Visual SLAM for Autonomous Ground Vehicles

Henning Lategahn, Andreas Geiger and Bernd Kitt

Abstract— **Simultaneous Localization and Mapping (SLAM) and Visual SLAM (V-SLAM) in particular have been an active area of research lately. In V-SLAM the main focus is most often laid on the localization part of the problem allowing for a drift free motion estimate. To this end, a sparse set of landmarks is tracked and their position is estimated. However, this set of landmarks (rendering the map) is often too sparse for tasks in autonomous driving such as navigation, path planning, obstacle avoidance etc. Some methods keep the raw measurements for past robot poses to address the sparsity problem often resulting in a pose only SLAM akin to laser scanner SLAM. For the stereo case, this is however impractical due to the high noise of stereo reconstructed point clouds.**

In this paper we propose a dense stereo V-SLAM algorithm that estimates a dense 3D map representation which is more accurate than raw stereo measurements. Thereto, we run a sparse V-SLAM system, take the resulting pose estimates to compute a locally dense representation from dense stereo correspondences. This dense representation is expressed in local coordinate systems which are tracked as part of the SLAM estimate. This allows the dense part to be continuously updated. Our system is driven by visual odometry priors to achieve high robustness when tracking landmarks. Moreover, the sparse part of the SLAM system uses recently published sub mapping techniques to achieve constant runtime complexity most of the time. The improved accuracy over raw stereo measurements is shown in a Monte Carlo simulation. Finally, we demonstrate the feasibility of our method by presenting outdoor experiments of a car like robot.

I. Introduction

A robot computing a map of a previously unknown environment while localizing itself within that map is referred to as Simultaneous Localization and Mapping (SLAM) [31], [11], [5]. Recently, cameras have been used as the sole sensor yielding visual SLAM (V-SLAM) [19], [7], [10].

The map is usually represented as a set of landmarks residing in 3D space. The camera traverses its environment yielding a trajectory. From each of the camera poses, a portion of the landmarks is observed. From these measurements, the most likely landmark positions and camera poses are estimated. Common methods designed to solve the estimation problem in real time rely on either Extended Kalman Filters (EKFs) [10], [25], [26], [9] or some variant of Bundle Adjustment (BA) [1], [29], [27], [30], [18]. However, both of these classes of methods are computationally very demanding. Therefore one usually seeks to limit the number of landmarks visible from each pose to remain computationally feasible. If for instance one included every pixel of every camera frame as a single landmark one would need to estimate the position of millions of landmarks after only a few time steps, thus burying any hopes for real time operability.

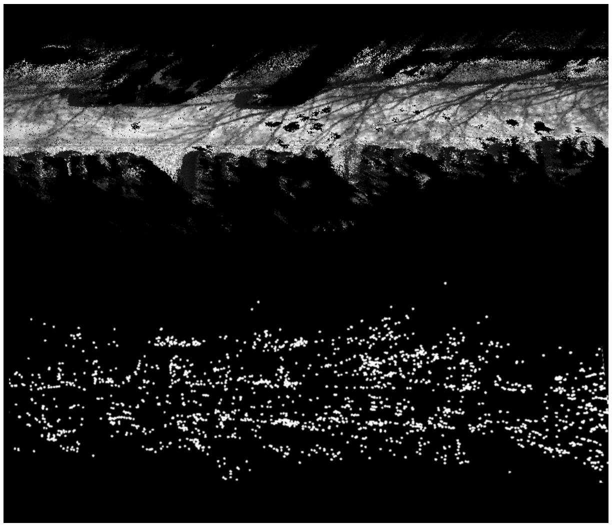

Fig. 1: The top shows a dense map computed by our algorithm. The bottom illustrates a map consisting of a sparse set of landmarks common in visual SLAM. The sparse map, however, is often insufficient for autonomous vehicles.

For tasks common in autonomous driving such as path planning, navigation, collision avoidance etc. these sparse map representations are often insufficient. Instead a dense map is sought. One way to address this problem is storing raw sensor measurements for a given number of camera poses. This is also common practice in laser scanner based SLAM systems [6]. However, the raw measurements are very noisy in the stereo camera case, making tasks in autonomous driving quite difficult.

