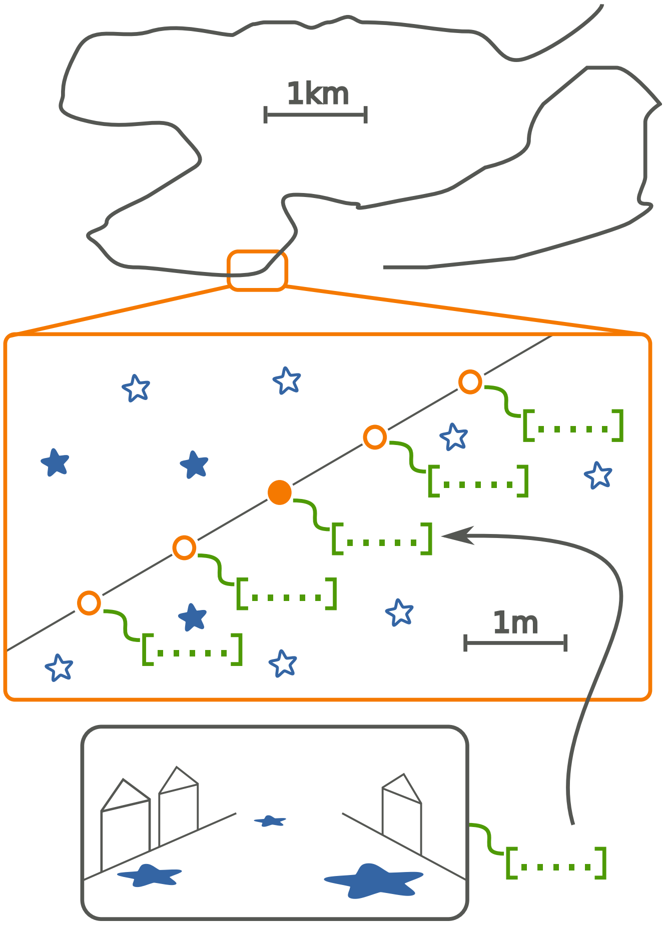

Fig. 4. An overview of the localization algorithm is shown. The map consists of poses (orange circle), associated holistic features (green brackets with dots) and landmarks (blue stars). The current camera image is depicted on the bottom. A holistic feature is extracted from it and the most similar vector of the map is found (arrow). All landmarks of the vicinity are then associated with salient points of the image. The three solid landmarks (stars) are associated successfully in the example and shown in the camera image. The landmark associations are translated into a high precision metric ego pose estimate.

a holistic feature vector is extracted from the current camera image akin to the approach presented in Section IV. We again tile the image and describe each tile by one DIRD [20] feature. Concatenating single tile features into one yields the holistic vector. Let this feature vector be denoted by f_i with i being the current time index. Furthermore, let $g_1, \dots, g_N$ be all holistic features of the map. A column vector D_i of L1 distances is then computed by

$$D_i \quad = \quad \left(||g_1 - f_i||_1, \dots, ||g_N - f_i||_1\right)^T . \tag{8}$$

Note that this operation can be performed quite efficiently on modern CPUs using SIMD instructions since DIRD features are byte vectors.

Simply taking the minimizing argument of (8) as the result of the nearest neighbor search is error prone and susceptible to visual aliasing and ambiguities. Hence we introduce a post processing step next to refine the search. The idea is to expect that some recently preceding holistic feature $f_{i'}$ matches $g_{k'}$ if the current f_i matches g_k and mapping pose $p_{k'}$ is in close proximity to pose p_k. Therefore, we match subsequences of features.

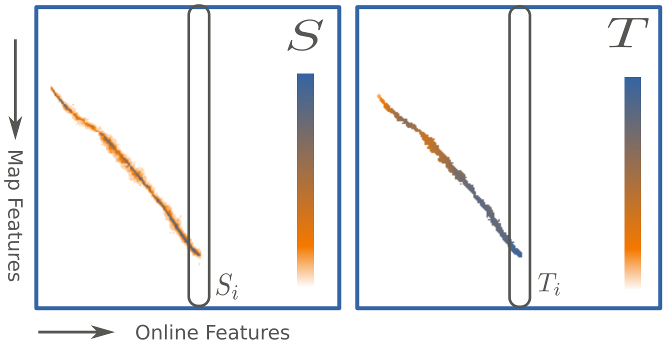

Fig. 5. A similarity matrix S is shown on the left. For each entry j, i it shows the similarity between pose j of the map and the current camera image at time i. One can observe a pronounced streak which strongly hints at a well matched subsequence of past poses. This is exploited and a noise removal procedure yields the matrix T depicted on the right. See text for details.

We formalize this requirement by first defining the similarity vector

$$S_i \quad = \quad \left(\mathrm{logit}(||g_1 - f_i||_1), \dots, \mathrm{logit}(||g_N - f_i||_1)\right)^T \tag{9}$$

with $\mathrm{logit}(\cdot)$ being a logistic function which translates all distances of (8) into a similarity scores in the range $(0, 1)$. Next the similarity matrix

$$S \quad = \quad (S_1, \dots, S_i) \tag{10}$$

with column vectors of (9) is defined. Thus, $S_{j,i} \in S$ denotes great similarity of the current image at time i to the map pose j if its value approaches one. A such defined similarity matrix is best shown visually and an example is depicted in Figure 5 (left). An off-diagonal streak of high similarities can clearly be seen. We aim at finding such streaks which hint at well matched subsequences.

