Lategahn, Henning ; Stiller, Christoph: City GPS using Stereo Vision. In: IEEE In- ternational Conference on Vehicular Electronics and Safety (ICVES). Istanbul, Turkey, July 2012, pp. 1–6

City GPS using Stereo Vision

Henning Lategahn and Christoph Stiller
Institute of Measurement and Control
Karlsruhe Institute of Technology
Karlsruhe, Germany
{henning.lategahn, stiller}@kit.edu

***Abstract*— Next generation driver assistance systems require a precise localization. However, global navigation satellite systems (GNSS) often exhibit a paucity of accuracy due to shadowing effects in street canyon like scenarios rendering this solution insufficient for many tasks. Alternatively 3D laser scanners can be utilized to localize the vehicle within a previously recorded 3D map. These scanners however, are expensive and bulky hampering a wide spread use.**

Herein we propose to use stereo cameras to localize the ego vehicle within a previously computed visual 3D map. The proposed localization solution is low cost, precise and runs in real time. The map is computed once and kept fixed thereafter using cameras as sole sensors without GPS readings. The presented mapping algorithm is largely inspired by current state of the art simultaneous localization and mapping (SLAM) methods. Moreover, the map merely consists of a sparse set of landmark points keeping the map storage manageably low.

I. Introduction

Autonomous driving and next generation driver assistance systems highly depend on a reliable localization solution. It is largely agreed that global localization is a necessary prerequisite for autonomous transportation systems. Solving this elusive problem may pave the way to fully autonomously operating vehicles at one time. Nowadays the issue is addressed by GNSS receivers which provide a global position estimate. However, their accuracy is largely limited by inference and reflection of the satellite signal. In urban scenarios the problem becomes even more pronounced due to shadowing effects and occluded lines of sight to the satellites. Differential solutions like DGPS often coupled with high precision integrated measurement units (IMUs) are very expensive and find their application only in test vehicles. The localization problem has recently been approached by matching a 3D laser scan of the environment to a previously recorded 3D map. This scan matching approach seems promising since many of the aforementioned problems are eluded. However, common laser scanners used for this task are massive and expensive hence preventing them from being used in mass production. The method presented in [15] shows some resemblance to the scan matching methodology. An image of a camera mounted on the vehicle is matched to an aerial image of the vicinity of the ego vehicle and hence deriving a position estimate. The recent explosive growth of imaging technology and the accompanying price reduction of camera systems makes their use particularly attractive. Nevertheless, aerial imagery is not always accessible and feature matching is non trivial.

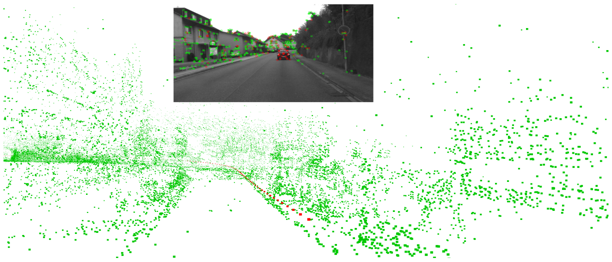

Fig. 1: An exemplar 3D map used for localization. The current image is also shown.

We herein follow the line of [15] and use a stereo camera to localize the vehicle within a map. However, the map is generated from the vehicle cameras during an offline computation. This frees us from the restriction of an available and up to date aerial map. To this end we apply current state of the art SLAM methods and solve a large scale non-linear least squares problem. The thus computed visual map consists of a sparse set of landmarks along with their visual description. During online localization this visual map is queried, landmarks are retrieved and matched to the current camera image. These matches are thereafter shaped into constraints of a non-linear least squares problem which is finally solved for the ego position. Figure 1 systematically depicts a 3D map and an estimated trajectory. The most current image of the stereo setup is also given showing landmark observations in green. The proposed method was tested in an inner city scenario. The mean localization error is 32.4 cm compared to a high precision real time kinematic (RTK) solution which we have recorded for evaluation purposes only. Moreover, we have experimental findings that urban scenarios are particularly feature rich (in terms of visual features) playing into the hands of this approach nicely.

