Submap-Based SLAM for Road Markings

Eike Rehder[1], Alexander Albrecht[1]

***Abstract*— Coherent road maps are a prerequisite for autonomous navigation. In case of an unknown environment, grid map and SLAM techniques are widely used. This paper takes a novel approach to vision based mapping of road markings by registration of local occupancy gridmaps for map stitching. We show that with reasonably accurate ego motion measurements, seamless global maps can be constructed from local grid maps. The approach is evaluated on real world data obtained from an autonomous model racing car.**

I. Introduction

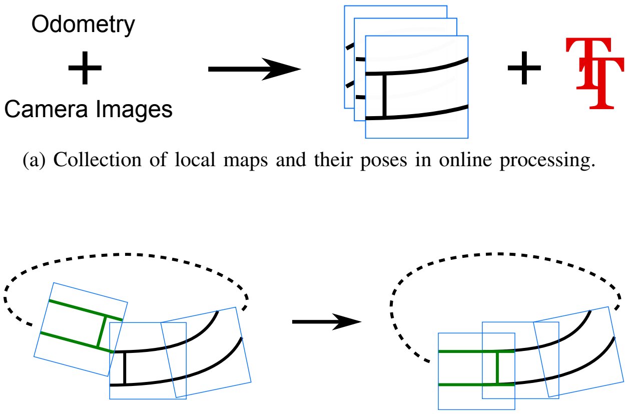
In autonomous navigation, the use of maps of the environment plays a significant role for localization. Yet, maps in sufficient accuracy are not widely available. However, robots operating in unknown environments as well as autonomous vehicles are equipped with increasingly accurate sensors to perceive their surrounding, enabling them to create a map on the fly. The problem of imperfect ego motion estimation combined with imperfect environment perception has given rise to the well known simultaneous localization and mapping (SLAM) approach. In SLAM frameworks, the autonomous agent constructs the map it wishes to localize itself on by maximizing the joint probability of map and ego position.

SLAM approaches have been tackled in two different implementations, online and offline SLAM. In the online version, all sensor readings of both ego motion and environment perception are integrated into map and localization on the fly. However, errors once made can rarely be corrected in hindsight.

In contrast to that, in offline SLAM all the data is collected during an exploratory drive. Upon arrival, a map is constructed of the recently discovered environment, thus enabling the use of global optimization. However, up to the point where full map processing is executed, all data has to be stored.

Both cases of SLAM are strongly dependent on distinct features of the environment. For mapping of road marking, this condition is not always met due to the high degree of ambiguity of features.

In this work we introduce a novel approach to visual offline mapping of road markings. A brief overview can be found in Figure 1. Our work is based on the assumption that today's odometry measurements are fairly accurate so that errors are locally bounded. Under this assumption, we construct local maps making use of the well understood occupancy grid map. During a drive through the unknown environment, we collect a multitude of these local maps together with their individual poses T relative to their predecessors. Upon arrival at our starting position, a global optimization is run on all map poses to allow optimal registration between the maps. This way a globally consistent map can be constructed in hindsight while the amount of data in storage can be reduced. The work was executed in the context of the Carolo-Cup, a student competition on autonomous model race cars, hosted by the University of Braunschweig [1].



(a) Collection of local maps and their poses in online processing.

(b) Offline optimization for consistent map stitching.

Fig. 1: Steps performed in the algorithm.

The remainder of this paper is structured as follows. In section II we give a brief overview over common mapping techniques including occupancy mapping and registration. Our approach of combining grid mapping and global optimization is introduced in section III and we evaluate the results of this approach in section IV. The paper is concluded in section V.

II. Related Work

In this section we give a brief overview over some existing approaches in mapping, both local and global, as well as on registration of images and point clouds which is closely related to the problem of map stitching.

A. Occupancy Grids

Local mapping is often performed in so called occupancy grid maps [2]. In general, occupancy grid maps are a discretization of the surrounding of an autonomous agent into cells. It is assumed that all cells are independent of each other. This assumption allows relatively simple map maintenance as every cell in itself can be represented as a Bayesian estimator for an occupancy probability. For map

[1] Eike Rehder and Alexander Albrecht are with the Institute of Measurement and Control Systems, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany eike.rehder@kit.edu

updates, all measurements are assigned to map cells. The corresponding cells are updated by the use of an inverse sensor model which relates a measurement to the occupancy probability. For most applications of grid mapping, the poses of the vehicle are assumed to be known perfectly.