For any pose j of the map that we consider as nearest pose candidate we search for a streak as in Figure 5 that ends at row j in column i which is the current camera index. Formally let

$$T_{j,i} \quad = \quad \max_{\substack{j_0,\dots,j_L \text{ s.t.} \\ (j_{k-1}-j_k)\in\{0,1,2,3\} \\ \text{and } j_L=j}} \sum_{k=0}^{L} S_{j_k, i-k} \tag{11}$$

be the maximum sum of one such streak of length $L + 1$. The matrix T with elements $T_{j,i}$ can be computed efficiently from S by dynamic programming and one example is shown in Figure 5. Obviously, only the most current column T_i of T needs to be computed at each time step. Moreover, we compute $T_{j,i}$ only for those j, i which seem promising. We choose the M best scores of S_i to compute $T_{j,i}$. The streak length in our experiments is $L = 30$ and we choose the $M = 10$ best candidates at each time step. If the maximum value of the matrix column T_i exceeds a threshold τ we output its index as the nearest pose of the map. In our experiments we set $\tau = 0.3L$. Updating the initial similarity matrix S takes time linear in the number of mapping poses whereas the refinement of computing T is constant for any map size.

Note that computing visual similarity for topological localization as presented above is concise and globally optimal. We see

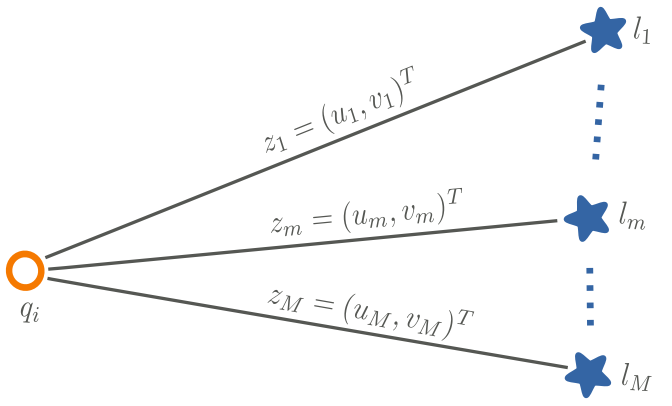

Fig. 6. The factor graph of the NLS problem of (12) is shown. The current pose (orange circle) is optimized for fixed landmark positions (solid blue stars) such that it matches the measured pixel positions z_m best.

some advantages over filter based alternatives like particle or histogram filters. Unsolved problems like the correct number of particles or issues related to particle depletion do not arise. Moreover, the dynamic programming is very efficient and takes only a few milliseconds on a single core for moderate map size (a few ten thousand features).

B. Metric Localization

Our metric localization method follows a two step approach. First an initial (metric) estimate of the camera pose is computed from landmark associations. Henceforth, we will refer to this as a *one shot estimate*. Since these one shot estimates can have varying accuracy and may even fail in some very unfavorable situations a windowed history of one shot estimates is stored. During the second step, these past one shot estimates are re-optimized jointly and we refer to this step as *pose adjustment*. To this end we fit a motion model to the sliding window of past one shot estimates. Both steps are elaborated in greater depth next.

During metric localization it is assumed that the pose of the map that is closest to the current camera position is already known. Either it is easily inferred from the immediately preceding time step or the topological localization provides a hint. This knowledge allows to load the associated map pose file which contains all nearby landmark positions and their visual descriptors from disk (see also Figure 2). Next, these landmarks are associated with pixel positions of the current camera image. Salient points are extracted, described by DIRD and matched with those of the map. The search space within the image plane can be restricted quite heavily since a good ego pose estimate is known from the previous time step already. Let the set of landmarks successfully matched be $l_1, \ldots, l_M$ and their associated pixel positions be $z_1 = (u_1, v_1)^T, \ldots, z_M = (u_M, v_M)^T$. The current ego pose is denoted by $q_i \in SE(3)$.

The 3D landmark positions are kept fixed and a one shot estimate $\bar{q}_i$ is found by seeking the minimizing argument of

$$E_{\text{one}}(q_i) \quad = \quad \sum_{m=1}^{M} ||\pi(l_m, q_i) - z_m||^2 \qquad (12)$$

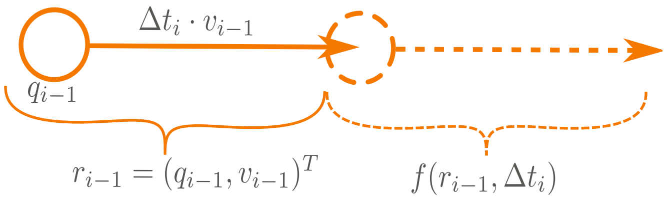

Fig. 7. The prediction of a pose from a previous *velocity augmented pose* is shown. The pose contains a velocity vector $v = (v_x, v_y, v_z)^T$ which can be used to extrapolate with the known time lage Δt.

where $\pi(l, q)$ computes the pixel position of the 3D point l projected into the camera at pose q [15]. This one shot estimate is found by NLS estimates and denoted by

$$\bar{q}_i \quad = \quad \arg\min_{q_i} \{E_{\text{one}}(q_i)\}\,. \qquad (13)$$

The factor graph associated with the NLS problem (12) is visualized in Figure 6. Landmarks are denoted by stars and since they are kept fixed (are not optimized for) are depicted with solid colors. The pose q_i is denoted by a hollow circle and is the argument of (12). The summation of (12) extends over the edges of the graph which are labeled with the measured pixel positions z_m.

