Hierarchical Loop Detection for Mobile Outdoor Robots

Dagmar Lang[a], Christian Winkens[a], Marcel Häselich[a], Dietrich Paulus[a]

[a] Active Vision Group, AGAS Robotics, University of Koblenz-Landau, Universitätsstr. 1, 56070 Koblenz, Germany

ABSTRACT

Loop closing is a fundamental part of 3D simultaneous localization and mapping (SLAM) that can greatly enhance the quality of long-term mapping. It is essential for the creation of globally consistent maps. Conceptually, loop closing is divided into detection and optimization. Recent approaches depend on a single sensor to recognize previously visited places in the loop detection stage. In this study, we combine data of multiple sensors such as GPS, vision, and laser range data to enhance detection results in repetitively changing environments that are not sufficiently explained by a single sensor. We present a fast and robust hierarchical loop detection algorithm for outdoor robots to achieve a reliable environment representation even if one or more sensors fail.

1. INTRODUCTION

Creation of globally consistent 3D maps can be realized by different SLAM approaches. The application of GraphSLAM[1,2] and the Iterative Closest Point (ICP) algorithm[3] depends on the quality of the pose estimation and the scan matching quality. During the scan matching process by ICP small residual errors quickly accumulate over time. In practice, this becomes obvious whenever the robot reaches and recognizes a location that he visited before. Therefore one fundamental part of most SLAM approaches is loop closing which can be separated into loop detection and optimization. During our studies we encountered different classes of loops. In Figure 1 three loop types are presented. The black line displays the trajectory of the robot and the crosses indicate positions that the robot reaches more than once.

In this paper, we present a fast and reliable hierarchical loop detection algorithm for outdoor robots. This hierarchical approach is based on GPS, camera, and 3D laser range data. According to the different sensors it is possible to find reliable loops, even if one sensor fails. Because our algorithm watches the sensor quality and changes its strategy to predict failure in case the quality drops. This is illustrated in Figure 2. So our algorithm can profit from the advantages of every sensor without suffering their disadvantages.

The remainder of this paper is organized as follows. First, related work is presented in Section 2 and is followed by a description of our sensor setup in Section 3. Afterwards our graph based SLAM approach (Section 4) and the hierarchical loop detector (Section 5) are explained. Section 6 presents the experimental setup and results, followed by conclusions and suggestions for future work in Section 7.

2. RELATED WORK

The problem of loop closing, especially loop detection for graph based SLAM, has been studied intensively in the past. According to the different SLAM approaches GPS, laser, camera or fused sensor data can be used for loop detection. A SLAM system can be divided in frontend and backend. The frontend is responsible for creating and updating the pose graph and also for loop detection, whereas the graph optimization belongs to the backend.

In order to build maps, several scans of the environment have to be integrated into one global map using a registration algorithm, eg. see[4–6] A basic loop detection approach was proposed by Nüchter et al.[4] and Sprickerhof et al.[7] During the registration of 3D scans applying the ICP algorithm, the different scan poses are compared using the Euclidean distance to detect loop closures. Granström and Schön[8] present an approach where the point clouds are described using features, which efficiently reduce the dimensions of the data. Loops are detected by a

Further author information: (Send correspondence to robots@uni-koblenz.de)

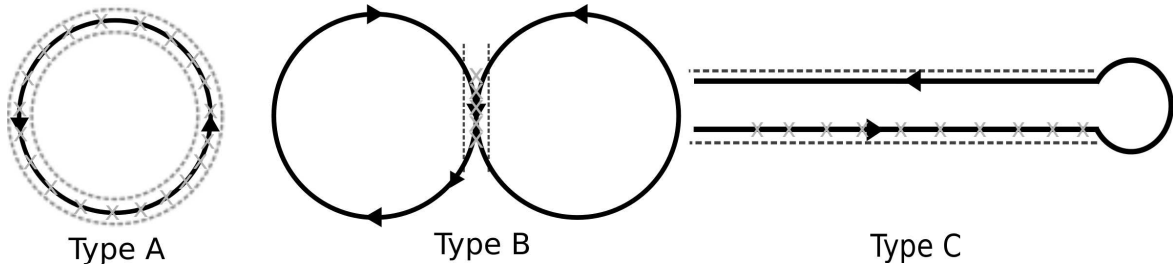


Figure 1. Different types of loops and loop closing. The black line is displaying the trajectory of the robot and the crosses are indicating the positions of the robot visited more than once.

supervised learning approach based on the generated features. For precise location description and recognition Magnusson et al.[9] used the 3D normal distribution transform to create feature histograms based on surface orientation and smoothness. This histograms are matched to find loop closures.

Another algorithm was developed by Olson et al.[10] The algorithm detects loop closures by performing pose-to-pose matches. Given a number of hypotheses generated by comparison of visual or spatial features the algorithm find a subset of hypotheses. Therefore it uses the knowledge that correct hypotheses must agree with each other and incorrect hypotheses are incorrect in different ways.

A registration approach that uses a combination of visual and 3D laser data is presented by Andreasson et al.[11] Based on this approach they proposed a global registration method that registers all scan poses simultaneously.