B. *Simultaneous Localization and Mapping*

When the motion of the agent is subject to uncertainty, simultaneous localization and mapping (*SLAM*) is performed. A good overview can be found in [3] and [4]. In this approach, a map is created and updated by maximizing the posterior probability of motion and map, i.e. motion and map are estimated at the same time from sensor measurements. Different approaches exist to handle the SLAM problem, depending on map representation and update. In one of the most used solutions, the mapping is represented by a Kalman filter where landmark positions and vehicle pose are tracked in the same state vector, giving rise to the EKF SLAM [5].

In graphical SLAM, the landmarks and vehicle poses are represented as a graph which is optimized in hindsight [6]. In FastSLAM, map and robot pose are maintained in form of a particle filter where each particle represents a pose together with a map consisting of Kalman Filter based landmark representations[7].

FastSLAM can also be applied together with occupancy grid maps [8]. In this case, the map is stored in form of an occupancy grid rather than individual landmarks.

A special case is the Atlas SLAM in which submaps are created from scan matching [9]. These submaps are later registered with respect to each other to obtain a globally consistent map.

C. *Registration*

The term registration refers to the problem of finding the rotation and translation of one dataset so that it is in congruence with a template. A well known application for this are panoramic images which are stitched from many smaller images.

The registration is computed easily if corresponding points in dataset and template are known, e.g. by optimization [10] or singular value decomposition.

However, correspondences are rarely known exactly. Thus, the main problem of registration remains in the correspondence search. In computer vision, interest points are matched using local descriptors [11] [12]. Robustness is gained by using RANSAC approaches to descriptor matching [13].

Another well known algorithm is the iterative closest point algorithm (*ICP*) in which the correspondences are iteratively optimized by matching points of a dataset to their nearest neighbor within the template [14]. However, this algorithm strongly depends on a good initial registration.

III. Grid based Atlas SLAM

This paper takes a different approach to offline SLAM. As introduced in [9], similar to an atlas, the entire map is divided into small submaps that are put in relation to each other over a pose graph as depicted in Figure 2. In our representation, the nodes of the graph represent maps with their relative pose shown in red arrows to their predecessor while edges represent the overlay matching of the individual maps as well as the deviation from the pose obtained from odometry shown in green.

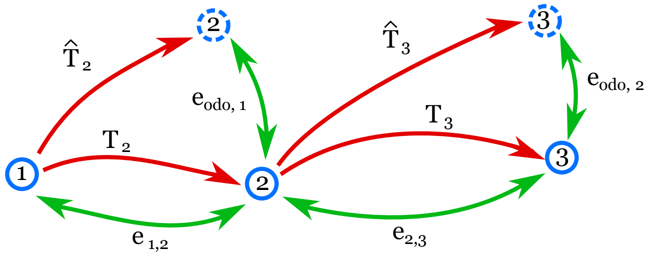


Fig. 2: Problem construction. Blue solid circles denote optimized maps, dashed circles denote predicted locations from odometry measurements. Poses are coupled with transformations shown as red arrows. Green arrows denote the error measures to be minimized.

In our implementation, two steps are executed, namely local mapping and global map alignment. The first, the acquisition of submaps, is based on occupancy grid maps and can be performed online. A map M_i consists of a tuple of a discretized occupancy grid map Θ_i and the map's pose relative to the previous map T_i together with the pose's covariance matrix Σ_{odo_i}. The pose is in the homogeneous form of

$$T_i = \begin{bmatrix} R_i & \vec{t_i} \\ 0 & 1 \end{bmatrix} \tag{1}$$

where R_i is a 2-D rotation matrix and $\vec{t_i}$ is a 2-D translation vector.

The second step, the global optimization, is performed upon arrival at the starting location. The optimization goal is to find a set of poses T^N for N maps so that the registration error between all overlapping maps and the deviation of the optimized poses from the original odometry poses is minimal.

In the following, the two steps are explained in detail.

A. *Local Mapping*

The local mapping uses the concept of occupancy grid maps. In their original formulation, these maps were used as a discretized representation of the environment for obstacles that would block the way for autonomous agents. However, the concept itself as a Bayesian filter map is quite general and can be applied to mapping of arbitrary environmental properties (e.g. [15]). Since our algorithm is tailored to fulfill the purpose of road mapping, we map road markings extracted from onboard camera images. We utilize occupancy grid mapping due to easy feature maintenance even in the presence of noisy observations.

In the following, the necessary steps of motion estimation as well as feature extraction are explained.