The remainder of the manuscript is structured as follows. Related work is reviewed in Section II. The mapping part of our algorithm is presented in Section III whereas the online localization is elucidated in Section IV. An experimental evaluation is given in Section V before summarizing our contribution in Section VI.

II. Related Work

The work presented herein is related to recent work on SLAM [9], [6], [18], [10], [4], [17], [14], visual odometry [5], [8], [1] and methods that use a previously acquired map for localization [15], [2], [13], [3], [12], [11].

The problem of SLAM is to compute a map while localizing the robot within this map at the same time. Early solutions to the problem rely on extended Kalman Filters (EKFs) and variants as e.g. in [14] but the focus has more recently shifted towards bundle adjustment like approaches [17]. Much effort has been put on exploiting the inherent sparsity of the problem. Very efficient solvers have recently emerged with [9], [18] being two examples. An overview of related methods is given in [6]. Despite the great success of reducing the computational complexity there exists no constant time solution that works in a Euclidean space to date. The approach of Sibley and co-workers [17] produce maps that are Euclidean only locally. This non-constant complexity is what impedes a wide spread use of SLAM methods. The localization solution we present is constant time.

Using vision to estimate the trajectory of the ego vehicle has occurred frequently as a surrogate for wheel speed sensors and the like. Agrawal et al. use stereo vision coupled with a low cost GPS receiver in [1]. Rough outdoor scenarios have been addressed by the same group in [8]. Recently Geiger at al. have presented a real time solution specifically tailored to intelligent vehicles in [5]. Salient features of the previous stereo image are reconstructed and the ego motion is estimated such that the reconstructed points have minimal re-projection error in the current frame. This is similar to our localization algorithm except that we minimize the back projection of a previously computed map which is fixed globally. Generally, visual odometry has matured to be broadly applicable. However, the problem of long term drift cannot be overcome. Our method on the other hand does not suffer from drift.

Scan matching is frequently used to localize the ego vehicle within an already computed 3D map. [3], [12] are two examples. Levinson and Thrun have proposed to incorporate the remittance of the laser scan for improved localization in [11] improving results further. The accuracy achieved by scan matching is good. Nonetheless, the laser scanners used for these purposes are very expensive and bulky. Their price and size is only admissible for experimental vehicles. Lately, cameras have found their way into vehicles and are exploited for localization purposes. Pink [15] and Napier et al. [13] both use aerial data to refine the current ego position. A relative bundle adjustment procedure is adopted and fused with an aerial map in [13]. Pink [15] uses a point matching algorithm to match the current stereo view into a global aerial map. Badino et al. [2] use vehicle recored imagery for mapping purposes. However, their position estimate is a location within a topometric map. Thus localization corresponds to selecting one of a finite set of poses. Hence, the resulting pose estimate is rather coarse.

III. Creating the Map

We describe the mapping algorithm in the following paragraphs. The visual map consisting of landmark and poses is defined as a large state vector. Moreover, all observations of the landmarks of the map are summarized in one large observation vector. Estimating the map then corresponds to finding a state vector that best explains the measurements. First we briefly define state and observation vectors before elucidating each part in more detail. The vision front end is presented in III-A and a detailed description of the estimator is given in III-B.

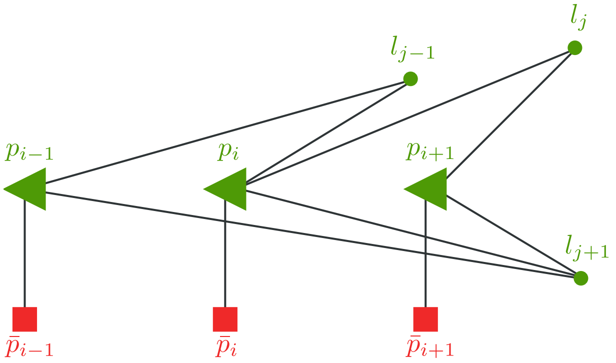

Fig. 2: Green triangles denote poses, red squares denote GPS readings and green circles denote landmarks. Each edge between any two variables corresponds to one soft constraint. Green variables are optimized whereas red ones are kept fixed.