Loop detection with cameras provides another loop detection approach, especially for visual SLAM. Kunze et al.[12] presented several detection methods, including salient region detection and maximally stable extremal region detection. Angeli et al.[13] extended the bag of visual words method to incremental conditions and Bayesian filtering to estimate loop closure probability. Another approach using a fully probabilistic framework is called *FAB-MAP* presented by Cummins et al.[14] They represent scenes as collections of visual words from a dictionary using a bag-of-words approach. A binary vector is used for indexing the presence of words in the dictionary.

3. SENSOR SETUP

For data collection we used a human driven car. As shown in Figure 3 we mounted a *Velodyne HDL-64E S2* 3D laser range finder on top of the car and three *Philips SPC1300NC* cameras located at the front, left, and right side of the construction. For pose estimation we used a *Navilock NL - 302U* GPS receiver and an *xSens MTi* inertial measurement unit (IMU).

4. GRAPH-BASED SLAM

In our approach we use the ICP algorithm to calculate the transformation between two sets of 3D points with close positions. The *Velodyne HDL-64E S2* delivers about 1.8 million 3D points per second up to 15 Hz. A scan is defined as a set of 3D points gathered during one full rotation of the laser scanner. The large number of 3D points is impossible to handle in real-time by the ICP algorithm. For data reduction we use the results of our terrain classification[15] to extract systematically areas with significant, feature-rich objects. Only 3D points from regions with obstacles are used for the scan matching.

To calculate the transformation between two scans with close positions, a model set $\boldsymbol{M}$, with $|\boldsymbol{M}| = N_m$, and the data set $\boldsymbol{D}$, with $|\boldsymbol{D}| = N_d$, is introduced. Each set consists of one scan with 3D points $\boldsymbol{m}_i \in \boldsymbol{M}$ and $\boldsymbol{d}_j \in \boldsymbol{D}$. The ICP algorithm calculates the transformation ($\boldsymbol{R}$, $\boldsymbol{t}$), consisting of rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$, as a minimum of the cost function

$$E(\boldsymbol{R}, \boldsymbol{t}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \left\| \boldsymbol{m}_i - (\boldsymbol{R}\boldsymbol{d}_j + \boldsymbol{t}) \right\|^2 \tag{1}$$

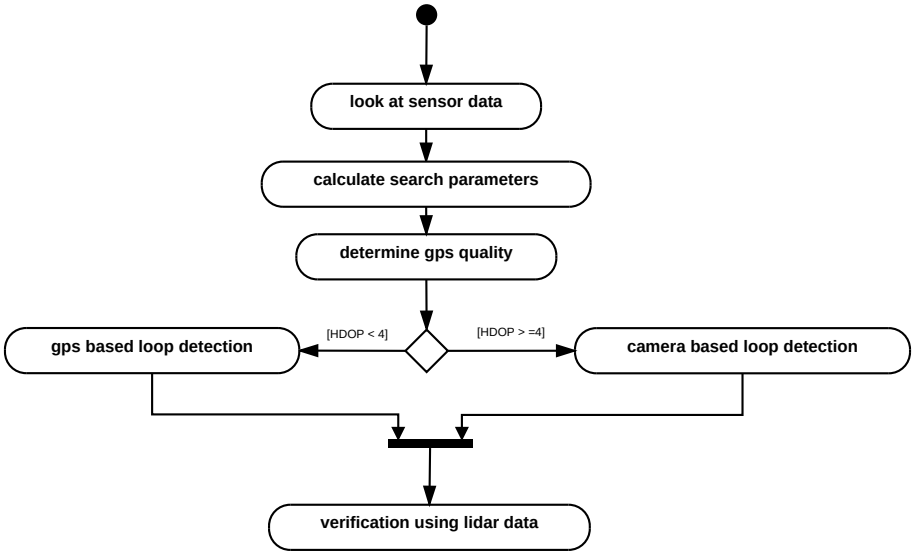


Figure 2. A flow diagram illustrating the decisions of the hierarchical loop detection algorithm.

where $w_{i,j}$ are the weights for a point match, which are set to 1 if the i-th point of $\boldsymbol{M}$ describes the same point in space as the j-th point of $\boldsymbol{D}$. Equation 1 is commonly used as abort criterion and evaluation of the quality of the minimization.

We use our own abort criterion based on dual quaternions[16] to accelerate the computations. Dual quaternion approach of Walker et al.[17] yields dual quaternion $\hat{\boldsymbol{q}}$ which is created by the quaternions $\dot{\boldsymbol{q}}$ and $\dot{\boldsymbol{s}}$ as $\hat{\boldsymbol{q}} = \dot{\boldsymbol{q}} + \epsilon\dot{\boldsymbol{s}}$. For every iteration step of the ICP algorithm $\dot{\boldsymbol{q}}$ is calculated. As small changes in $\dot{\boldsymbol{q}}$ have significant effects on Equation 1, the difference between $\dot{\boldsymbol{q}}$ of the last iteration step and $\dot{\boldsymbol{q}}$ of the previous iteration step can be used as abort criterion.