1) Motion Estimation: Motion estimation is performed using an onboard sensor setup consisting of two wheel encoders attached to left and right rear wheels, a 3-axis gyroscope as well as a 3-axis accelerometer. The measurements obtained from the sensors are fused in an Extended Kalman Filter framework with an extended single track motion model.

The discrete motion equation from a previous pose defined by $(x_{t-1}, y_{t-1}, \theta_{t-1})^T$ can be computed by the momentary curve radius r_t and angular velocity ω_t according to

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} -r_t sin(\theta_{t-1}) + r_t sin(\theta_{t-1} + \omega_t \Delta t) \\ r_t cos(\theta_{t-1}) - r_t cos(\theta_{t-1} + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix} . \tag{2}$$

2) Feature Extraction and Accumulation: In general, all kinds of features can be used for grid mapping. Most common is the mapping of obstacles sensed with 3D sensor systems. In the context of our field of application however, we focus on the mapping of road markings from camera images. For this, we extract features that hint to road markings and accumulate those in a standard gridmap using the Bayesian formulation as introduced in [2].

Features are extracted in multiple steps (see Figure 3). In the first step, we extract a region of interest where we expect to find the road markings, e.g. the lower third of the image. From extrinsic calibration, we compute the homography to project the image to birds eye view. Since the region of interest is selected to be relatively small and our robot's chassis is near rigid, we neglect roll and pitch angles for transformation.

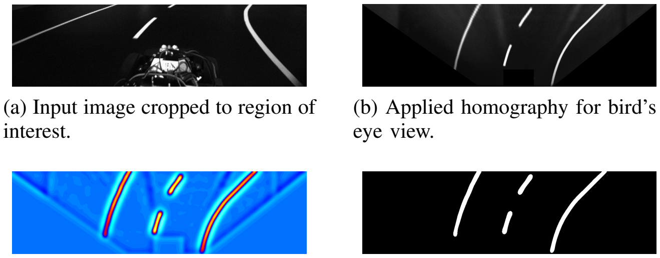
Due to the transformation in top view, the perspective distortion of road markings in the image is corrected and a matched filter for the size of the markings can be applied. We selected a 2-D difference of Gaussian (*DoG*) as a filter, as the filter response is close to the top hat filter but features additional noise attenuation.

The final update of the map is performed using a binarized version of the filtered image. The image is transformed to map coordinates as depicted by the green trapezoid in Figure 4. Additionally, every pixel value $p_i \in \{0, 1\}$ is modified according to

$$\tilde{p}_i = \alpha p_i - \beta \tag{3}$$

where α, β are positive constants depending on the expected reliability of the binarized image. This way, in *logOdds* representation high filter responses will yield a positive update and vice versa. For the final map update, the modified image is added to the map.

Whenever the field of view (*FOV*) of the camera would lie outside of the current map a new map is created. Its initial pose is set so that the vehicle's orientation is aligned to the map's x-axis and the vehicle's position within the map is at the left border and at half the width of the map (see Figure 4).



(a) Input image cropped to region of interest.

(b) Applied homography for bird's eye view.

(c) Filter response map.

(d) Binarized feature map.

Fig. 3: Process steps in feature extraction.

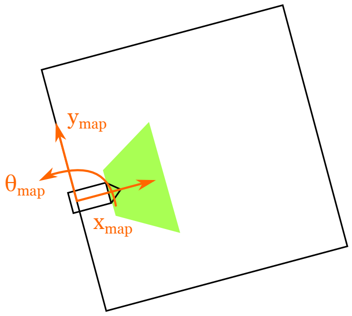


Fig. 4: Grid initialization and update with FOV.

B. Global Alignment

After local mapping of the surrounding, a global association and optimization step is performed. As we use local grid maps, the association and pose optimization comes down to a registration problem with unknown correspondences. However, if we assume a fairly accurate initial pose estimation, we can easily apply active correspondence matching. In order to do that, we try to find a set of poses for all nodes, individually given by T_i that minimizes the error w.r.t. odometry measurements $\hat{T}_i$ and registration error from the map content.

For this we iteratively repeat map data association and pose optimization similar to the ICP algorithm. The procedure of data association is explained in Section III-B.1 followed by the optimization in Section III-B.2.

1) Data Association: Data association has to be done in multiple levels. For once, expected overlap of maps has to be found. In a second step, the content of overlapping maps has so be associated for registration. A common approach is key point matching using different classes of detectors and descriptors. However, as we are dealing with road markings and thus with few distinct features but a high degree of ambiguity, we apply point cloud matching for registration.