We herein present a stereo V-SLAM system which computes dense maps that are more accurate than simply reconstructed disparity images. Our system is composed of two main parts. First a sparse V-SLAM system based on an EKF is run. The state vector of the EKF contains all landmark positions, the current camera pose and a, yet to specify, subset of past camera poses. To tackle the computational complexity problem inherent to EKF SLAM we utilize the sub mapping method of [25], [26] dubbed *conditionally independent sub maps* which has constant run time complexity most of the time. After incorporating new observations and updating the EKF state vector a new camera pose is obtained. We assume the transition from the immediately foregoing pose

H. Lategahn and A. Geiger and B. Kitt are with the Department of Measurement and Control, Karlsruhe Institute of Technology, 76131 Karlsruhe Germany. Email: henning.lategahn@kit.edu, geiger@kit.edu, bernd.kitt@kit.edu

to the current one to be of sufficiently high accuracy. This assumption has also been made in [22] for a monocular setup. During the second part, we compute a dense point cloud from stereo. This dense point cloud is a local map. It is derived by filtering disparity values for individual pixels of the image. One Kalman Filter is used for every pixel taking the local ego displacement for granted. The dense reconstruction of these disparities is expressed in local coordinate systems. Each one of the past camera poses which are tracked during the sparse V-SLAM spans one of these local coordinate systems. Put differently, each past camera pose has a dense local map attached. During loop closure the past camera poses are updated and drift is resolved. Thereby the global position of the local dense maps are also updated. The separation of the sparse map (part of the SLAM estimate) from the dense map (expressed in local coordinates, defined by the SLAM state vector) is the main contribution of our work. Figure 1 illustrates the difference between a sparse map comprised of a few hundred landmarks (bottom) and a dense map consisting of several magnitudes more points (top). The dense reconstruction shows parked cars on the side of the street. Shadows of trees can be seen on the road surface.

The paper is organized as follows. In Section II we review related work. In section III we present our method in more detail. The sparse SLAM algorithm is introduced in section III-A and the dense mapping is presented in section III-B. Experimental results are given in section IV. Finally a conclusion is drawn and ideas for future research are given in section V.

II. Related Work

Agrawal and Konolige [1] compute a skeleton of poses. Poses are connected by non-linear constraints obtained from feature matches between consecutive frames. The skeleton graph is then solved by BA. The map of their SLAM system is obtained by reconstructing the stereo frames at the nodes of the pose graph. Their work differs from our method in the underlying estimator (BA vs. EKF) and in the fact that their dense part of the map is not locally smoothed.

The work closest to ours is probably the work of Nieto and co-workers [24], [23]. A hybrid metric mapping method that embeds local coordinate systems to represent the dense map in is presented. The local coordinate systems are spanned by three or more landmarks of the state vector. The dense maps consist of 2D occupancy grids amongst others. The dense parts are also locally estimated like in our method. However, their maps are only two dimensional. Moreover, care must be taken to correctly choose appropriate landmarks to span each local coordinate system. Moreover, the scalability problem of SLAM is not explicitly addressed.

The system developed by Franke and co-workers [12], [4] also aims at increasing accuracy of stereo reconstructions by integrating disparity. However, their method relies on a known ego motion from external sensors. Moreover, the method is only used locally and not embedded into a SLAM framework.

III. Algorithm Overview

Figure 2 schematically shows the traversal of a robot. The triangles denote robot poses, the stars denote a sparse set of landmarks. Solid triangles denote past poses which are kept during estimation. The dense part is modeled by dense point clouds in our algorithm. However, in Figure 2 it is denoted by the "house" icons for clarity. Each past pose has a portion of the dense map attached. The associated dense parts are entirely expressed in coordinate systems defined by preceding poses.