To improve the measurement of the minimization quality, we count the correctly corresponding points, where $w_{ij} = 1$, as inlier In and non assigned points, where $w_{ij} = 1$, as outlier Out. The ratio between In and Out is used for loop verification in Equation 3.

All gathered and calculated data is stored in a data structure called pose graph. The pose graph is a directed graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \phi)$ consisting of a set of nodes $\boldsymbol{v} \in \boldsymbol{V}$, which are connected by a set of edges $\boldsymbol{e} \in \boldsymbol{E}$. $\phi : \boldsymbol{E} \to \boldsymbol{V} \times \boldsymbol{V}$ defining a node as a direct predecessor of another node.

Each node $\boldsymbol{v}_i$ contains a system state $\boldsymbol{a}_i \in \boldsymbol{A}$ which consists of data gathered and calculated by the system. A system state is defined as $\boldsymbol{a}_i = (\boldsymbol{s}_i, \boldsymbol{g}_i, \boldsymbol{c}_i, \boldsymbol{i}_i)$ with a scan $\boldsymbol{s}_i$ in local coordinates, GPS position $\boldsymbol{g}_i$, images taken from the several cameras $\boldsymbol{c}_i$ and the IMU data $\boldsymbol{i}_i$ taken at time t_i in an incremental fashion. The time t_i is a system defined timestamp identifying each element, synchronizing the data. The edges contain the transformations calculated by the ICP algorithm and connect exactly two nodes with ϕ.

5. HIERARCHICAL LOOP DETECTION

In this section we present a new hierarchical approach for loop detection using multiple sensors. Our approach decides individually which sensors to use and adjusts the parameters dynamically in order to achieve the best detection results. In the following sections we present the verification criteria for detected loops. Afterwards, the different approaches using GPS and camera data are described. Depending on the sensor quality criteria different sensors are chosen for loop detection and parameters are adjusted (see Figure 2). The criteria are described in each loop detection section based on the different sensors.

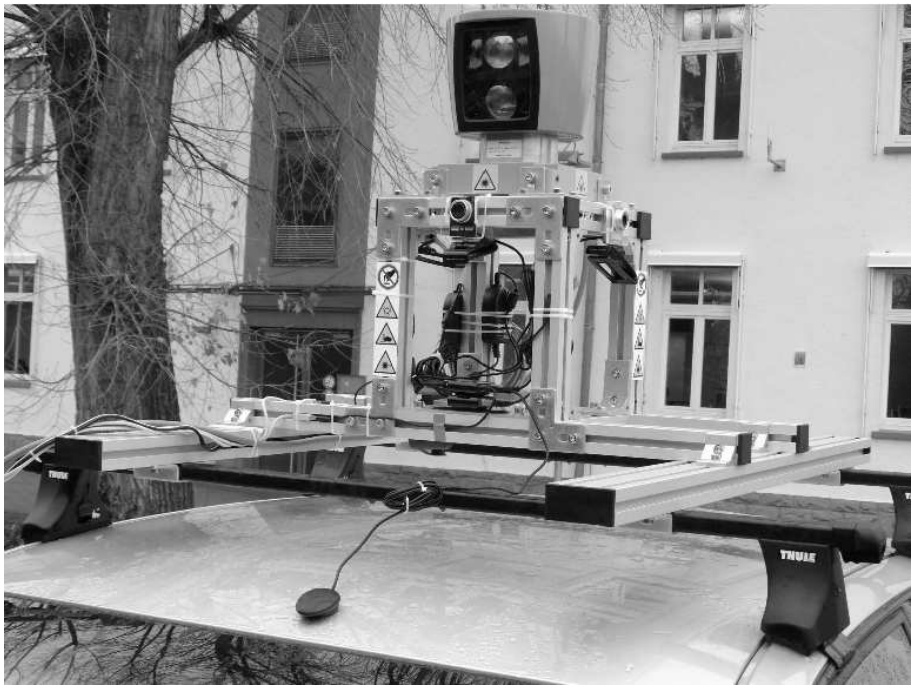


Figure 3. Sensor setup for data collection. A *Velodyne HDL-64E S2*, three cameras *Philips SPC1300NC*, *xSens MTi* and *Navilock NL - 302U* are mounted on a car to perceive the environment.

5.1 Definition of a Loop

For a proper loop detection a precise definition of a loop closure is needed. This is necessary to define the parameters for our algorithm. We define a loop closure as follows: The robot visits the same area for another time, which is called loop closure, if the following conditions are satisfied:

$$\begin{aligned} ||\boldsymbol{p}_i^w - \boldsymbol{q}_j^w|| &< D^w \\ |t_i - t_j| &> \Delta t \end{aligned} \tag{2}$$

The Euclidean distance between the two positions $\boldsymbol{p}_i^w$ (first visit / loop start) in world coordinates w and $\boldsymbol{q}_j^w$ (another visit / loop end) is below a threshold D^w and there is a minimum time spread Δt between the two visit time points t_i and t_j. In general there must be a great conformity between the system states of the loop start and end to ensure a loop, e.g. position, orientation, sensor data (laser range data, cameras, ...).