The visual map is computed from a stereo image stream of the area to be mapped. To this end we define an auxiliary state vector $\mathbf{x}$ describing all poses $p_i = (x_i, y_i, z_i, \alpha_i, \beta_i, \gamma_i)^T$ of the stereo camera when recording the map and landmarks $l_j = (X_j, Y_j, Z_j)^T$. The pose vectors consist of three translation and three rotation parameters. Landmarks are represented by three spatial coordinates. The state vector is

$$\mathbf{x} = \begin{bmatrix} \vdots \\ p_i \\ \vdots \\ l_j \\ \vdots \end{bmatrix} \tag{1}$$

where the pose part of the vector is stored as part of the map for fast feature retrieval. More details on feature retrieval are given below. The final visual map consists of the landmark estimates l_j along with their visual description and the estimated poses p_i .

If the stereo camera observes landmark l_j from pose p_i then this measurement is defined as $z_{ij} = (u_{ij}, v_{ij}, d_{ij})^T$ with u_{ij} and v_{ij} being column and row of the image and d_{ij} disparity. Moreover we have a GPS reading for pose p_i denoted by $\bar{p}_i$.

Figure 2 shows a graph of poses, landmarks and GPS readings. An edge between any two variables defines one soft constraint. The edge connecting landmark l_j and pose p_i for instance corresponds to the soft constraint that pose and landmark shall have a constellation such that it explains the corresponding measurement z_{ij} well. An edge connecting a pose p_i and GPS reading $\bar{p}_i$ constraints the pose not to deviate much from its GPS reading, thus acts as a prior for the pose. More details are postponed until further below.

A. Vision Front End

The vision front end is outlined only very briefly and left largely untouched as it is straight forward. The detector is taken from [5] whereas the descriptor is replaced by a simple gray value surrogate.

During map building each stereo image is filtered with a blob and corner filter. Thereafter a non-maximum suppression is applied to the filter responses yielding a set of salient points. Each point is thereafter described by a 256 dimensional feature vector comprised of gray values of the vicinity of the key point. The search space for feature matching is spatially constrained and matching is efficiently performed using SIMD instruction. Outliers are heuristically removed using a voting scheme after delauny triangulation. For further details the reader is referred to [5]. Finally frame to frame feature associations are translated into tracklets assigning a unique identification number to each landmark.

B. Estimating the Map

Next, the non-linear least squares problem of creating the visual map is explained in more detail. Let $\pi(p,l) = (u,v,d)^T$ be a function that computes the pixel position and disparity of landmark l projected into the stereo setup at pose p. π is the standard stereo projection as given in e.g. [7]. It follows that each measurement of a landmark can be decomposed into the correctly projected point with additive noise

$$z_{ij} \quad = \quad \pi(p_i, l_j) + \epsilon_{ij} \tag{2}$$

with ϵ_{ij} following a zero mean normal distribution with known covariance Σ. Analogously, the GPS reading is a noisy version of the true pose

$$\bar{p}_i \quad = \quad p_i + \theta_i \tag{3}$$

where θ_i is from a zero mean Gaussian with covariance Ω_i. The total error induced by any given state $\mathbf{x}$ is

$$\begin{aligned} E(\mathbf{x}) \quad = \quad & \sum_{i,j} (\pi(p_i, l_j) - z_{ij})^T \Sigma^{-1} (\pi(p_i, l_j) - z_{ij}) \\ & + \sum_i (p_i - \bar{p}_i)^T \Omega_i^{-1} (p_i - \bar{p}_i) \end{aligned} \tag{4}$$

where the first summation extends over all possible measurements z_{ij} and the second over all poses p_i. The final state estimate is the minimizing argument of (4) and defined as

$$\hat{\mathbf{x}} \quad = \quad \arg\min_{\mathbf{x}} E(\mathbf{x}). \tag{5}$$

The non-linear least squares problem defined above can be visualized using a graph. Figure 2 shows one such graph consisting of nodes and edges. A node is a variable appearing in the minimization problem. Green denotes variables that are optimized whereas red indicates fixed variables. An edge between nodes corresponds to a soft constraint. In fact, the first summation of (4) extends over all edges that connect one pose with one landmark. The second summation extends over edges between poses and GPS readings. A much broader class of minimization problems of this sort are presented in [9].