Since (12) is a quadratic error function it is naturally very susceptible to any outliers. Outliers can arise from miss-associations which cannot be fully avoided in practice despite a carefully designed feature matcher. Moreover, any incorrectly estimated landmark can cause such outliers as well. Undetected outliers can cause catastrophic divergences of the pose estimator. Therefore, we wrap the estimate of (13) in a random sampling consensus [12] (RANSAC) algorithm. We randomly draw minimum sets of three landmarks, estimate $\bar{q}_i$ and evaluate all landmark pixel positions for support of the current hypothesis. After one hundred such iterations the largest inlier set is optimized jointly in a NLS sense to yield the final one shot estimate $\bar{q}_i$.

Measurement covariance matrices have been neglected so far in favor of better readability. The norm of (12) is in fact a squared Mahalanobis norm which considers measurement uncertainty. Next, we introduce the pose adjustment step of the localization algorithm which jointly re-optimizes a set of past one shot estimates. This step, however, requires a certainty measure of the estimate $\bar{q}_i$ and we denote its covariance matrix by Σ_i. We find a judicious choice of Σ_i by checking the number of inlier landmark associations (according to RANSAC). Uncertainty is increased for fewer inlier landmark matches and vice versa.

A windowed history of past one shot estimates $\bar{q}_{i-K}, \ldots, \bar{q}_i$ are now to be re-optimized jointly to yield the final ego pose estimate. Due to the absence of any additional external hardware like odometers or the like we resort to forcing the motion induced by these past one shot estimates to follow certain dynamics. At this point we exploit the knowledge that the camera is mounted inside a vehicle which naturally follows non-holonomic motion models.

Thus, we augment each pose of the window by velocities in each dimension and set $r_i = (q_i, v_i)$ with velocity vector

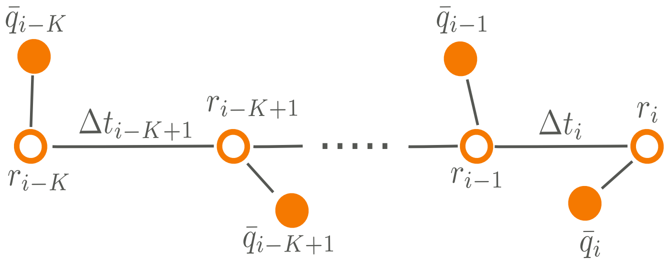

Fig. 8. The factor graph associated with (16) is shown. The one shot estimates $\bar{q}_i$ serve as a prior during pose adjustment. The priors are balanced with their predictions from the velocity augmented poses.

$v_i = (v_i^x, v_i^y, v_i^z)^T$. We refer to r_i as *velocity augmented pose*. Moreover, the time lag Δt_i between any two poses q_{i-1} and q_i is known. This allows to compute a prediction $\bar{r}_i$ of the velocity augmented pose r_i by applying a small perturbation $\Delta t_i \cdot v_{i-1}$ to pose q_{i-1}. A prediction of the velocity augmented pose r_i is thereby obtained from the prediction function

$$\begin{aligned}\bar{r}_i &= f(r_{i-1}, \Delta t_i) \\ &= \left(q_{i-i} \oplus \left(\Delta t_i \begin{pmatrix} v_{i-1} \\ \mathbf{0}_{3\times 1}\end{pmatrix}\right), v_{i-1}\right) \end{aligned} \tag{14}$$

assuming constant velocity. The prediction function is shown in Figure 7.

The pose adjustment then tries to balance the prediction $f(r_{i-1}, \Delta t_i)$ with its one shot estimate $\bar{q}_i$ (which serves as a prior) while penalizing velocity changes. To this end we define the subtraction of the velocity augmented poses

$$\begin{aligned} r' \ominus r &= (q', v') \ominus (q, v) \\ &= \left((q' \ominus q)^T, (v' - v)^T\right)^T \in \mathbb{R}^9 \end{aligned} \tag{15}$$

and derive the error function

$$\begin{aligned} E_{\text{adj}}(r_{i-K}, \ldots, r_i) &= \sum_{k=0}^{K} ||q_{i-k} \ominus \bar{q}_{i-k}||^2_{\Sigma_{i-k}} + \\ & \sum_{k=0}^{K-1} ||f(r_{i-k-1}, \Delta t_{i-k}) \ominus r_{i-k}||^2_{\Gamma} \end{aligned} \tag{16}$$

with the weight matrix Γ that balances angel, position and velocity differences accordingly. The minimizing argument of (16)

$$\hat{r}_{i-K}, \ldots, \hat{r}_i = \underset{r_{i-K}, \ldots, r_i}{\arg\min} \{E_{\text{adj}}(r_{i-K}, \ldots, r_i)\} \tag{17}$$

is taken as the final metric pose estimate.

The factor graph of (16) is depicted in Figure 8. The velocity augmented poses $r_{i-K}, \ldots, r_i$ which are subject to optimization are shown. These are the arguments of the error function (16). These are inter connected and each edge corresponds to one constraint. Edges connecting consecutive poses are constraints stemming from the motion model (prediction function (14)), are labeled with the time lag Δt and correspond exactly to the second summation of (16). Edges that connect to a one shot estimate $\bar{q}_i$ penalize any deviation of r_i to $\bar{q}_i$ and these edges represent the first summation of (16).