5.2 Verification of Detected Loops

Given a loop hypothesis generated by the loop detection algorithm in Section 5.3 consisting of two positions $(\boldsymbol{p}_i^w, \boldsymbol{p}_j^w)$ with corresponding nodes $(\boldsymbol{v}_{\text{start}}, \boldsymbol{v}_{\text{end}})$ of the pose graph, we need to verify this hypothesis. We take the two corresponding scans $(\boldsymbol{s}_{\text{start}}, \boldsymbol{s}_{\text{end}})$ and register them using the ICP algorithm. If the scans represent the same location they have to share common structures and the registration leads to good results. As criterion of the matching quality the inliers and outliers are counted as described in Section 4. To verify the loop, the ICP algorithm must satisfy the constraint

$$\frac{In}{In + Out} > 0.5 \quad . \tag{3}$$

In order to verify the detected loop, we need an initial estimation of the transformation

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ 0 & 1 \end{pmatrix} \tag{4}$$

between $\boldsymbol{v}_{\text{start}}$ and $\boldsymbol{v}_{\text{end}}$. If the quality of the estimation is insufficient, the ICP algorithm will not converge. As discussed by Nüchter et al.[4] the correct rotation $\boldsymbol{R}$ is much more important, than the translation $\boldsymbol{t}$ for the estimated $\boldsymbol{T}$ so we need a good rotation estimation.

The transformation estimation is bound to the loop type. Assuming type A and B, shown in Figure 1, it is possible that $\boldsymbol{s}_{\text{start}}$ and $\boldsymbol{s}_{\text{end}}$ represent the same position with nearly the same orientation. So $\boldsymbol{T} = \mathbb{I}_{4x4}$ is chosen for the estimation.

For loop type C we need another approach, according to the U-turn and the mirrored environment the ICP algorithm will fail and not converge. Therefore we use the GPS positions of $\boldsymbol{v}_{\text{start}}$ and $\boldsymbol{v}_{\text{end}}$ to compute an initial estimation

$$\boldsymbol{T}_{z_i}(\alpha) = \begin{pmatrix} \cos\alpha_i & -\sin\alpha_i & 0 & x_{i_E} \\ \sin\alpha_i & \cos\alpha_i & 0 & y_{i_N} \\ 0 & 0 & 1 & z_{i_U} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad . \tag{5}$$

With α denoted as the true course (described in Section 5.3.1) of the robot at the current position. Because the vehicle has to be nearly at the same place to close a loop according to the range of the laser range finder, we make the assumption that pitch and roll angles are unchanged. Only the orientation (yaw) is considered which corresponds to a rotation along the z-axis. Using the true course from the two positions the initial estimation is now calculated as

$$\boldsymbol{T}_{z_{\text{Diff}}} = \boldsymbol{T}_{z_{\text{start}}} \cdot \boldsymbol{T}_{z_{\text{end}}}^{-1} \quad . \tag{6}$$

If the GPS quality is insufficient for a prediction, we use the corresponding images from $\boldsymbol{v}_{\text{start}}$ and $\boldsymbol{v}_{\text{end}}$ to estimate the epipolar geometry[18] and the eight-point algorithm[19] with RANSAC[20] to get an initial estimation. Since this approach is time consuming, it is only considered if the others provide no satisfactory result. There is no possibility to know the type of the loop, therefore we tested all estimations and choose the outcome with best In/Out ratio according to Equation 3.

5.3 Loop Detection

In this section we present our different methods used during the hierarchical loop detection.

5.3.1 GPS based Loop Detection

In order to detect loop closures with GPS, it is necessary to check the GPS quality. A bad GPS quality yields enormous localization errors and false loop detections. There are several phenomena which can cause poor GPS quality. The signal is distorted by the atmosphere, each satellite has slight inaccuracies in his ephemeris, the receiver can not only gather the satellite's signal, but additional signals that bounced off buildings and other obstacles, which is called multi-path, to name only a few. To describe the inaccuracy the RMS (root mean square error), a precision estimation based on the above mentioned effects, is used. Additionally, there is a factor called dilution of precision (DOP). It acts as a multiplier to the measurement error, a low DOP value indicates a good accuracy. DOP is a unit-less mathematical representation for the quality of the GPS position and often divided into several components (HDOP, VDOP, PDOP, TDOP, GDOP) to describe the effects to the accuracy. For example, the measuring of the horizontal position is often more accurate than for the vertical. The main factors affecting DOP are the number of satellites being tracked and their position in the sky, for a low DOP you need widely spread satellites. To suit our needs for a precise localization we decided that a DOP value under 4 is necessary. Thus, GPS based loop detection is chosen as first hierarchy of the hierarchical approach (see Figure 2), if the DOP value is under 4. Given a start node $\boldsymbol{v}_b$, the search-space in horizontal r_h and vertical r_v direction is created based on the DOP values and a RMS value $\gamma = 4\,m$, with respect to the above mentioned information, as

$$\begin{aligned} r_h &= 2 \cdot \gamma \cdot HDOP_{\text{b}} \\ r_v &= 3 \cdot \gamma \cdot VDOP_{\text{b}} \quad . \end{aligned} \tag{7}$$