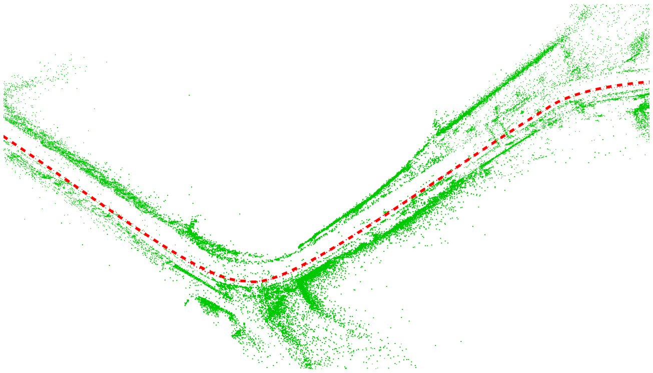

Fig. 3: A sample region of the 3D map as computed by the proposed algorithm. Landmarks are shown in green. Pose estimates are shown in red.

(5) can be approached by the Levenberg-Marquardt algorithm [16] which finds at least a local minimum of the fitness function. The Levenberg-Marquardt algorithm iteratively finds a solution to (5) by linearizing (4) around a current estimate and solving an ordinary least squares problem. In particular let the error vector be

$$e(\mathbf{x}) \quad = \quad \begin{bmatrix} \vdots \\ \pi(p_i, l_j) - z_{ij} \\ \vdots \\ p_i - \bar{p}_i \\ \vdots \end{bmatrix} \tag{6}$$

which leads to the total error (compare (4))

$$E(\mathbf{x}) \quad = \quad e(\mathbf{x})^T \Lambda e(\mathbf{x}) \tag{7}$$

with the block diagonal information matrix

$$\Lambda \quad = \quad \begin{bmatrix} \ddots & & & \\ & \Sigma^{-1} & 0 & \\ & & \ddots & \\ & 0 & \Omega_i^{-1} & \\ & & & \ddots \end{bmatrix}. \tag{8}$$

For a small deviation $\Delta\mathbf{x}$ from $\mathbf{x}$, (7) is well approximated locally by

$$\begin{aligned} E(\mathbf{x} + \Delta\mathbf{x}) \quad & = \quad e(\mathbf{x} + \Delta\mathbf{x})^T \Lambda e(\mathbf{x} + \Delta\mathbf{x}) \\ & = \quad (e(\mathbf{x}) + J(\mathbf{x})\Delta\mathbf{x})^T \Lambda (e(\mathbf{x}) + J(\mathbf{x})\Delta\mathbf{x}) \\ & = \quad e^T(\mathbf{x})\Lambda e(\mathbf{x}) + 2e^T(\mathbf{x})\Lambda J(\mathbf{x})\Delta\mathbf{x} \\ & \qquad + \Delta\mathbf{x}^T J(\mathbf{x})^T \Lambda J(\mathbf{x})\Delta\mathbf{x} \end{aligned} \tag{9}$$

whose minimizing argument can be found by equating its derivate with zero. J denotes the Jacobian of e. The minimum of (9) is found by solving

$$J(\mathbf{x})^T \Lambda J(\mathbf{x}) \Delta\mathbf{x}^* \quad = \quad -e^T(\mathbf{x}) \Lambda J(\mathbf{x}) \tag{10}$$