Our algorithm consists of two phases which are both performed for every time step. During the first phase the set of landmarks $\{m_1, \ldots, m_M\} \subset \mathbb{R}^3$, some of the past poses $\{X_{t_1}, X_{t_2}, \ldots, X_{t_N}\}$ and the current pose X_T are stacked into a vector which is sequentially estimated by an EKF from stereo images. The poses are two spatial coordinates and orientation $X_t = (x_t, y_t, \psi_t)^T$. Implementation details are given further below. After each EKF update one obtains a new current pose X_T. The transition from the immediately foregoing pose to the current one ($X_{T-1} \to X_T$) is used to compute the dense representation of the environment during the second phase. We integrate the disparity values by individual Kalman Filters for every pixel. A thorough description is given in section III-B. The thus computed 3D position of the pixels are expressed with respect to one of the past poses of the EKF state vector. In Figure 2 the dark gray "houses" (corresponding to a portion of the dense map) are expressed in the coordinate system spanned by the pose X_{t_1} whereas the white "houses" are in the coordinate system of X_{t_2} etc. Therefore, each of the past poses has a small portion of the dense map attached to it. After loop closing a trajectory these past poses will be updated and hence will affect the global position of the dense embeddings.

The two phases are detailed below in sections III-A and III-B.

In the sequel we will assume that dense stereo correspondences are at our disposal. Dense disparity can nowadays already be computed by FPGAs and efficient matching algorithms are emerging (e.g. [13]).

A. Sparse EKF SLAM

The details of basic EKF SLAM will not be reviewed here for it being standard nowadays. For an introduction into the subject the reader is referred to [31]. It is rather focused on the prediction of the filter driven by visual odometry priors and the sub mapping techniques which is used to reduce computational complexity.

First a set of point-point matches between the previous and current left image is computed. These matches are computed by a computationally lean algorithm. In our case we have matched sets of maxima in the sobel filtered images (after non-maxima suppression). These putative matches are used to compute a first guess of the ego displacement. To this end the method of [16] is applied. The rotation matrix R and translation vector t is computed by non linear least squares. However, first only the rotation R is estimated from points

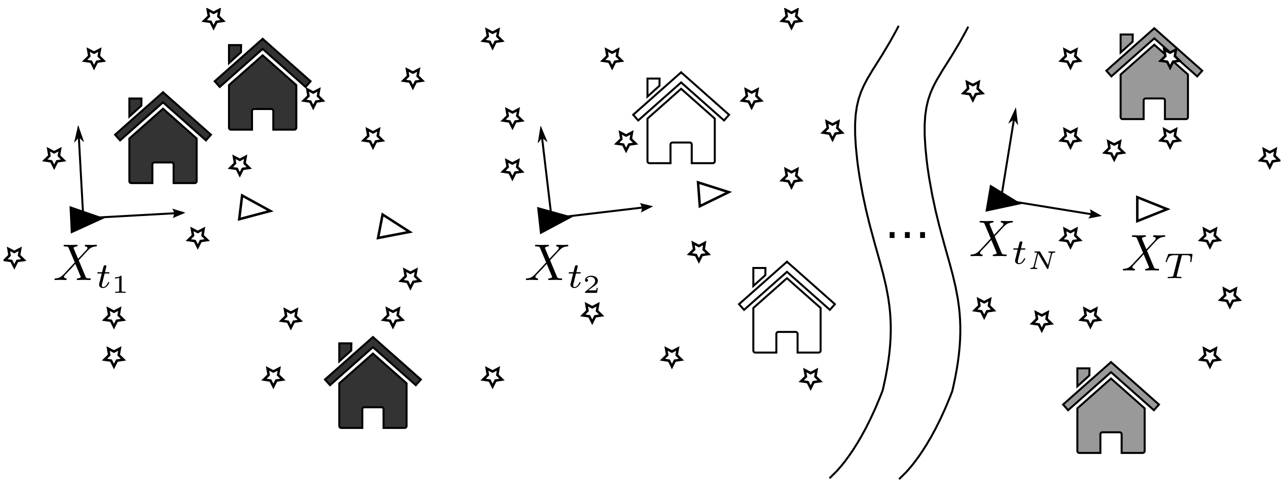

Fig. 2: Triangles denote robot poses. Solid triangles are a subset of past poses which are kept as part of the sparse EKF SLAM state. Stars denote landmarks. The dense map is schematically depicted by the house icons. The dense map is partitioned into small fragments. Each such fragment is expressed in local coordinates defined by the associated pose. The black houses are expressed with respect to X_{t_1}, the white houses with respect to X_{t_2} etc. X_T is the current pose.