Where HDOP describes the dilution of precision in horizontal direction and VDOP in vertical direction. With $\boldsymbol{v}_b$ as root and the pose graph a spanning tree is created using the Dijkstra's algorithm.[21] All nodes with a shorter distance to $\boldsymbol{v}_b$ than pre-defined threshold $d_{\min}$, which depends on vehicle speed, are filtered to prevent continuous loop closing within consecutive scans. Afterwards potential loops are detected using the Euclidean distance between the GPS position $\boldsymbol{p}_b^{\text{utm}}$ of $\boldsymbol{v}_b$ and the GPS positions $\boldsymbol{p}_{i..n}^{\text{utm}}$ of all left nodes $\boldsymbol{v}_i..\boldsymbol{v}_n$ using the search-space of Equation 7. The GPS position representation is based on the Universal Transverse Mercator (UTM) geographic coordinate system that is basically a application of the 2D Cartesian coordinate system. The node with the shortest distance to the position of $\boldsymbol{v}_b$ is chosen as a potential loop closing candidate $\boldsymbol{l}_c$. In the next step the GPS quality of the loop closing candidate $\boldsymbol{g}_c$ is checked to prevent coincidence correspondence. To verify the loop candidate the ICP algorithm is executed with scan $\boldsymbol{s}_b$ from node $\boldsymbol{v}_b$ representing the beginning and the scan $\boldsymbol{s}_c$ from $\boldsymbol{l}_c$ representing the hypothesis end of the loop candidate.

5.3.2 Camera based Loop Detection

The camera based loop detection is executed if the DOP values of the current node are greater than our threshold γ as second hierarchy of the hierarchical approach. According to our task, a precise and correct localization is not possible with a potential position error of r_h. Candidate images $\boldsymbol{I}_{c_{0..2}} \in \boldsymbol{c}_c$ are taken from the robots three cameras 0..2 together with a scan $\boldsymbol{s}_c$ in an incremental fashion and applied to an image processing pipeline. This pipeline generates rectified gray images and computes speeded up robust features (SURF) features[22] and stores them in the pose graph.

Sensor Fusion In the next step the data from the *Velodyne HDL-64E S2* is fused with the image data, using the calibration from Häselich et al.[23] For the fusion we use nearest neighbor matching to pair image pixels and laser range data. This allows us to determine the distance of objects in the field of view of each camera. So it is now possible to generate a depth profile $dp_{\boldsymbol{I}}^{w_{\text{plane}}}$ of every image. This is done by projecting a plane into the image and calculating the mean distance of the objects along this plane, as

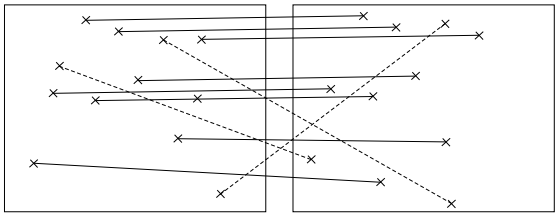
$$dp_{\boldsymbol{I}}^{w_{\text{plane}}} = \frac{\| p^{w_{\text{plane}}} \|}{n} \quad . \tag{8}$$

Where $p^{w_{\text{plane}}}$ are the 3D points in world coordinates of the plane and n the number of 3D points. Given two images $\boldsymbol{I}_i$ and $\boldsymbol{I}_j$ our algorithm is able to find all images matching the depth profiles dp_i^{plane} and dp_j^{plane}, simply by comparing two numbers. Thereby we get a set of images $\boldsymbol{I}_{k..o}$ that are potentially taken from the same place.

Feature Matching The feature matching method calculates the Euclidean distance between each *SURF* feature descriptor $\boldsymbol{f}_{i_{0..m}} \in \boldsymbol{F}_i$ from the set $\boldsymbol{F}_i$ with $m+1$ features in image $\boldsymbol{I}_i$ and all feature descriptors $\boldsymbol{f}_{j_{0..r}}$ in image $\boldsymbol{I}_j$. The distance ratio was defined by Lowe et al.[24] as the ratio of the distance to the closest neighbor and to the second-closest neighbor of a match $m = (\boldsymbol{f}_{i_m}, \boldsymbol{f}_{j_r})$. A match is found if the distance ratio is below a threshold d_{ratio}. After all matches are identified a match filtering is executed to discard outliers. All matches are filtered depending on their y-coordinate. Given a match $m = (\boldsymbol{f}_{i_m}, \boldsymbol{f}_{j_r})$, we expect that the y-coordinate of the features $\boldsymbol{f}_{i_m}$ and $\boldsymbol{f}_{j_r}$ are nearly equal in order to be a correct match. This filtering is shown in Figure 4(a).