A joint re-optimization of a set of previous one shot estimates increases the accuracy of the estimate. Additional constraints (motion model) provide additional cues which can only be exploited in joint optimization. Furthermore, the squared Mahalanobis norm of the residual $||\hat{q}_i \ominus \hat{q}_i||_{\Sigma_i} = (\hat{q}_i \ominus \bar{q}_i)^T \Sigma_i^{-1} (\hat{q}_i \ominus \bar{q}_i)$ is interpreted to disclose any one shot estimate $\bar{q}_i$ that is an outlier in the pose adjustment sense. If, for any reason one such one shot estimate has yielded an unreasonable value it is found hereby, pruned from (16) and the final estimate (17) is re-computed. This one shot outlier detection further contributes to the overall robustness of the method.

Finally, we justify our choice of the constant velocity model to constrain the camera motion. Many more elaborate motion models like the curve linear models of [34] or dynamic single track models may appear deceptively tempting. However, these models require the knowledge of the mounting position of the camera relative to the vehicle center. The aforementioned models require a notion of the heading of the vehicle; not the heading of the camera which can be arbitrary in our case. Our goal was to allow this localizer to work without any troublesome camera to vehicle calibration. Nothing keeps one from using this approach with a sidewards facing camera even though we admit to have tested it only with forward and backward facing configurations.

Figure 9 shows a flow chart of the metric localization and compares it to the traditional satellite based navigation approach. The GNSS solution derives the ego position by reasoning about pseudo range measurements and finally smoothing the result by integrating IMU readings in a filter framework whereas our method minimizes landmark observation errors in a NLS framework and finally reoptimizes these one shot estimates jointly to integrate a motion model.

VI. Experiments

Next we present experiments on real world data to assess and evaluate our method. First we describe the results of our mapping tool chain. Thereafter, localization experiments are shown. Finally, we present results of our AR system where manually labeled objects of the mapping trajectory are projected into the camera image during online localization.

We have equipped a standard station wagon with two stereo camera setups. One stereo rig is facing forward whereas the other is facing backwards. Imagery is recorded and analyzed thereafter. Note that forward and backward facing cameras are never used jointly but are always evaluated independently. Hence we obtain one set of recordings for each stereo setup. Stereoscopy is required and used only for the creation of the map. A mere single monocular camera is used in all localization experiments. No additional sensors like GPS are used anywhere in the experiments. We note that the forward facing camera setup has a slightly narrower field of view which seems to impact some of the experiments (see Section VI-B). We have picked a 7km route through mostly urban and partially rural areas as representative testing ground. We have traveled this route on three different days each two weeks apart. The first traversal was used to create the visual map and we will refer to this test set as *MAP*. The two remaining

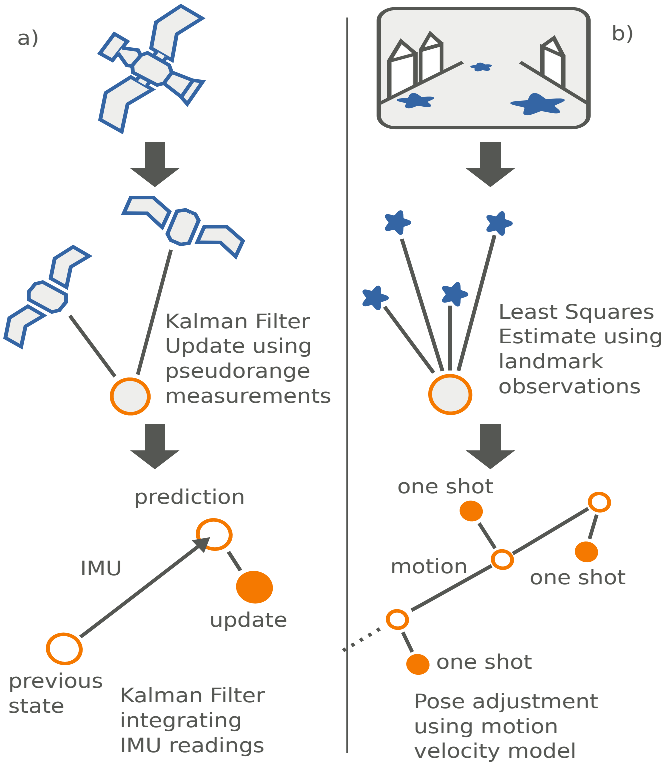
a)
Kalman Filter Update using pseudorange measurements
prediction
IMU
update
previous state
Kalman Filter integrating IMU readings
b)
Least Squares Estimate using landmark observations
one shot
motion
one shot
one shot
Pose adjustment using motion velocity model

Fig. 9. A flow chart comparing the a loosely coupled GPS/IMU solution (left side, (a)) to the proposed metric localization (right side, (b)). Both follow a two step approach. The satellite approach fuses pseudo range measurements in a Kalman filter framework and thereafter fuses IMU readings in another filter. Our approach relies on NLS and computes a one shot estimate from landmark observations. One shot estimates are finally reoptimized jointly integrating a motion model.

recordings are used for the localization experiments we refer to them as *LOC1* and *LOC2* respectively.

A. Mapping Experiments

A visual map was created for both backward and forward facing cases of the mapping test set *MAP*. Map computation is not time critical for real time operation and takes roughly three hours for each set. Almost twenty million landmark candidates are created (for each case) of which approximately 30% are rejected and pruned from the map. The storage size for these map sizes is roughly 5 gigabytes each in a compact binary format. It includes all poses, landmarks and their visual descriptors.