To obtain point clouds from grid maps, we extract all map points from the grid maps that belong to the set of road markings with a probability value above some predefined threshold.

Then all consecutive map poses are computed so that the relative poses between any two maps i, j can be constructed

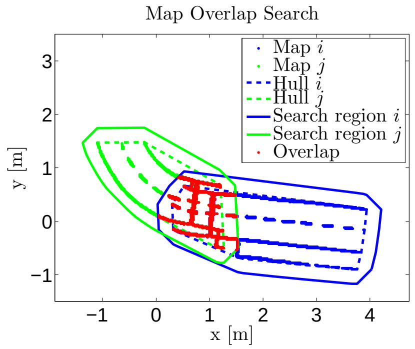


Fig. 5: Correspondence search for maps.

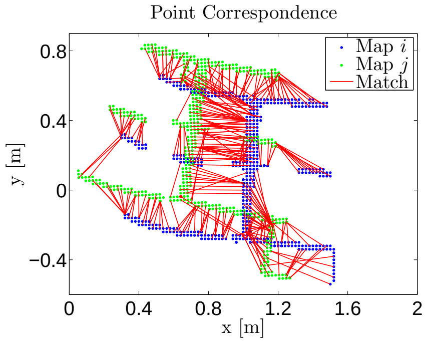


Fig. 6: Point correspondence with local orientation criterion.

according to

$$T_{i,j} = T_j T_{j-1} \cdots T_{i+1} T_i. \tag{4}$$

In the next step, we compute the convex hull for all maps for map overlap assessment. Two maps are expected to overlap if their convex hulls overlap. However, due to pose uncertainty, we widen the hulls by some distance proportional to the expected uncertainty of the relative map pose to obtain the correspondence search region. That is, from the odometry filtering, we can extract the localization covariance and compute the estimated standard deviation of the two maps' relative pose estimation. We then widen the convex hull in dependence of the pose's standard deviation. Map overlap is then computed from the intersection of the search region of one map with all other search regions. By this means, an adjacency matrix is constructed. In Figure 5, one example is depicted. The point clouds of two maps are shown in dots while the convex hull is displayed as the dashed lines. After hull widening, the solid lines represent the overlap area. The red points are the ones considered for point matching.

The adjacency matrix is used to extract point correspondences between the two point clouds of overlapping maps. We apply an approach similar to the ICP algorithm [14], however we include prior knowledge from local point cloud descriptors. That is, for each point within the overlapping region of two maps, we compute a principal component analysis of the local surrounding. The covariance matrix for a point p_i is given by

$$\begin{aligned} P &= \{p_j | \, \|p_j - p_i\|_2 < \epsilon\} && (5) \\ \Sigma_i &= \Sigma_{\text{Grid}} + \frac{1}{|P|} \sum_{j \in P} (p_j - p_i)(p_j - p_i)^T && (6) \end{aligned}$$

where Σ_{Grid} denotes the inherent uncertainty of every grid cell that arises from discretization. We then only match points from different maps with similar eigenvectors. The matching for the example overlap from Figure 5 is depicted in Figure 6. The green and blue point clouds represent the two maps, the red lines depict the established correspondences. Note that even tough the finish line of the green map is closest to the middle stripe of the blue map, the correspondences between the finish lines are matched correctly.

2) Optimization: Once the correspondences have been found, global optimization is run on the set of maps. The optimization criterion is the concatenated error of all map matches and all odometry errors.

Let's assume we have constructed the adjacency matrix from the overlapping point cloud hulls according to III-B.1 and map i and j are found to be overlapping. Then the matching error is a concatenation of all M matching correspondence errors in dependence of the relative map pose $T_{i,j}$ with

$$\vec{e}_{i,j} = (T_{i,j}\vec{p}_{i,1} - \vec{q}_{j,1}, \cdots, T_{i,j}\vec{p}_{i,M} - \vec{q}_{j,M})^T \tag{7}$$

where p_i are points of map i and q_j are points of map j, both in homogeneous coordinates.

From the correspondence description, the covariance matrices of local point neighborhoods are known. For every point matching, the covariance matrix of a match is computed as the sum of the covariance matrices of the individual points.

$$\Sigma_{i,j} = \Sigma_i + \Sigma_j \tag{8}$$

Apart from the matching error, the error with respect to the odometry can be considered in optimization. The error from odometry per map is given by

$$\vec{e}_{odo,i} = \left(\hat{\alpha}_i - \alpha_i, \hat{\vec{t}_i} - \vec{t}_i\right)^T \tag{9}$$

where α_i is the rotation angle extracted from the rotation matrix in T_i. From the Kalman filtering the covariance of a map's pose $\Sigma_{odo,i}$ can be estimated.