This algorithm is capable of filtering false matches and stabilizes the loop detection based on camera images. Briefly, camera based loop detection procedure can be described as follows:

1. Given a start node $\boldsymbol{v}_b$ for loop detection the algorithm iterates over all images.
2. For every image of $\boldsymbol{v}_b$ the depth profile is checked. If one depth profile is below the threshold $dp_{\max}$, the image is chosen as reference image $\boldsymbol{I}_b$ and the algorithm proceeds to the next step. If the depth profile lies above the threshold $dp_{\max}$ the image is rejected and the next is taken. If there is no image left, the loop detection is aborted, because the images are not appropriate for precise loop detection.
3. A spanning tree is created using the Dijkstra's algorithm and all nodes with a distance lower than our threshold $d_{\min}$ are filtered out.



(a) Filtering of matches with different y-coordinates. Dashed lines represent correspondences which are rejected by the filter algorithm and solid ones are accepted.

(b) Results of the image feature matching. Circles are illustrating SURF features and lines mark matched correspondences of the features.

Figure 4. Methods and results of the accelerated feature matching.

4. The algorithm iterates over all nodes $\boldsymbol{v}_k..\boldsymbol{v}_o \in \boldsymbol{V}$ left, takes the next node $\boldsymbol{v}_i$ and extracts all images $\boldsymbol{I}_{i_{0..2}} \in \boldsymbol{c}_i$ and *SURF* feature sets $\boldsymbol{F}_{i_{0..2}} \in \boldsymbol{c}_i$

5. The depth profiles of the images $(\boldsymbol{I}_b, \boldsymbol{I}_{i_{0..2}})$ are compared. If the difference exceeds a threshold the image is rejected and the next image is taken. If no image is left the algorithm continues with the next node.

6. The feature descriptor sets $(\boldsymbol{F}_b, \boldsymbol{F}_{i_{0..2}})$ are matched using the distance ratio. If the matching ratio lies above a threshold d_{ratio} a loop candidate is found.

7. The loop candidate is verified using the method described in Section 5.2.

6. EXPERIMENTAL RESULTS

Using the described sensor setup we recorded different logfiles in urban and rural areas. In the logfiles loops of type A, B and C can be found. For evaluation we use three different logfiles:

Type A was recorded in a small round abound in the city and includes a lot of people and driving and parked cars. Data was captured during a approx. 0.4 km trail with an average speed of 17.61 km/h.

Type B was recorded on an eight shaped course of approx. 1.2 km in an urban area with an average speed of 23.48 km/h. This loop includes also a lot of people as well as driving and parked cars.

Type C was recorded on the campus at the University of Koblenz in a approx 1.0 km trail with an average speed of 16.59 km/h. All buildings on the campus look very similar.

In all of our recordings we found none, where the GPS signal quality was under our threshold. Hence we evaluated all loop detection algorithm apart from each other.

loop type	loop detection			loop verification		quantitative results		
	type	mean [ms]	std. deviation [ms]	mean [ms]	std. deviation [ms]	correct [no.]	incorrect [no.]	c_r [%]
Type A	GPS	2.68	3.59	287.70	168.37	10	0	100
	Kamera	450.10	373.03	407.40	132.20	7	0	100
Type B	GPS	2.02	4.16	424.71	236.25	34	0	100
	Kamera	1161.50	1016.90	594.63	209.60	4	0	100
Type C	GPS	3.19	6.34	453.15	195.88	32	0	100
	Kamera	1432.10	1117.50	574.95	269.87	11	6	64.7058

Table 1. Measured results of the loop detection for different loop types.

For evaluation we repeated our experiments with the described logfiles on a Intel(R) Core(TM) i7 with 2.66 GHz and 4 GB RAM. In addition to the calculation of the runtime for the several parts of the loop detection we measured of all loop detection, if they are correct or incorrect and calculated the classification rate

$$c_r = \frac{N_{\text{correct}}}{N_{\text{correct}} + N_{\text{incorrect}}} \cdot 100 \quad .$$

N_{correct} and $N_{\text{incorrect}}$ are the numbers of correct and incorrect classified loop detections. All obtained information can be found in Table 1. Loop detection time is measured from the call of the loop detection method up to the moment of finding a loop candidate or not. Verification time is excluded and measured separately. The time needed for verifying a loop consists of all executions of the ICP algorithm called by the verification method. The decision whether the found loop is correct or not was made by hand depending on the previous mentioned definition of a loop.

The runtime for the GPS loop detection algorithm is very fast, only if the loop is verified, the algorithm needs approximately 300 ms. Our mapping algorithm is updating with 1 Hz and pose estimation, data reduction and registration are measured with a mean of 200 ms and a standard deviation of 134 ms.

In contrast, camera loop detection needs more time based on the feature calculation and matching. Whereas the runtime is reduced by the sensor fusion and the depth profile. One example of the image matching of loop type A is presented in Figure 4(b). The image shows the same street and a moving car. The circles illustrate the SURF features and the lines the matched correspondences of the features. Nevertheless, incorrect classified loop detections occur very often for the logfile of type C. As mentioned before the buildings on the campus look very similar. In Figure 5 the results of the loop detection based on loop type A are presented. The image displays the results of the GPS based loop detection. The nodes of the pose graph are presented as small coordinate systems and the edges as white connections. If the loop detection algorithm identifies a loop, the new edges are drawn between the corresponding nodes and highlighted with a white sphere.