B. Localization Experiments

In the following we will present several localization experiments which are performed on each of the test sets. In particular we use the mapping trajectory for the localization experiments as well. The localization experiment for the mapping case obviously yields excellent results and shall serve as an upper (or lower) bound which cannot be exceeded by any other test set. Thus, we obtain six results for each test which are *MAP*, *LOC1* and *LOC2* for the forward and backward facing camera configuration each.

At first we determine the traveled distance before a topological

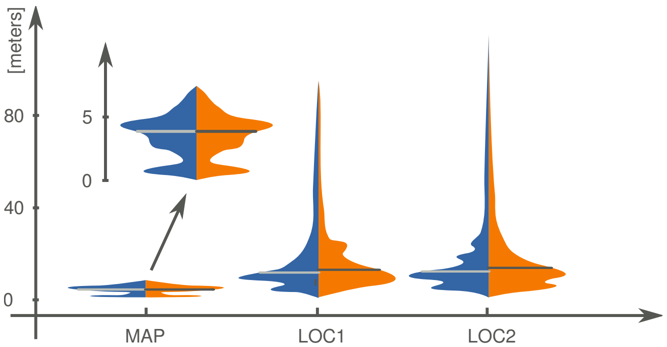

Fig. 10. Two sided violin plots (vertical, smoothed histograms) for the required traveling distance until a topological localization is possible are shown for three test sets. The first plot shows results for the mapping sequence (shown again with more appropriate scale) whereas the others show findings for the two localization test sequences. The left half of each plot shows results for the backward facing configuration. The right portion represents the forward facing camera setup. The median is shown for each plot.

localization is possible, hence until the nearest pose of the map is found. We have replayed each test set from 500 equidistantly placed starting positions and determined the distance until a topological localization is achieved.

We present results as two sided violin plots in Figure 10. One plot represents one test set each. The plots show vertical and smoothed histograms of the distance until topological localization is possible. The left halves (blue) of each plot represents the results for the backward facing case whereas the right portion (orange) shows findings for the forward facing setup. The median is marked as well.

Since topological localization works perfectly for the mapping trajectory its plot is re-sketched with a more appropriate scale in Figure 10. We have removed the upper 5% quantile from the plots for better visibility. The median for topological localization is 3.1 meters for the mapping test set (both forward and backward), 8.0 meters for *LOC1/backward*, 9.1 meters for *LOC1/forward*, 7.8 meters for *LOC2/backward* and 9.6 meters for *LOC2/forward*. However, we note that some areas (especially rural ones) are unfavorable for topological localization and may easily require one hundred meters and more of traveling before topological localization is possible. Furthermore, our experiments indicate a high sensitivity to lane differences between mapping and localization. This can be seen from the spiking tops of the plots in Figure 10. All topological localizations that the localizer has output are corrected and are verified by the subsequent metric localization.

We follow the same testing procedure as before to asses the number of inlier landmark associations for each time step during metric localization. To this end, each test set is used for metric localization and the number of inlier landmark associations (according to RANSAC) are tracked. Results are again shown by two sided violin plots for the six cases in Figure 11. Point matching works perfectly for the mapping trajectory since matching images are identical. Hence, the left plot of Figure 11 corresponds to the histogram of number of visible landmarks during mapping (every single landmark is associated correctly). Matching images from a different day (*LOC1* and *LOC2*) is more realistic. The number of

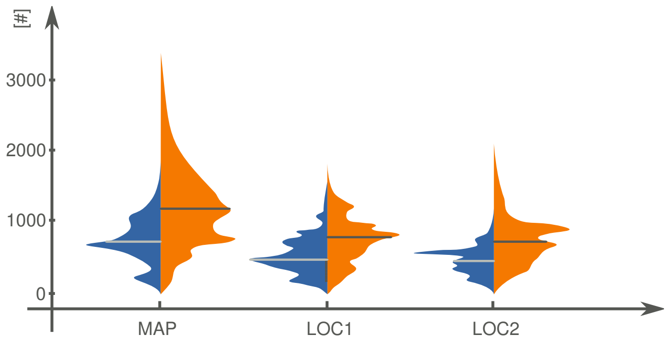

Fig. 11. Violin plots for the number of successfully associated landmarks per camera image during localization are shown for the three test sets. See also Figure 10.

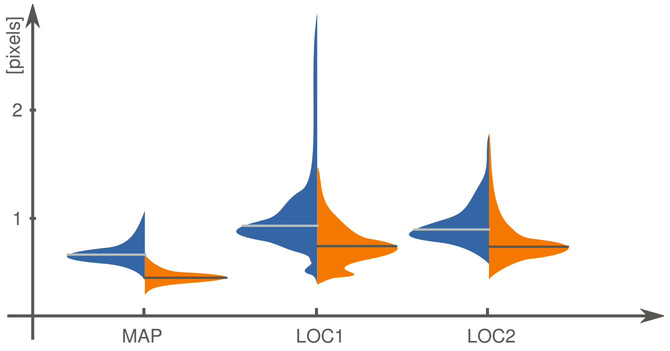

Fig. 12. Violin plots for the mean back projection error of all inlier landmark associations (in pixels) are shown for the three test sets. See also Figure 10.

correctly found landmarks ranges from zero (underexposed camera in underbridge) to almost two thousand in some case. A significantly higher number of landmarks are associated for the forward facing case. We attribute this to the narrower field of view which makes camera calibration and feature matching easier.