The optimization criterion $J(T^N)$ then is the Mahalanobis distance of all odometry errors and all matching errors with

$$J(T^N) = \vec{e}^T \Sigma^{-1} \vec{e} \tag{10}$$

where $\vec{e}$ is the concatenation of all error vectors given as

$$\vec{e} = (\vec{e}_{0,1}, \vec{e}_{odo,1}, \ldots, \vec{e}_{i,N})^T \tag{11}$$

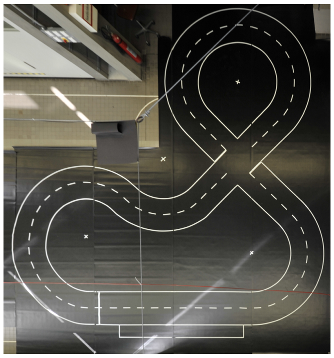


Fig. 7: Reference image of test track.

and Σ with the block diagonal form of

$$\Sigma = \begin{bmatrix} \Sigma_{0,1} & 0 & \cdots & 0 \\ 0 & \Sigma_{odo,1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \Sigma_{i,N} \end{bmatrix} \quad (12)$$

Note that not only maps with subsequent indices may contribute to the matching error and covariance but also maps of loop closure inbetween.

The use of the Mahalanobis distance also allows to fuse errors from both rotation and translation as the error measure is transformed from different units to multiples of the variance. Also, the used representation of points with information on their surrounding allows the optimization to be less sensitive to invariant features such as lines which only yield reliable translatory measurements perpendicular to the line orientation.

The optimal set of map poses $\tilde{T}^N$ is found using the Levenberg-Marquardt algorithm so that

$$\tilde{T}^N = \arg\min_{T^N} \left(J\left(T^N\right)\right). \quad (13)$$

For computational efficiency the error is computed in the form of

$$J(T^N) = \vec{e}^T L^T L \vec{e} \quad (14)$$

so that numeric differentials can be computed easily. Since the covariance matrix of the concatenated error vector is of block diagonal form, the Cholesky decomposition $L^T L = \Sigma^{-1}$ can be computed from the Cholesky decomposition of the individual blocks, i.e. the decomposition of the inverse covariance matrices of matching and odometry errors.

IV. Experiments

The work was conducted in the context of the Carolo-Cup [1] in which student teams compete with self built autonomous model race cars. Thus, the track to be mapped is a small replica of the actual race track with white lane markings on a black surface as depicted in Figure 7. The track features two loop closures, i.e. the crossing and the finish line.

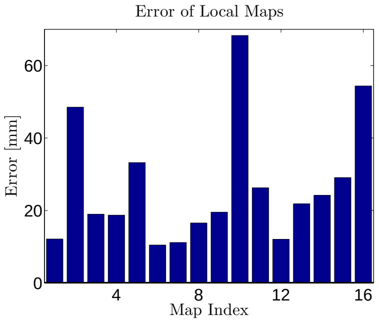


Fig. 8: Error of local maps.

A. Experimental Setup

The picture from Figure 7 served as a ground truth. It was taken with an ordinary digital camera and undistorted and transformed using a set of points measured with a laser range finder.

The model car is a 1:10 model car equipped with wheel encoders, accelerometer and gyroscope as well as a 0.3MPix camera. Processing is executed on a standard Intel Core i3-4010U processor.

The algorithm was evaluated on multiple drives. We experienced that the odometry was extremely accurate and thus the results did not sufficiently show the strength of the approach. In order to create a more challenging starting position, we introduced additive Gaussian noise with non-zero mean to the angular velocity, as it might be caused for example by unequal wheel diameters on left and right rear wheels.

B. Local Map Accuracy

The first step is to evaluate the accuracy of the local submaps. For this purpose, the middle lane marking was manually annotated in the ground truth image. We used end points of the individual lines of the marking to interpolate their midpoints. The same was done for the submaps.

For evaluation, we aligned the first stripe of every submap and computed the mean Euclidean distance between all following stripes. The average error per map is displayed in Figure 8. As it can be seen, the local error stays below the boundary of roughly 7cm. It can be noted that the local maps with the highest average error (no. 2, 10, 16) were those with the longest distance traveled within that map. This finding coincides with the expectation of increasing error over distance.