far away from the sensor by solving

$$\hat{R} \quad = \quad \arg\min_{R} \sum_i \left|\left|z_{i,t} - \mathtt{proj}(R \cdot p_{i,t-1})\right|\right|^2 \qquad (1)$$

where $z_{i,t}$ is the measurement (pixel position and disparity) of the i^{th} point match in the current frame and $p_{i,t-1}$ is the reconstructed matched point of the previous time step. The stereo camera projection function is denoted by $\mathtt{proj}(\cdot)$. Thereafter the translation t is recovered while treating the rotation R as known. Again, a non-linear least squares problem

$$\hat{t} \quad = \quad \arg\min_{t} \sum_i \left|\left|z_{i,t} - \mathtt{proj}(\hat{R} \cdot p_{i,t-1} + t)\right|\right|^2 \qquad (2)$$

is solved for nearby point matches. Both steps are repeated in a RANSAC scheme. The RANSAC scheme runs very fast because only two matches are needed to generate a hypothesis for R and only one is required to compute a translation hypothesis t which is why we have chosen this method. Computing visual odometry priors for SLAM has been proposed in [2], [32].

This visual odometry is used to predict the position of the landmarks in the current left image. Landmarks are tracked by a KLT tracker [20]. If the KLT tracker yields a position that deviates more than a threshold from its predicted position this point-to-landmark association is invalidated. The filter state is also predicted from the visual odometry priors. At first glance it may seem cumbersome to first compute putative matches while thereafter actively searching for the landmarks (KLT). We find that investing this extra time is beneficial. Easily computable salient points yield unstable tracks whereas block matching search type methods are computationally quite demanding. Well initialized KLT features yield accurate and long lasting tracks. Also it is easily possible to compute a much higher number of putative matches than number of landmarks thus improving visual odometry. New landmarks are initialized such that landmarks are equally distributed over the entire image. Harris corners [14] have been used for landmark candidates.

To counteract the poor scalability of EKF SLAM algorithms we use *conditionally independent sub maps* as proposed by Piniés et al. in [25], [26]. The map is partitioned into local sub maps each containing its own state vector. The size of the state vector of each sub map is bounded thus achieving constant runtime complexity. At any time, a global update can be performed by backpropagating current observations down the sub map chain. More specifically two consecutive sub maps A and B are modeled by two Gaussians

$$\begin{aligned} p\left(\mathcal{X}_A, \mathcal{X}_C | z_a\right) &= \mathcal{N}\left(\begin{bmatrix} \hat{\mathcal{X}}_{A,a} \\ \hat{\mathcal{X}}_{C,a} \end{bmatrix} \middle| \middle| \begin{bmatrix} P_{A,a} & P_{AC,a} \\ P_{CA,a} & P_{C,a} \end{bmatrix} \right) \\ p\left(\mathcal{X}_C, \mathcal{X}_B | z_a, z_b\right) &= \mathcal{N}\left(\begin{bmatrix} \hat{\mathcal{X}}_{C,ab} \\ \hat{\mathcal{X}}_{B,ab} \end{bmatrix} \middle| \middle| \begin{bmatrix} P_{C,ab} & P_{CB,ab} \\ P_{BC,ab} & P_{B,ab} \end{bmatrix} \right) \end{aligned}$$

where capital letters indicate map membership and lowercase letters indicate the observations which have been used to estimate the states, that is z_a are all observations seen from poses of sub map A and z_b are measurements from sub map B. C denotes a common part of the two sub maps. $\mathcal{X}_C$ can be interpreted as the overlap between the two maps. A back propagation yields an estimate of the entire map modeled by

$$p\left(\mathcal{X}_A, \mathcal{X}_C, \mathcal{X}_B | z_a, z_b\right) = \qquad (3)$$
$$\mathcal{N}\left(\begin{bmatrix} \hat{\mathcal{X}}_{A,ab} \\ \hat{\mathcal{X}}_{C,ab} \\ \hat{\mathcal{X}}_{B,ab} \end{bmatrix} \middle| \middle| \begin{bmatrix} P_{A,ab} & P_{AC,ab} & P_{AB,ab} \\ P_{CA,ab} & P_{C,ab} & P_{CB,ab} \\ P_{BA,ab} & P_{BC,ab} & P_{B,ab} \end{bmatrix} \right)$$

where all observations (z_a, z_b) have been considered. This back propagating update step is accomplished by the following equations

$$\begin{aligned} K &= P_{AC,a} P_{C,a}^{-1} && (4) \\ P_{AC,ab} &= K P_{C,ab} && (5) \\ P_{A,ab} &= P_{A,a} + K(P_{CA,ab} - P_{CA,a}) && (6) \\ \hat{\mathcal{X}}_{A,ab} &= \hat{\mathcal{X}}_{A,a} + K(\hat{\mathcal{X}}_{C,ab} - \hat{\mathcal{X}}_{C,a}). && (7) \end{aligned}$$