Next, we illustrate the mean back projection error of the inlier landmark matches during one shot estimation (cf. (12)) in pixels in Figure 12. Significant differences can be observed between forward and backward facing setups. We again attribute this to the narrower field of view. The median back projection error is between 0.5 and 1.0 pixels; a range we would have expected for good localization. The one shot estimator can be well initialized using a prediction from the preceding pose adjustment step since velocities are known. The good state initialization and the low state dimensionality allows the estimator to converge within a few ten iterations and we can therefore afford a tight termination threshold.

So far only the left camera image of the stereo recordings has been used. Since we have recorded both left and right images of the stereo setup in all cases we are now able to estimate the trajectory for both the left and right camera. We compute the one shot estimates for every left and right image independently and compare them for consistency. Since the base length of the stereo rig is known the right camera estimate can be compensated for it and subtracted from the left camera estimate. The norm of the difference between the two estimates are depicted in Figure 13. The left and right one shot estimates clearly agree to within centimeter level accuracy. The experiment is repeated for the final localization after pose adjustment and the results are visualized in Figure 14. We believe that the consistency measure presented here is of the same magnitude as the localization accuracy is. We note that the wider field of view of the backward facing camera overrules any effects of poorer point feature associations and yields a better localization accuracy nevertheless.

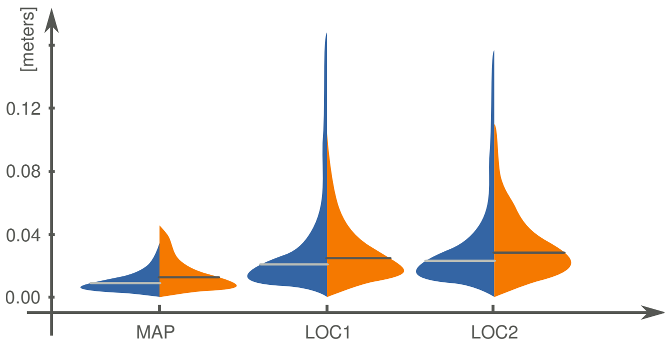

Fig. 13. Violin plots of the difference of two independent localization estimates of two independent cameras mounted on the same vehicle are shown. They indicate the magnitude of the localization error relative to the visual map. Findings for the one shot estimate are shown.

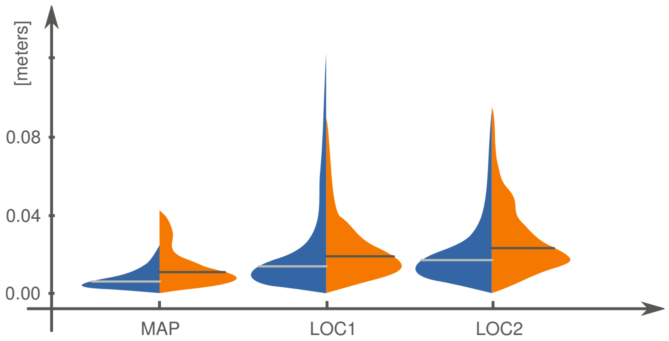

Fig. 14. Violin plots of the difference of two independent localization estimates of two independent cameras mounted on the same vehicle are shown. They indicate the magnitude of the localization error relative to the visual map. Findings for the finale ego pose estimate (pose adjustment) are shown.

C. Augmented Reality Experiments

Quantitative experimental findings were highlighted in the previous sections. Next, we present results of some qualitative tests. To this end, we have labeled static objects of interest like pedestrian crossings, road signs etc. in images of the mapping sequence. Since we have stereo data recorded for all frames of the mapping survey we are able to reconstruct these objects in 3D using stereo vision. The 3D reconstructions are then stored in the map data structure. In the sequel we will refer to these objects as map objects.

Once the vehicle approaches one such map object during online localization, the 3D positions of these objects are loaded from disk. As the 3D position of the ego vehicle is precisely known relative to the map from the presented localization method, the ego position is also known relative to the map objects with high accuracy. This in turn allows to overlay the

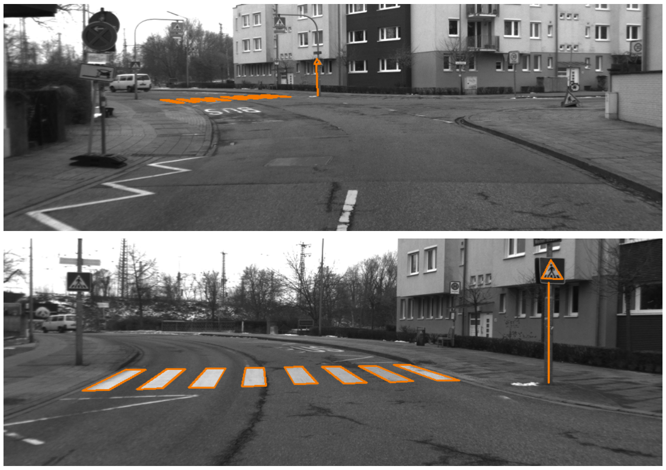

Fig. 15. Infrastructural objects (pedestrian crossing) are overlaid onto the camera image after localization yielding an AR system. The same crossing of Figure 1 is shown for another trajectory.

map objects onto the camera image yielding an AR system. Any imprecise ego localization results in an overlay of the map object that does not align with its image content. One example is shown in Figure 1 of the introduction. The same area is shown for the other localization test set in Figure 15. The examples show a good fit of the project objects with the camera image. A false localization would be seen as a significant deviation of the objects projection from the actual image position.