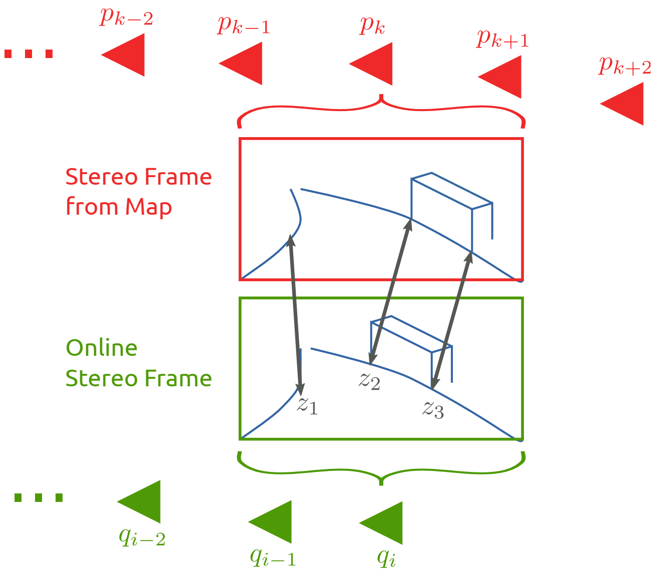

Fig. 4: Red poses are from the mapping sequence. Green ones are online poses. Features are matched between nearest neighbor poses. See text for details.

for $\Delta\mathbf{x}^*$. By introducing a damping factor λ to (10) it is easier to escape local minima. Thus the Levenberg-Marquardt algorithm iteratively solves

$$(J_i^T \Lambda J_i + \lambda I)\Delta\mathbf{x}^* \quad = \quad -e_i^T \Lambda J_i. \tag{11}$$

for $\Delta\mathbf{x}^*$ with $J_i = J(\mathbf{x}_i)$, $e_i = e(\mathbf{x}_i)$ and $\mathbf{x}_i$ being the current state at iteration i. The state estimate is finally updated by $\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta\mathbf{x}^*$. Iterations are continued until convergence is reached. We use the currently published library g2o presented in [9] for the optimization which fully exploits the inherent sparsity of the problem.

One subtlety that has not been mentioned so far is the handling of miss associated features during point tracking. This naturally occurs occasionally despite a rather sophisticated outlier detection strategy during matching. Since (4) is essentially a quadratic form it will be highly susceptible to these outliers polluting the final estimate. We tackle the problem as follows. The entire map is computed (with outliers) first. Then the back projection error $(\pi(p_i, l_j) - z_{ij})$ is examined for every landmark. Once the average back projection error of one landmark is exceeding a threshold τ it is flagged as an outlier and pruned from the map along with its associated measurements. In our experiments we chose a rather tight bound of $\tau = 2$pixels. Finally the entire map is re-estimated. After optimization the mean back projection error for both pixel position and disparity is approximately 0.5 pixel. Figure 6 shows one frame of the mapping sequence. Green circles indicate landmarks. Red circles show the outliers that are detected as mentioned before. One may observe that points on a moving vehicle are correctly classified as outliers. Outliers found on static parts of the scene have simply been too unstable or incorrectly matched causing the classifier to detect them.

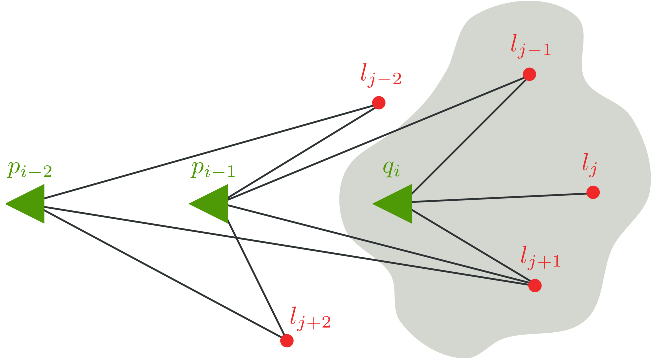

Fig. 5: Green triangles denote poses that the optimization problem is solved for. Red landmarks are kept fixed. The subgraph shown in gray is conditionally independent of the rest of the graph given landmark positions and can thus be solved independently.

IV. Localization

During online localization the visual map is queried and landmarks in the immediate vicinity of the ego vehicle are retrieved and matched into the current stereo image. GPS readings are not used during localization. The ego pose is estimated entirely from the current stereo image. First we explain the matching process. Thereafter the estimator is presented.