After backpropagation, the global map (equation 3) is an exact solution to the non-submapped EKF SLAM. No approximations are introduced by using the sub mapping technique. Moreover, the backpropagation is linear in the number of sub maps and can be launched rarely yielding an EKF SLAM algorithm which is constant most of the time. For a detailed derivation of this method see [25], [26].

B. Dense Local Maps

It is a well known phenomena that stereo reconstructions suffer from a high noise level in depth direction. This is caused by the uncertain disparity estimate from which depth is inferred. For an investigation of this phenomena see [28]. Therefore we filter disparities by iconic Kalman filters operating on each pixel of the image. The state of the filter is the disparity. At this point we assume the ego motion from pose X_{T-1} to X_T to be known and sufficiently exact. The filter states are predicted by reconstructing the previous state (from pixel position and current disparity state), compensating the ego motion and backprojecting it into the current image. More specifically, let u, v be the pixel position with Kalman filter state d_{T-1}. The prediction is computed by

$$\begin{aligned} p_{T-1} &= \mathtt{reconst}(u, v, d_{T-1}) && (8) \\ \bar{p}_T &= \Phi\left(p_{T-1} | X_{T-1}, X_T\right) && (9) \\ \left[\bar{u}_T, \bar{v}_T, \bar{d}_T\right] &= \mathtt{proj}(\bar{p}_T) && (10) \end{aligned}$$

where $\bar{\cdot}$ denotes prediction, $\mathtt{reconst}(\cdot)$ is the stereo reconstruction and $\Phi(\cdot)$ compensates ego motion. The predicted pixel position $\bar{u}_T, \bar{v}_T$ is used for association. Once a track is lost it is added to the local coordinate system permanently. If a loop closure occurs, the displacement from X_{T-1} to X_T may be too large. In such cases the ego motion is compensated by the visual odometry priors (equations 1, 2) i.e. $\bar{p}_T = \Phi(p_{T-1} | \hat{R}, \hat{t})$.

This method of increasing depth accuracy for stereo cameras has been used in [15], [12], [4]. All these methods are based on the work of Matthies and co-workers [21].

Observe, that the dense part of the map is tracked in camera coordinates. However, for our method, it needs to be transformed into the coordinate system which is spanned by the last pose $X_{t_N} = (x_{t_N}, y_{t_N}, \psi_{t_N})^T$ that has been permanently added to the state vector.

IV. Experimental Results

In the following we present experimental results. First we validate our claim, that the iconic Kalman Filters improve reconstruction accuracy. We show that the standard deviation of the depth of a point is considerably lower compared to the non-filtered points. Second, we show results of a 3D map computed from a real world sequence.

A. Simulation Results

To assess the improvement in depth accuracy we have run a Monte Carlo simulation as follows. First we generate a trajectory of the camera. Then landmarks are placed around the trajectory. During the simulation run, the camera traverses the environment and perceives the landmarks once they are within a predefined distance to the camera. The exact pixel positions and disparities, which are computed from the locations of the landmarks, current camera pose and camera parameters is disturbed by Gaussian noise with covariance matrix $\sigma_{cam}^2 I$. This is taken as the measurement and fed into the SLAM algorithm. The exact ego motion (yaw rate and speed) is also disturbed by Gaussian noise with standard deviations $\sigma_{yaw}, \sigma_{speed}$. The disturbed ego motion is an approximation of the ego motion priors of equations 1 and 2. This part of the simulation only affects the sparse SLAM system.

The dense local map is simulated by 1000 independent tracks of the same point. This point is also added to the simulation environment. The virtual camera perceives this point (pixel position and disparity) 1000-fold for each time step, again disturbed by independent Gaussian noise with covariance $\sigma_{cam}^2 I$. Each occurrence of this point is tracked by an iconic Kalman Filter as presented before. At each time step, all of the tracks of the dense local map are reconstructed and the *empirical* standard deviation of the depth is computed. By using many realizations of the same point (1000 in our case) we obtain a faithful approximation of the expected accuracy. Note that this point is not used as a landmark in the sparse SLAM system.