Lastly we have tested the performance of DIRD for illumination robust point matching. We have picked two stereo images which are captured on different times of the day with harsh cast shadows coming from different direction. The images are shown on the top of Figure 16. Then we detected salient points and matched them using DIRD to find correspondences. We display only the point correspondences which comply with the robustly estimated motion between the two frames. The result is shown in Figure 16 third from top with 1391 points matched correctly. We repeated the same experiment using the (extended) upright SURF as the descriptor. We have used the exact same detector as before so that the result is solely dependent on the descriptor choice. The result is depicted on the bottom of Figure 16 showing only 550 correctly matched points. DIRD more than doubles the point matching performance of USURF.

VII. Conclusion

We have presented a system for six DOF real time ego localization using only a single monocular camera. The camera is localized relative to a previously computed visual map which is created automatically from stereoscopy. During online localization a holistic feature vector is extracted from the current camera image and compared to a all vectors of the map. A dynamic programming procedure ensures to find the pose of the map that is closest to the current ego position with great robustness. A map relative metric localization starts from there by matching image points to landmarks and deriving the ego pose estimate. Finally, a motion model constraint is applied to a windowed history of one shot ego pose estimates to further stabilize the process. Jointly reestimating previous one shot estimates allows to exploit the nonholonomic motion inherent to car like vehicles.

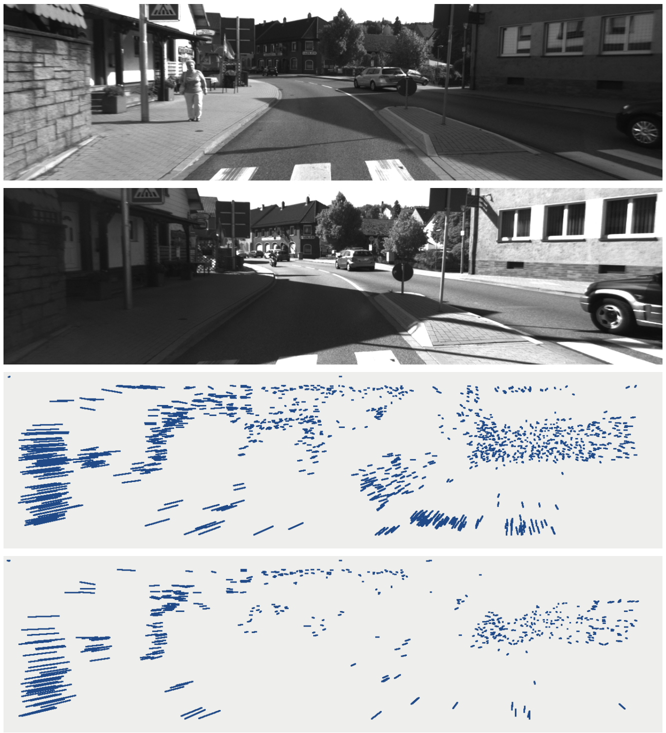

Fig. 16. Two images showing the same place but captured under severely varying lighting conditions are shown on the top. Salient points are detected and described by DIRD [20]. Finally point correspondences are found and those that comply with the motion between the two frames are counted and shown (third from top). The point matching experiment is repeated with the exact same detector but points are described by USURF. The resulting point correspondences are shown on the bottom. DIRD more than doubles the performance of USURF.

In extensive experiments a centimeter level accuracy was demonstrated. The achieved precision enables an AR system which displays relevant infrastructural objects like pedestrian crossings and the like. These infrastructural objects have been manually labeled and stored in the map data structure.

Computing the aforementioned objects fully automatically seems an exiting and obvious next step which we plan to pursue. Computing such objects during mapping has three advantages over the alternative of computing them online. First, the need for hard real time constraints on weak automotive hardware is completely obviated. Second, a far sensing range is irrelevant in this case. Such objects can be detected once they are very close to the camera yielding a much greater robustness. Finally, the detection result can be manually verified if necessary.

While the topological localization has proved robust for a fixed camera mounting, future work will include an investigation of bag of feature approaches to allow for variable camera orientation. This is especially tempting, since point features of the camera image need to be computed for metric localization already and are hence obtained without any additional computational expenses.

References

[1] P. Agarwal and E. Olson. Variable reordering strategies for slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[2] A. Bachmann and I. Lulcheva. Combining low-level segmentation with relational classification. In *Computer Vision Workshops (ICCV Workshops)*, pages 1216–1221, 2009.

[3] H. Badino, D. Huber, and T. Kanade. Visual topometric localization. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 794–799, 2011.

[4] H. Badino, D. Huber, and T. Kanade. Real-time topometric localization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1635–1642, 2012.

[5] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *Robotics & Automation Magazine, IEEE*, 13(3), 2006.

[6] M. Bosse, P. Newman, J. Leonard, and S. Teller. Simultaneous localization and map building in large-scale cyclic environments using the atlas framework. *The International Journal of Robotics Research*, 23(12):1113–1139, 2004.

[7] J. Courbon, Y. Mezouar, and P. Martinet. Autonomous navigation of vehicles from a visual memory using a generic camera model. *IEEE Transactions Intelligent Transportation Systems*, 10(3):392–402, 2009.