In the sequel the current ego position is denoted by q_i and is parameterized just like the six degrees of freedom (DOF) poses of the map (see Section III). We assume an already good estimate of the previous pose q_{i-1} and thereby a very rough idea of q_i. A pose of the map p_k that is closest to q_i is found by nearest neighbor search. The landmarks seen from p_k are readily known from the mapping process and their pixel position and feature vectors are loaded from disk. Feature points of the current stereo image are extracted and matched to those of p_k. Figure 4 illustrates this process. Figure 8 shows these matches from a test sequence.

We believe that storing a feature descriptor of one landmark multiple times for each frame of the mapping sequence makes matching robust. No invariance to scale or rotation needs to be incorporated into the feature extraction allowing substantial computational savings. The slight increase in storage is well justified by this advantage.

Next we present the localization estimator which takes matches from the current stereo image into the map as a starting point. Let $\{n_1, \ldots, n_N\} \in \mathbb{N}$ be the index set of the landmarks that are observed from q_i. For landmark l_{n_j} we obtain a pixel position and disparity in the current image as a measurement denoted by $z_j = (u_j, v_j, d_j)^T$. All measurements of time step i are stacked into a joint measurement vector

$$\mathbf{z} \quad = \quad \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix}. \tag{12}$$

Moreover, an expected measurement vector can be computed for pose q_i by

$$h(q_i) \quad = \quad \begin{bmatrix} \pi(q_i, l_{n_1}) \\ \vdots \\ \pi(q_i, l_{n_N}) \end{bmatrix} \tag{13}$$

with $\pi(q, l)$ projecting a landmark l into the frame of q (see also Section III). The current ego pose is finally chosen to be

$$\hat{q}_i \quad = \quad \arg\min_q \left\{ (h(q) - \mathbf{z})^T \, \Gamma \, (h(q) - \mathbf{z}) \right\} \tag{14}$$

with Γ being block diagonal with the inverse covariances Σ^{-1} of the pixel noise (compare (2)) on the diagonal. (14) is again solved by the Levenberg-Marquardt algorithm.

This least squares problem can again be represented by a graph consisting of nodes that present variables and edges presenting soft constraints. The associated graph is shown in Figure 5. The pose q_i is colored in green indicating a variable. Landmarks are shown in red indicating a fixed variable that appear only as constants in the optimization process and are not estimated. An edge connecting pose and landmark corresponds to the soft constraint associated with $\pi(q, l) - z$. Note that landmarks are kept fixed during localization hence yielding a low dimensional optimization problem. The subgraph shown in gray is conditionally independent of the rest given the landmark position and is therefore solved on its own.

V. Experiments

First we present our experimental setup. Thereafter results of the mapping process are presented before describing the localization results.

Our vehicle is equipped with a stereo rig with a base width of 30cm. The cameras field of view is approximately 80°. Image resolution is 1263×389 pixels after rectification.

Two passes through inner city villages are used to benchmark the proposed algorithm. The first pass is used for mapping whereas the second is used for testing the localization method. Each trajectory is approximately one kilometer in length. During mapping stereo frames are processed and landmark IDs are assigned to all salient points as described in Section III. Thereafter the constraints are solved for the map. A top view of a part of the map as it is estimated by our method is depicted in Figure 3. Green indicate landmarks and red points show the estimated poses. The entire map consists of 1,200 poses and approximately 500,000 landmarks. Recent progress in SLAM allows to compute the entire map in one sweep. Estimating the map takes roughly twenty minutes on off-the-shelf computing hardware. If the area to be mapped is very large then the map computation can easily be broken up into slightly overlapping chunks of manageable size. Each chunk can be computed independently keeping the overlap fixed and hence allowing to compute arbitrarily large maps.

During online localization feature points are extracted and matched with landmarks of the map. Extracting salient points takes approximately 15ms, computing gray value feature vectors requires roughly 8ms for all extracted points and matching finally takes another 25ms. Timing results for solving the non-linear least squares problem for the current ego position range from 3ms to 15ms depending on feature configuration and number. The total computing time stays well below 100ms corresponding to 10Hz of our camera setup. Figure 8 shows matches between the current image and the map.