The base width of the stereo camera setup used in our simulation is 57.5 cm. The standard deviation of the disparity matcher is set to $\sigma_{cam} = 1.0px$ and the ego motion is disturbed by $\sigma_{yaw} = 0.3\frac{rad}{s}$ and $\sigma_{speed} = 0.6\frac{m}{s}$. The ego velocity is $10\frac{m}{s}$ and points are observed at 10 Hz. Figure 3 shows the mean depth of the iconically tracked point over time in green. It is first observed by the camera at a distance of 40 meters. The empirical standard deviation of the depth of all tracks is shown as a confidence interval in blue (amplified by a factor of five for better visibility).

Moreover, we have reconstructed each occurrence of the point of the dense local map from its non-filtered measurements. The empirical standard deviation of the depth is computed for every time step and plotted in red in Figure 3 (again five-fold amplified). It can be seen that the accuracy of the filtered point is considerably better.

B. Real World Experiments

In the following we present experimental results of a car like robot.

We have equipped our testing vehicle with stereo cameras with a base width of approximately 57.5 cm. The region of interest of our cameras is of size 1350×370 pixels with an opening angle of 90 degrees. The disparity images are computed by a block matching based stereo matcher as presented in [13]. The stereo matcher produces very dense disparity images which plays into our hands nicely. Images

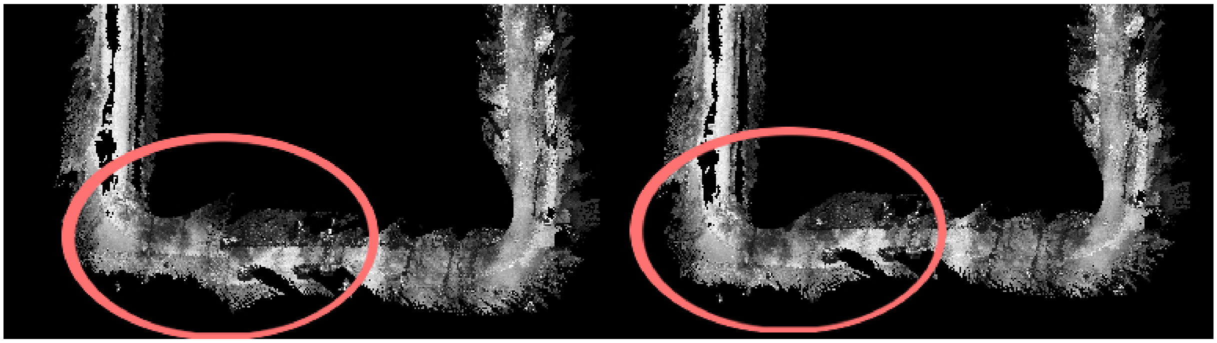

Fig. 5: The bottom part of the dense map before (left) and immediately after (right) closing the loop. A slight miss alignment can be observed on the left bottom of the loop. After loop closure this miss alignment is successfully resolved. The position of past poses is shown in Figure 4.

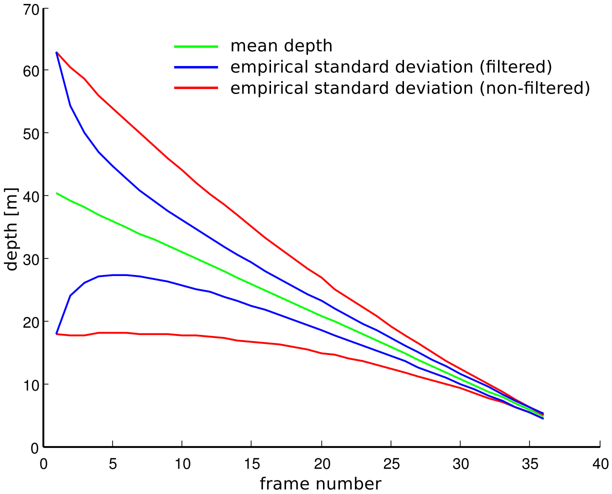

Fig. 3: Results of a Monte Carlo simulation are illustrated. The mean depth of 1000 independent Kalman tracks of the same point of the dense local map is shown over time in green. The *empirical* standard deviation is depicted in blue (five-fold amplified for clarity). For comparison, the empirical standard deviation of the raw reconstruction (non-filtered) is shown in red. A considerable improvement of the accuracy of the filtered point over the raw reconstruction can be seen.

are recorded with 10 Hz. We have driven a loopy trajectory of approximately 520 meters on our campus with speeds of up to 30 $\frac{km}{h}$. Note that the loop closure has been triggered manually since place recognition is not the focus of our contribution.