7. CONCLUSIONS

This paper presented a hierarchical loop detection for outdoor robots based on GPS and camera images. Our approach achieves reliable loop detection even if one sensor fails. The algorithm models the different sensors separately and delivers reliable good results. Furthermore we presented a feature matching approach for images that is able to reduce the time required for feature matching. The real quality and correctness of the loop detection can only be evaluated by an overall system including the loop detection and optimization.

As proposed by Ebrahimi et al.[25] feature calculation and matching the SURF features could be optimized by other features. Additionally, Andreasson et al.[11] present an approach which uses a combination of visual and 3D range information to identify correspondences. Thereby a further acceleration of our ICP algorithm for loop verification could be achieved.

Moreover loop verification using the ICP algorithm can be optimized. One drawback of the ICP algorithm is that it cannot match the geometry of the surrounding environment, only the corresponding 3D points. Therefore the real-time correlative scan matching approach of Olson[26] could be used to estimate the optimal transformation in 2D between the corresponding laser scans of the loop start and end without initial guess. 3D scans have to be reduced to 2D and the algorithm has to be extended to estimate the 3D pose. Also, correlative scan matching supports a quality measurement of the registration. Creation of a real-time 3D mapping system requires the development of a global optimization method for outdoor mapping.

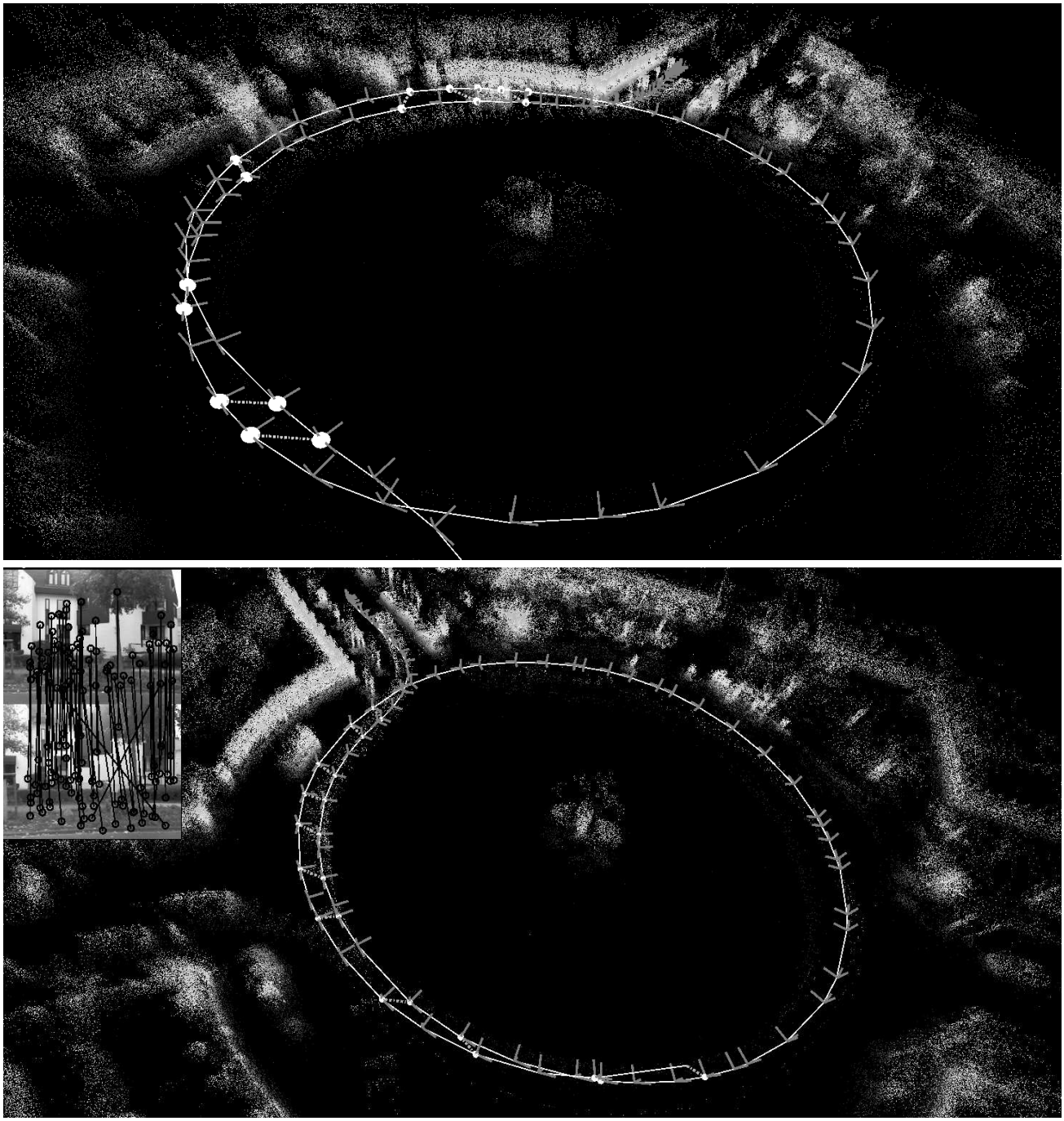


Figure 5. Results of the loop detection for loop type A with GPS based loop detection in the upper image and with camera based loop detection in the lower one. The nodes of the pose graph are displayed by small coordinate systems, that represents the orientation of each node and the laser range data. The edges, resulting from the registration by the ICP algorithm, are displayed by white lines. Furthermore, edges found by the loop detection algorithm are illustrated as dashed lines and the corresponding nodes are represented by white spheres.