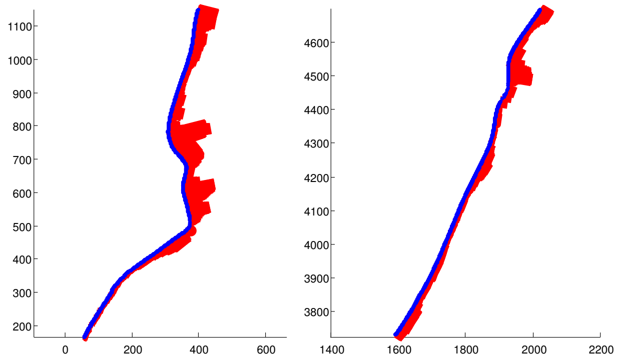

Fig. 7: Localization results for two sample trajectories. Axis are in meters. Red denotes the magnitude of error amplified for better visibility.

For evaluation purposes, we recorded high precision DGPS data during data acquisition which we consider ground truth. Our localization results for each of the trajectories are illustrated in Figure 7. Blue denotes estimated ego poses whereas red indicates the error compared to our ground truth data. Note that the error is scaled for better visibility. The pose error is 32.4 cm on average. Moreover, it can be seen that localization accuracy varies over the trajectory. This is caused by variability in imaging conditions, scene uniqueness and occlusion.

VI. Conclusion and Future Work

Herein we have proposed to use cameras as a sole sensor for localizing the ego vehicle. To this end we have computed a map consisting of 3D landmarks accompanied by their visual description. This map is computed by a large scale non-linear least squares problem. Outliers are reliably detected during estimation and pruned from the constraint graph. Moreover, rough GPS readings are fused into the estimation problem fixing the map globally.

During online localization the map is queried, landmarks of the immediate vicinity are retrieved and matched into the current stereo image pair. Matches are finally exploited yielding a judicious ego pose estimate. The proposed algorithm runs in real time and achieved a mean localization error of 32.4 cm in an urban scenario.

Integrating odometry reading of on-board vehicle sensors into the localization seems exciting ground for future research. Moreover, it seems quite straight forward to replace the stereo cameras during localization by their monoscopic counter parts. Furthermore, we plan to advance the vision

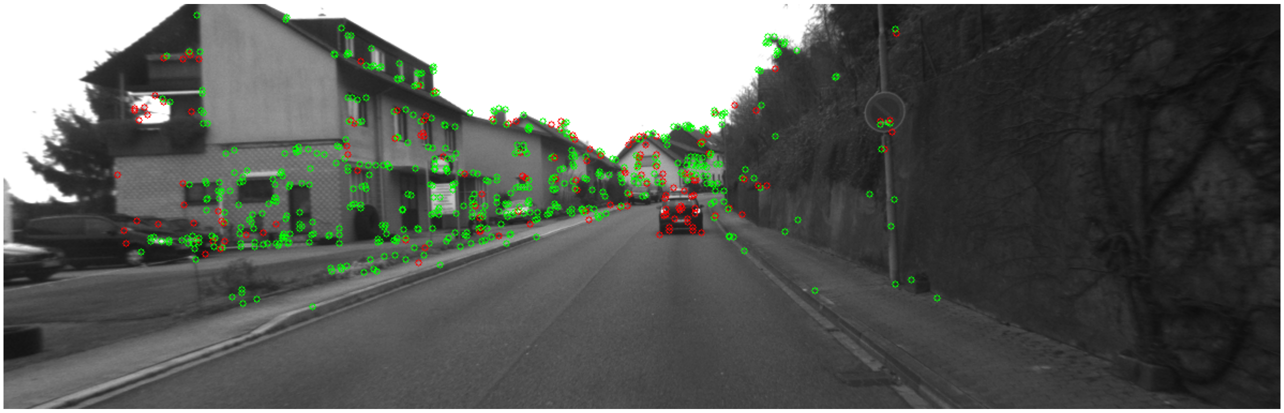

Fig. 6: Mapping sequence: Red landmarks are flagged as outliers.