Figure 4 depicts the chain of past poses $X_{t_1}, \ldots, X_{t_N}$ from a bird's eye perspective. The left part of Figure 4 shows the poses directly before closing the loop. On the bottom left corner a drift of approximately 5 m can be observed. The right part of the Figure shows the past poses immediately after closing the loop. However, these poses are only part of the sparse mapping part of our algorithm.

Each one such pose has a portion of the dense map attached to it. The resulting dense maps are shown in Figure 5. Again the left half illustrates the dense map immediately before closing the loop whereas the right half shows the dense map after loop closure. It can be seen how the left bottom corner of the map is misaligned directly before the drift is resolved. After loop closure, however, the dense map fragments align well.

Our system is mostly implemented in MATLAB and running times are currently not real-time. However, we plan to refactor our system and reimplement a C++ version which is expected to reach real-time capabilities.

V. Conclusion and Future Research

Herein we have presented a method to compute dense local maps which are embedded into a sparse SLAM algorithm. The dense local maps are continuously updated and 3D global maps are obtained after loop closure. Moreover, it was shown experimentally that our dense maps are potentially more accurate than raw stereo reconstructions.

The sparse SLAM algorithm estimates a subset of past robot poses jointly with a sparse set of landmarks. This estimation is achieved by conditionally independent sub maps [25], [26] which are based on an EKF and has constant runtime complexity most of the time. The prediction of the filter state is driven by visual odometry priors computed from a set of cheaply computed point matches.

The dense map is computed from the local ego displacement (from pose to pose) and dense disparity images. To increase reconstruction accuracy, the disparities are tracked by iconic Kalman filters. Their reconstructed 3D position is expressed in coordinate systems spanned by past robot poses which are part of the sparse EKF SLAM state vector.

The presented algorithm brings some flexibility. The sparse SLAM algorithm can easily be replaced by any one of the published SLAM algorithms. It can thereby be tuned to specific needs. Moreover, the dense local mapping algorithm which merely needs the ego displacement and dense disparity images can be enhanced as long as the map is expressed in local coordinates.

In the future we plan to integrate an appearance based place recognition engine into our SLAM system to automatically detect loop closures. The method of Cummnis and Newman

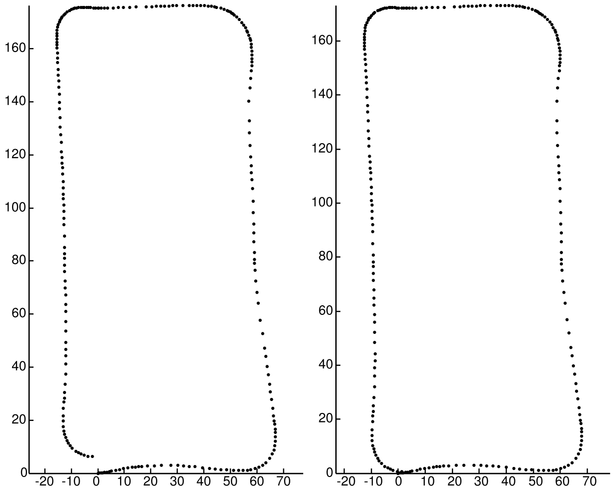

Fig. 4: The set of past robot poses which span a local coordinate system each is depicted. On the left the position directly before loop closure is shown. The right shows the position immediately after loop closure. The driving direction is counter clockwise. The bottom part of the dense map for that trajectory is depicted in Figure 5.

[8] seems promising. Moreover, we believe that developing more advanced locally dense mapping algorithms is exciting ground for future research. Specifically, we plan to fuse dense optical flow and disparity. Dense flow algorithms are becoming increasingly feasible [17]. Furthermore, the detection and tracking of moving objects (e.g. [3]) as an intermediate step will improve the operation in dense urban traffic.