REFERENCES

[1] Thrun, S. and Montemerlo, M., "The GraphSLAM algorithm with applications to large-scale mapping of urban structures," *International Journal on Robotics Research* **25**(5/6), 403–430 (2005).

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

[3] Besl, P. J. and McKay, N. D., "A Method for Registration of 3D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14** (Feb. 1992).

[4] Nüchter, A., [*3D Robotic Mapping*], Springer Tracts in Advanced Robotics, Springer Verlag (2009).

[5] Pfaff, P., Triebel, R., and Burgard, W., "An efficient extension to elevation maps for outdoor terrain mapping and loop closing," *The International Journal of Robotics Research* **26**(2), 217–230 (2007).

[6] Lamon, P., Stachniss, C., Triebel, R., Pfaff, P., Plagemann, C., Grisetti, G., Kolsky, S., Burgard, W., and Siegwart, R., "Mapping with an autonomous car," in [*IEEE/RSJ IROS Workshop: Safe Navigation in Open and Dynamic Environments*], (2006).

[7] Sprickerhof, J., Nüchter, A., Lingemann, K., and Hertzberg, J., "An explicit loop closing technique for 6d slam," in [*Proceedings of the European Conference on Mobile Robots*], 1–6 (2009).

[8] Granström, K. and Schön, T. B., "Learning to close the loop from 3d point clouds," in [*Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*], 2089–2095 (2010).

[9] Magnusson, M., Andreasson, H., Nüchter, A., and Lilienthal, A. J., "Automatic appearance-based loop detection from 3d laser data using the normal distribution transform," in [*Proceedings of the IEEE International Conference on Robotics and Automation*], 23–28 (2009).

[10] Olson, E., *Robust and Efficient Robotic Mapping*, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (June 2008).

[11] Andreasson, H. and Lilienthal, A. J., "6d scan registration using depth-interpolated local image features," *Robotics and Autonomous Systems* **58**(2), 157–165 (2010).

[12] Kunze, L., Lingemann, K., Nüchter, A., and Hertzberg, J., "Salient visual features to help close the loop in 6d slam," in [*Proceedings of the Workshop on Computational Attention and Applications*], (2007).

[13] Angeli, A., Doncieux, S., Meyer, J.-A., and Filliat, D., "Real-time visual loop-closure detection," in [*Proceedings of the IEEE International Conference on Robotics and Automation*], 1842–1847 (2008).

[14] Cummins, M. and Newman, P., "Highly scalable appearance-only SLAM - FAB-MAP 2.0," in [*Proceedings of Robotics: Science and Systems*], (2009).

[15] Neuhaus, F., Dillenberger, D., Pellenz, J., and Paulus, D., "Terrain drivability analysis in 3d laser range data for autonomous robot navigation in unstructured environments," in [*Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation*], 1–4 (2009).

[16] Pellenz, J., Lang, D., Neuhaus, F., and Paulus, D., "Real-time 3d mapping of rough terrain: A field report from disaster city," in [*Proceedings of the IEEE International Workshop on Safty, Security and Rescue Robotics*], (2010).

[17] Walker, W. and Shao, L., "Estimating 3-d location parameters using dual number quaternions," *CVGIP: Image Understanding* **54**, 358 – 367 (1991).

[18] Hartley, R. I. and Zisserman, A., [*Multiple View Geometry in Computer Vision*], Cambridge University Press, 2 ed. (2004).

[19] Hartley, R. I., "In defense of the eight-point algorithm," *Pattern Analysis and Machine Intelligence* **19**, 580–593 (6 1997).

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

[21] Dijkstra, E. W., "A note on two problems in connexion with graphs," *Numerische Mathematik* **1**(1), 269–271 (1959).

[22] Bay, H., Tuytelaars, T., and Van Gool, L., "SURF: Speeded up robust features," in [*Proceedings of the European Conference on Computer Vision*], 404–417 (2006).

[23] Häselich, M., Bing, R., and Paulus, D., "Calibration of Multiple Cameras to a 3D Laser Range Finder," in [*Proceedings of the IEEE International Conference on Emerging Signal Processing Applications*], (2012). to appear.

[24] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision* **60**, 91–110 (2004).

[25] Ebrahimi, M. and Mayol-Cuevas, W., "Susure: Speeded up surround extrema feature detector and descriptor for realtime applications," in [*"Workshop on Feature Detectors and Descriptors: The State Of The Art and Beyond" as part of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*], 9 –14 (2009).

[26] Olson, E., "Real-time correlative scan matching," in [*Proceedings of the IEEE International Conference on Robotics and Automation*], 4387–4393 (2009).