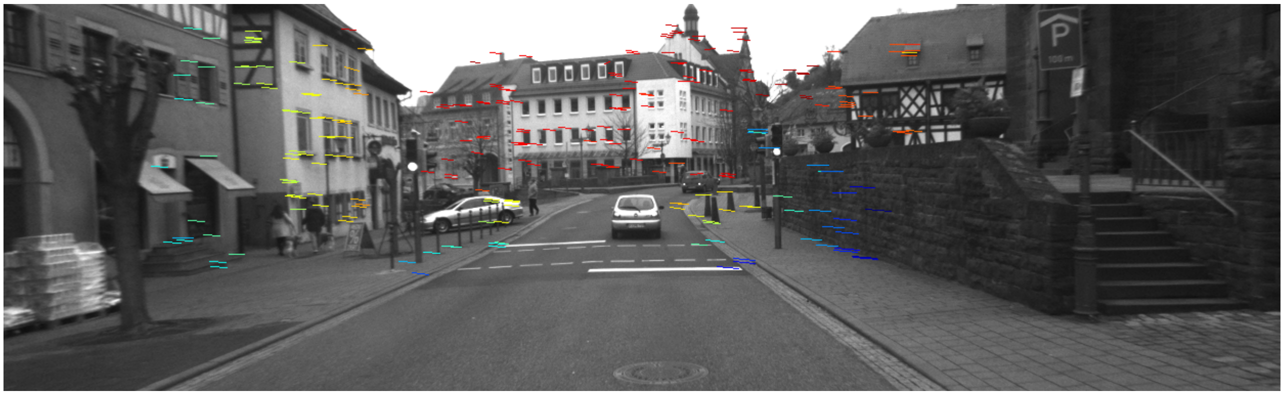

Fig. 8: Online sequence: Matches between the current image and map landmarks are shown.

front end to handle more challenging feature matching situations. Finding appropriate descriptors that are robust to severe lighting changes between mapping and localization is of paramount importance for continuous and life long localization of any mobile robot. Finally, we strive at thinning out the map to further reduce memory consumption.

References

[1] M. Agrawal and K. Konolige, "Real-time localization in outdoor environments using stereo vision and inexpensive gps," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3. IEEE, 2006, pp. 1063–1068.

[2] H. Badino, D. Huber, and T. Kanade, "Visual topometric localization," in *Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, June 2011.

[3] M. Bosse and R. Zlot, "Continuous 3D scan-matching with a spinning 2D laser," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 4244–4251.

[4] Z. Chen, J. Samarabandu, and R. Rodrigo, "Recent advances in simultaneous localization and map-building using computer vision," *Advanced Robotics, 21*, vol. 3, no. 4, pp. 233–265, 2007.

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

[6] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *Intelligent Transportation Systems Magazine, IEEE*, vol. 2, no. 4, pp. 31–43, 2010.

[7] R. Hartley and A. Zisserman, *Multiple view geometry*. Cambridge university press Cambridge, UK, 2000, vol. 642.

[8] K. Konolige, M. Agrawal, and J. Sola, "Large-scale visual odometry for rough terrain," *Robotics Research*, pp. 201–212, 2011.

[9] 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*. IEEE, 2011, pp. 3607–3613.

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

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

[12] F. Moosmann and C. Stiller, "Velodyne SLAM," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011, pp. 393–398.

[13] A. Napier, G. Sibley, and P. Newman, "Real-time bounded-error pose estimation for road vehicles using vision," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 2010, pp. 1141–1146.

[14] P. Piniés and J. Tardós, "Large-scale slam building conditionally independent local maps: Application to monocular vision," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1094–1106, 2008.

[15] 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*. IEEE, 2008, pp. 1–7.

[16] W. Press, B. Flannery, S. Teukolsky, W. Vetterling, *et al.*, *Numerical recipes*. Cambridge Univ Press, 1986, vol. 547.

[17] 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.

[18] H. Strasdat, A. Davison, J. Montiel, and K. Konolige, "Double window optimisation for constant time visual slam," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2352–2359.