References

[1] M. Agrawal and K. Konolige. FrameSLAM: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, 2008.

[2] P.F. Alcantarilla, L.M. Bergasa, and F. Dellaert. Visual odometry priors for robust EKF-SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[3] Alexander Bachmann and Hildegard Kuehne. An iterative scheme for motion-based scene segmentation. In *Workshop on Dynamical Vision (ICCV)*, pages 735–742, Kyoto, Japan, 2009.

[4] H. Badino, U. Franke, and R. Mester. Free space computation using stochastic occupancy grids and dynamic programming. In *Workshop on Dynamical Vision (ICCV)*, 2007.

[5] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *Robotics and Automation Magazine*, 13(3):108–117, 2006.

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

[7] Z. Chen, J. Samarabandu, and R. Rodrigo. Recent advances in simultaneous localization and map-building using computer vision. *Advanced Robotics, 21*, 3(4):233–265, 2007.

[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, 2008.

[9] A.J. Davison and N. Kita. 3D Simultaneous Localisation and Map-Building Using Active Vision for a Robot Moving on Undulating Terrain. 2001.

[10] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1052–1067, 2007.

[11] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *Robotics and Automation Magazine*, 13(2):99–110, 2006.

[12] U. Franke, S. Gehrig, H. Badino, and C. Rabe. Towards optimal stereo analysis of image sequences. *Robot Vision*, pages 43–58, 2008.

[13] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Asian Conference on Computer Vision (ACCV)*, Queenstown, New Zealand, November 2010.

[14] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[15] C. Hoilund, TB Moeslund, CB Madsen, and MM Trivedi. Improving stereo camera depth measurements and benefiting from intermediate results. In *Intelligent Vehicles Symposium (IV)*, pages 935–940. IEEE, 2010.

[16] M. Kaess, K. Ni, and F. Dellaert. Flow separation for fast and robust stereo odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3539–3544, 2009.

[17] Bernd Kitt, Benjamin Ranft, and Henning Lategahn. Block-matching based optical flow estimation with reduced search space based on geometric constraints. In *IEEE International Conference in Intelligent Transportation Systems (ITSC)*, Madeira Island, Portugal, September 2010.

[18] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *IEEE and ACM International Symposium on Mixed and Augmented Reality (IS-MAR)*, pages 1–10. IEEE Computer Society, 2007.

[19] T. Lemaire, C. Berger, I.K. Jung, and S. Lacroix. Vision-based slam: Stereo and monocular approaches. *International Journal of Computer Vision (IJCV)*, 74(3):364, 2007.

[20] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International joint conference on artificial intelligence*, volume 3, pages 674–679, 1981.

[21] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision (IJCV)*, 3(3):209–238, 1989.

[22] R.A. Newcombe and A.J. Davison. Live dense reconstruction with a single moving camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1498–1505. IEEE, 2010.

[23] J. Nieto, J. Guivant, and E. Nebot. Denseslam: Simultaneous localization and dense mapping. *International Journal of Robotics Research*, 25(8):711–744, 2006.

[24] J.I. Nieto, J.E. Guivant, and E.M. Nebot. The hybrid metric maps (HYMMs): A novel map representation for DenseSLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, 2004.

[25] P. Piniés and J.D. Tardós. Scalable SLAM building conditionally independent local maps. In *IEEE conference on Intelligent Robots and Systems (IROS)*, 2007.

[26] 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.

[27] G. Sibley. Relative bundle adjustment. *Department of Engineering Science, Oxford University, Tech. Rep*, 2307(09), 2009.

[28] G. Sibley, L. Matthies, and G. Sukhatme. Bias reduction and filter convergence for long range stereo. *Robotics Research*, pages 285–294, 2007.

[29] G. Sibley, C. Mei, I. Reid, and P. Newman. Adaptive relative bundle adjustment. In *Robotics Science and Systems Conference (RSS)*, 2009.

[30] 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.

[31] S. Thrun. Robotic mapping: a survey, Exploring artificial intelligence in the new millennium, 2003.

[32] B. Williams and I. Reid. On combining visual slam and visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.