C. Global Accuracy

After the registration of the maps, the global error is evaluated. The complete map was aligned using the first five stripes using a minimum least square alignment. The midpoints of the stripes of the middle lane marking were used to evaluate the mean Euclidean distance.

The results of the global map are depicted in Figure 9. For reference, we included the global error of the unoptimized

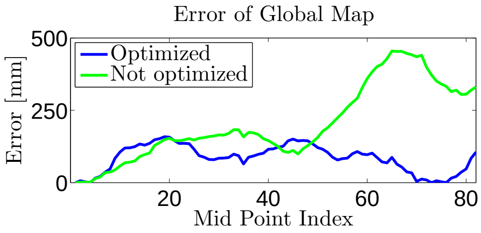


Fig. 9: Error of global map.

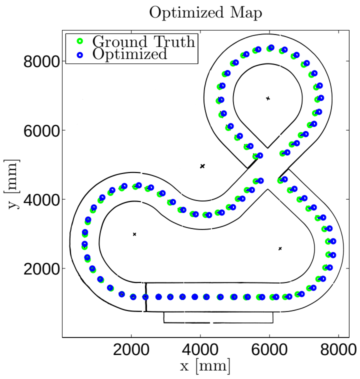


Fig. 10: Map overlay.

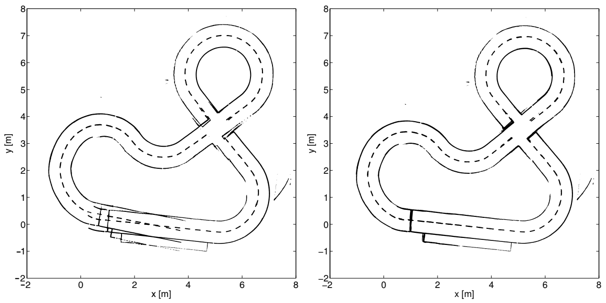
map. It can be seen that the error is drastically reduced in the case of the optimized map. Especially in the case of loop closure at midpoint index 70, the error drops close to zero. However, the error in the first loop closure stays higher.

This is best explained by looking at the overlay of ground truth and map mid points in Figure 10. Green dots denote the stripe mid points of the ground truth while blue dots represent the map stripe mid points. While the dots coincide in the region of the finish line, the distance traveled is overestimated in the first curve. With the detection of the loop closure in the upper part of the map and its optimization, only the local error within that circle can be corrected. However, the offset up to the crossing remains undetected.

In a final comparison, the global map before and after optimization is depicted in Figure 11. In Figure 11a, the non overlapping submaps in cases of loop closures are clearly visible from double road markings. However, in the optimized case in Figure 11b, these errors have been corrected and a consistent map has been built.

V. Conclusion

In this paper we have shown a relatively easy way of extending standard grid mapping to a SLAM framework. For this we introduced a submap based SLAM approach with registration and pose optimization for global alignment. The results of our work were evaluated on real world data obtained from a model race car. We argue that the proposed framework is especially suitable for applications with a high degree of ambiguity in local features such as road markings.



(a) Non optimized global map. (b) Global map after optimization.

Fig. 11: Consistent map.

The outcome of our work is also a reasonable foundation for more elaborate perception problems such as extraction of semantic information from grid maps as the submaps allow for graph solvers to be applied to connected components.

References

[1] Carolo Cup, *URL http://carolocup.de/*, 2014.

[2] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, June 1989.

[3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[4] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *Robotics Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, June 2006.

[5] R. Chatila and J. Laumond, "Position referencing and consistent world modeling for mobile robots," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, Mar 1985, pp. 138–145.

[6] J. Folkesson and H. Christensen, "Graphical slam - a self-correcting map," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 1, April 2004, pp. 383–390 Vol.1.

[7] M. Montemerlo and S. Thrun, "Fastslam 2.0," *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*, pp. 63–90, 2007.

[8] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 1, Oct 2003, pp. 206–211 vol.1.

[9] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An atlas framework for scalable mapping," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 1899–1906.

[10] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 5, pp. 698–700, 1987.

[11] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 404–417.

[12] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[13] L. Ailing and Z. Boheng, "Registration algorithm of panchromatic and multi-spectral images based on sift and ransac," in *Photonics and Optoelectronics (SOPO), 2012 Symposium on*, May 2012, pp. 1–4.

[14] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606.

[15] M. Kellner, M. Bouzouraa, and U. Hofmann, "Road curb detection based on different elevation mapping techniques," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, June 2014, pp. 1217–1224.