[8] M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[9] F. Dellaert, J. Carlson, V. Ila, K. Ni, and C.E. Thorpe. Subgraph-preconditioned conjugate gradients for large scale slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2566–2571, 2010.

[10] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *Robotics & Automation Magazine*, 13(2):99–110, 2006.

[11] H. Fang, C. Wang, M. Yang, and R. Yang. Ground-texture-based localization for intelligent vehicles. *IEEE Transactions Intelligent Transportation Systems*, 10(3):463–468, 2009.

[12] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[13] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968, 2011.

[14] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard. A tutorial on graph-based slam. *Intelligent Transportation Systems Magazine, IEEE*, 2(4):31–43, 2010.

[15] R. Hartley and A. Zisserman. *Multiple view geometry*, volume 642. 2000.

[16] C. Hertzberg. A framework for sparse, non-linear least squares problems on manifolds. In *UNIVERSITÄT BREMEN*, 2008.

[17] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebl, F. von Hundelshausen, et al. Team annieway's autonomous system for the 2007 darpa urban challenge. *Journal of Field Robotics*, 25(9):615–639, 2008.

[18] 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.

[19] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3607–3613, 2011.

[20] 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 (IV)*, Gold Coast, Australia, 2013.

[21] H. Lategahn, A. Geiger, and B. Kitt. Visual slam for autonomous ground vehicles. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1732–1737, 2011.

[22] H. Lategahn, H. Schreiber, J. Ziegler, and C. Stiller. Urban localization with camera and inertial measurement unit. In *IEEE Intelligent Vehicles Symposium (IV)*, Gold Coast, Australia, 2013.

[23] H. Lategahn and C. Stiller. City gps using stereo vision. In *IEEE Conference on Vehicular Electronics and Safety*, Turkey, July 2012.

[24] J. Levinson, M. Montemerlo, and S. Thrun. Map-based precision vehicle localization in urban environments. In *Robotics: Science and Systems Conference (RSS)*, 2007.

[25] J. Levinson and S. Thrun. Robust vehicle localization in urban environments using probabilistic maps. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4372–4378, 2010.

[26] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4(4):333–349, 1997.

[27] M. Milford. Visual route recognition with a handful of bits. In *Robotics: Science and Systems Conference (RSS)*, 2012.

[28] F. Moosmann and C. Stiller. Velodyne SLAM. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 393–398, June 2011.

[29] M. Obst, C. Adam, G. Wanielik, and R. Schubert. Probabilistic multipath mitigation for gnss-based vehicle localization in urban areas. In *ION GNSS Conference*, 2012.

[30] 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.

[31] O. Pink. Visual map matching and localization using a global feature map. In *Computer Vision and Pattern Recognition Workshops, IEEE Conference on*, 2008.

[32] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes 3rd edition: The art of scientific computing*. 2007.

[33] M. Schreiber, C. Knoeppel, and U. Franke. Laneloc: Lane marking based localization using highly accurate maps. In *IEEE Intelligent Vehicles Symposium (IV)*, Gold Coast, Australia, 2013.

[34] R. Schubert, E. Richter, and G. Wanielik. Comparison and evaluation of advanced motion models for vehicle tracking. In *IEEE International Conference on Information Fusion*, pages 1–6, 2008.

[35] 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.

[36] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous robot vehicles*, 1:167–193, 1990.

[37] N. Sunderhauf, M. Obst, G. Wanielik, and P. Protzel. Multipath mitigation in gnss-based localization using robust optimization. In *IEEE Intelligent Vehicles Symposium (IV)*, 2012.

[38] N. Sunderhauf and P. Protzel. Brief-gist-closing the loop by simple means. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1234–1241, 2011.

Henning Lategahn studied Computer Science at RWTH Aachen University, Germany and at the University of Jyväskylä, Finland. He received his Diploma degree with distinction in 2008. Since 2009 he is with the Institute for Measurement and Control Systems at the Karlsruhe Institute of Technology, Germany. In 2013 he obtained his Phd degree with distinction. Dr. Lategahn is co-founder of Atlatec, a high-tech startup company specializing on vision based localization.

His research interest is in computer vision in general and vision based localization and mapping in particular.

Christoph Stiller (S'93-M'95-SM'99) studied Electrical Engineering in Aachen, Germany and Trondheim, Norway, and received the Diploma degree from Aachen University of Technology in 1988. In 1988 he became a Scientific Assistant at Aachen University of Technology where he completed his Dr.-Ing. degree (Ph.D.) in 1994. He worked at INRS-Telecommunications in Montreal, Canada as a post-doctoral researcher during 1994-1995. In 1995 he joined the Corporate Research and Advanced Development of Robert Bosch GmbH, Hildesheim, Germany, where he was responsible for "Computer Vision for Automotive Applications". In 2001 he became chaired professor and director of the Institute for Measurement and Control Systems at Karlsruhe Institute of Technology, Germany. In 2010 he was appointed as Distinguished Visiting Scientist for three months at CSIRO in Brisbane, Australia.

Dr. Stiller serves as President of the IEEE Intelligent Transportation Systems Society (2012-2013) and was Vice President for Publications (2009-2010) and for Member Activities (2006-2008). He served as Editor-in-Chief of the IEEE Intelligent Transportation Systems Magazine (2009-2011) and as Associate Editor for the IEEE Transactions on Image processing (19992003), for the IEEE Transactions on Intelligent Transportation Systems (2004-ongoing) and for the IEEE Intelligent Transportation Systems Magazine (2012-ongoing).